IBM

Exploiting HACMP 4.4: Enhancing the Capabilities of Cluster Multi-Processing

Determining the state of your application with Application Monitoring

Getting the most out of Tivoli Cluster Monitoring

Utilizing Cascading without Fallback

Yoshimichi Kosuge
Chizuru Hirano
Octavian Lascu
Claudio Marcantoni
Janez Vovk

Redbooks

ibm.com/redbooks

SG24-5979-00

International Technical Support Organization

Exploiting HACMP 4.4: Enhancing the Capabilities of Cluster Multi-Processing

December 2000

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 279.

First Edition (December 2000)

This edition applies to High Availability Cluster Multi-Processing for AIX Version 4.4.0, Program Number 5765-E54 for use with the AIX Version 4.3.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JN9B Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Contents

Preface

IBM High Availability Cluster Multi-Processing for AIX (HACMP) is designed to detect system failures and manage fallover to a recovery processor with a minimal loss of end-user time. HACMP Version 4.4.0 offers improved usability, more flexible installation options, and additional hardware and software support for RS/6000 customers with mission-critical applications.

This IBM Redbook provides information on Application Monitoring, Tivoli cluster monitoring, cascading without fallback, cluster verification enhancements, tuning parameters, administrative task enhancements, NFS function of HACMP 4.4, and upgrading/migrating to HACMP 4.4.

This IBM Redbook is intended to help IBM customers, IBM business partners, IBM sales professionals, and IBM I/T specialists interested in using HACMP 4.4.

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. Since he joined IBM, he has worked in the following areas: LSI design, S/390 CP microcode, VM, MVS, OS/2, and AIX. After joining the ITSO in 1998, he has been involved in writing redbooks and teaching IBM classes worldwide on all areas of AIX, HACMP, PSSP, and RSCT.

Chizuru Hirano is an IT specialist at the IBM Japan Systems Engineering Co. Ltd., in Makuhari, Japan. She has worked at IBM for six years, has four years of experience with AIX and RS/6000s, and currently works in the HACMP support team in Japan.

Octavian Lascu is an IT Specialist in IGS Romania. He is an IBM Certified Advanced Technical Expert in AIX/SP and holds a degree in Electronic Engineering from Polytechnic Institute in Bucharest, Romania. He has 8 years of UNIX experience. He has worked at IBM for 9 years. He also teaches PSSP and HACMP courses. His areas of expertise include AIX, SP, HACMP, Networking, and Linux.

Claudio Marcantoni is an instructor at the IBM Education Center of Novedrate, Italy, where he teaches HACMP courses on standalone RS/6000 systems and the RS/6000 SP. He has 10 years of experience in the AIX field. He has worked at IBM for 11 years. His areas of expertise include AIX, Networking, and HACMP. He is a co-author of the HACMP Enhanced Scalability Handbook.

Janez Vovk is an IT Specialist in ITS Slovenia, and works in post-sales support and AIX L2 support for Eastern Europe and Russia in Ljubljana. He has 3 years of AIX experience and has worked at IBM for one year. His areas of expertise include AIX, HACMP, and SP. He holds a degree in Mathematics from the University of Ljubljana, Slovenia.

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

IBM Austin
Budi Darmawan
Ernest A. Keenan
Dennis Ross
Holger Stamme

IBM Poughkeepsie
Michael K Coffey

IBM Germany
Bernhard Buehler

Tivoli Systems
Yoichiro Ishii

eMpact Solutions Inc., Wehawken, NJ.
Gabriel Radu

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 “IBM Redbooks review” on page 297 to the fax number shown on the form.
- Use the online evaluation form found at **ibm.com**/redbooks
- Send your comments in an Internet note to redbook@us.ibm.com

Chapter 1. Introduction

IBM High Availability Cluster Multi-Processing for AIX (HACMP) Version 4.4.0 was announced on June 20, 2000. HACMP is designed to detect system failures and manage fallover to a recovery processor with a minimal loss of end-user time. HACMP 4.4.0 offers improved usability, more flexible installation options, and additional hardware and software support for RS/6000 customers with mission-critical applications.

1.1 Terminologies

HACMP has had many variations through its long history. To avoid confusion, we will first define the following, which are intended only for the discussion in this book.

HAS Used for HACMP LPPs that are formally referred to as *HACMP classic*. This does not use RS/6000 Cluster Technology (RSCT).

ES Used for HACMP LPPs also known as HACMP Enhanced Scalability. This uses RSCT.

HACMP Encompasses both HAS and ES.

1.2 HACMP 4.4 Enhancements

Several enhancements were added in HACMP 4.4. These enhancements are covered in detail later in this redbook.

Application Monitoring (ES only)

This allows Process Application and Custom Application monitoring to determine the state of an application, and to restart the application or fall the resource group over to another node in the case of failure.

This function is accessed through the SMIT panels. The user may define one of two modes; Process Application Monitoring or Custom Application Monitoring. Process Application Monitoring detects the death of one or more processes using RSCT Event Management. Custom Application Monitoring checks the health of an application at user-specified polling intervals, and takes user-specified action upon detection of a problem.

In either case, when a problem is detected HACMP attempts to restart the application for a user-specified number of times. When the application cannot be restarted, HACMP is designed to cause one of the following pre-defined actions:

1. The resource group containing the application falls over to the node with the next highest priority according to the resource policy.
2. Generate a server_down event to inform the cluster of the failure.

For more information, refer to Chapter 2, “Application Monitoring” on page 5.

Tivoli cluster monitoring

Users with Tivoli management can now monitor the state of an HACMP cluster and its components on a Tivoli Desktop window. This function provides monitoring capability similar to HAView, but uses the Tivoli management interface instead of NetView. With this function, it is possible to monitor the state of the HACMP clusters and cluster components. This includes nodes and networks. The ES also supports monitoring the state of the resource groups and showing the ownership and location of the individual resources. For more information, refer to Chapter 3, “Tivoli cluster monitoring” on page 65.

Cascading without fallback

This function enhances the fallover policy to permit specifying that the resource group not return to the original node when that node rejoins the cluster. In the HAS, the number of nodes that can participate in a cascading without fallback resource group is two. For more information, refer to Chapter 4, “Cascading without fallback” on page 157.

Cluster verification enhancements

The cluster verification utility is enhanced to detect additional start-up and fallover problems:

- Check for invalid characters in cluster names, node names, network names, adapter names, and resource group names
- Check each cluster node to determine whether multiple serial networks exist on the same tty device
- Make cluster verification optional during cluster synchronization
- Check to ensure that no more than two non-IP networks of one type exist per node (ES only)

For more information, refer to Chapter 5, “Cluster verification enhancements” on page 181.

New tuning parameters

This enhancement is designed to provide easier and more granular control over parameters that affect the cluster’s performance. Through SMIT, the user may now specify:

- High and low watermarks for I/O pacing
- Syncd frequency rate
- HACMP failure detection rate (heartbeat rate and HACMP cycles to failure)

For more information, refer to Chapter 6, “Tuning parameters” on page 193.

Enhanced LVM TaskGuide

The enhanced LVM TaskGuide provides a display of the physical location of each available disk, and will create automatically a JFS log file. The TaskGuide panel for choosing physical volumes now displays the physical location of each available disk. This display allows you to determine at a glance whether the disk you choose belongs to other nodes besides the nodes you currently have selected for your shared volume group. For more information, refer to Section 7.1, “Enhanced LVM TaskGuide” on page 209.

C-SPOC file system enhancements

C-SPOC is enhanced to ease the process of creating a file system on a shared volume group by allowing the creation of a logical volume prior to creating a file system, and the creation or use of an existing log logical volume.

For more information, refer to Section 7.2, “C-SPOC file system enhancements” on page 212.

C-SPOC password configuration enhancements

The C-SPOC functionality that allows the administrator to define a user is enhanced to include a new/changed user password. This is similar to the functionality provided by the SMIT “Change a User’s Password” panel. The new/changed password will be reflected to all appropriate cluster nodes. Refer to Section 7.3, “C-SPOC password configuration enhancements” on page 221.

HACMP logs on non-local file systems

If the target directory for an HACMP log is on a remotely mounted file system, or on a shared volume group, the user is prompted with a warning that use of a non-local file system for HACMP logs will prevent log information from being collected if the file system is unavailable, and asked to confirm the choice. For more information, refer to Section 7.4, “HACMP logs on non-local file systems” on page 226.

NFS Migration (HAS only)

This feature provides for migration from HANFS 4.3.1 to HAS 4.4. HANFS is no longer included as a feature of HACMP. This feature provides a means for performing a migration from a running HANFS 4.3.1 cluster to a running HAS 4.4 cluster without bringing the cluster off-line. For more information, refer to Chapter 8, “HACMP 4.4 and NFS” on page 231.

Chapter 2. Application Monitoring

Application Monitoring provides Process Application Monitoring and Custom Application Monitoring to determine the state of an application, and to restart the application or fallover the resource group to another node in case of application failure. This provides a way to avoid unplanned downtime.

Application Monitoring is only available in ES 4.4.

2.1 System downtime

System downtime identifies the time-frame when a computer system is not working properly. System downtime can be either *planned* or *unplanned*. Typical examples of planned downtime are backups, software upgrades, hardware upgrades, and so on. While unplanned downtime is caused by an unexpected event. Examples of unplanned downtime are hardware failures, software failures, user errors, and so on.

At the beginning of its life, the HACMP was designed in such a way to be able to manage the following three failures:

- Node failure
- Network adapter failure
- Network failure

These three failures are obviously hardware failures.

The Application Monitoring available in ES allows the user to monitor one or more applications, defined through SMIT, and to specify actions the system should take upon detection of process death or application failure. which extends the capabilities of ES to manage a very common software failure, the failure of the customer application, in addition to the three types of hardware failure.

2.1.1 User-defined events

While Application Monitoring is introduced with ES 4.4, an earlier version of ES already included an infrastructure to allow the user to configure so-called user-defined events.

By configuring user-defined events, the user is able to customize the HACMP to monitor and manage many common software problems; CPU usage, system memory usage, process life, and many others. However, the configuration of user-defined events through the /usr/es/sbin/cluster/events/rules.hacmprd file has proven to be a complicated task for most people. Errors updating the rules.hacmprd file can easily cause cluster nodes to halt at startup.

Application Monitoring now provides a much simpler and user-friendly interface to monitor and react in case of application failure, allowing more flexibility on what actions to take. Instead of having to manually edit critical configuration files like rules.hacmprd, Application Monitoring can be set through SMIT, reducing the possibility of error.

The following publications are excellent sources of information about user-defined events:

- *HACMP Enhanced Scalability User-Defined Events*
- *HACMP Enhanced Scalability Handbook*

All the user-defined events configured in the earlier version of ES still work in ES 4.4.

2.2 Application monitoring overview

The ES provides two different application monitors:

- *Process Application Monitor*
- *Custom Application Monitor*

The *Process Application Monitor* is often referred to as just *Process Monitor*. The *Custom Application Monitor* is often referred to as *Custom Monitor* or *User-Defined Monitor*. Both allow you to monitor an application and take an action when the application fails.

The Process Monitor relies upon the Event Management (EM) infrastructure provided by RS/6000 Cluster Technology (RSCT), while the Custom Monitor uses programs or shell scripts written by users.

Process Monitor is easier to configure, as it uses the built-in monitoring capabilities provided by RSCT and does not require any custom shell scripts. However, it has the limitation that it can only monitor applications (actually processes) that are executable binaries. It cannot monitor shell scripts because they are abbreviated in the system process table rather than being listed by the full shell script name.

On the other hand, Custom Monitor provides more customization options to users, but requires more planning and user-written shell scripts. While Process Monitoring relies on RSCT, Custom Monitoring requires users to write a shell script to monitor the state of their application. This script is called *Monitor Method*. The script is not limited to just checking if the application is running or not, it may also test other aspects of the application such as its response time.

2.2.1 Application Monitoring components

Figure 1 helps you to better understand the difference between Process Monitor and Custom Monitor. For both monitors, when a Resource Group is acquired by a node, the cluster manager (clstrmgr) checks if a monitor has been configured for the Application Server defined inside this Resource Group. If monitor is configured, the cluster manager starts the `run_clappmond` daemon, which launches the `clappmond` daemon.

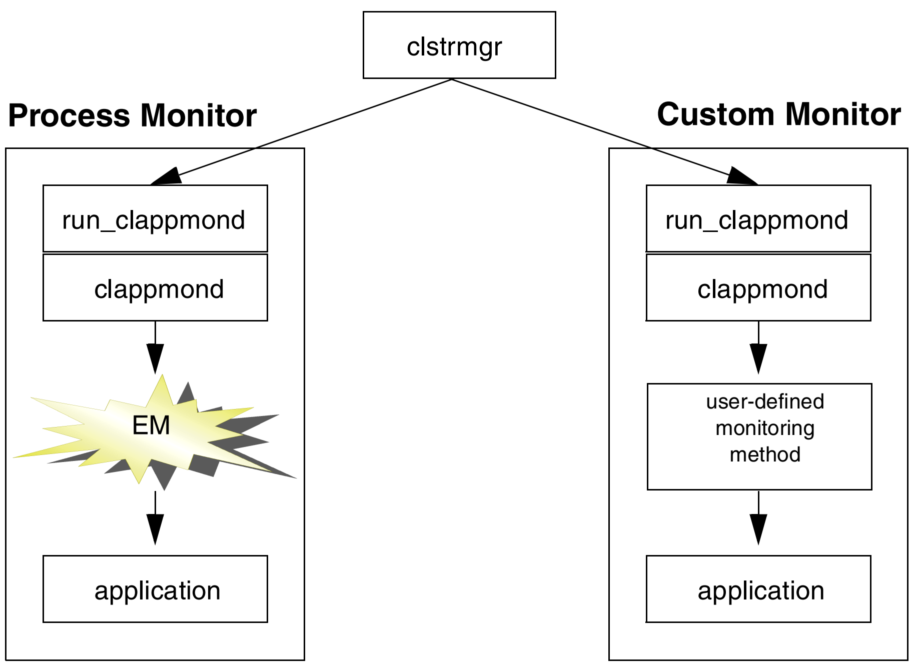

Figure 1. Application monitoring components

In case of Process Monitor, `clappmond` registers a resource variable to the EM to monitor the Application Server, while in Custom Monitor, `clappmond` invokes the user-written shell script to monitor the Application Server.

When `clappmond` detects the failure of the application, it exits. The cluster manager realizes this and takes the appropriate action to make the application available again by either trying to restart it on the same cluster node or by moving it to a different node in the same cluster.

The cluster manager starts one pair of instances, the `run_clappmond` and `clappmond` daemons, for each Application Server being monitored.

2.2.2 Application Monitoring prerequisites

Keep the following requirements in mind when planning both Process and Custom Monitors:

- Application Monitoring is only available in ES.
- Any application to be monitored must be defined inside an Application Server, and this Application Server must be present in a Resource Group.
- Only one Application Server per Resource Group can be monitored. In case you need to monitor multiple applications, one Resource Group must be configured for each application.
- Any monitored application can only be present in one Resource Group.
- The monitored application can not be under the control of the System Resource Controller (SRC).

2.3 Process Application Monitor

This section provides information on configuration parameters, new events, and diagrams of Process Monitor.

2.3.1 Configuration parameters

Figure 2 on page 9 shows the SMIT menu to adjust the Process Monitor configuration parameters. This menu can be reached with the `smit clappserv_to_monitor_by_process.select` fastpath.

```
                    Add Process Application Monitor

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

                                                [Entry Fields]
* Application Server Name                     imageappsrvr
* Processes to Monitor                       []
* Process Owner                              []
  Instance Count                             []                      #
* Stabilization Interval                     []                      #
* Restart Count                              []                      #
  Restart Interval                           []                      #
* Action on Application Failure              [notify]                +
  Notify Method                              []
  Cleanup Method                             [/usr/sbin/cluster/even>
  Restart Method                             [/usr/sbin/cluster/even>



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
```

Figure 2. Process Monitor SMIT menu

This SMIT menu includes the following fields:

Note

The information typed in this SMIT menu is saved in a new ODM object class called HACMPmonitor.

Application Server Name

This field contains the name of the Application Server.

Processes to Monitor

Specify the name of the process to monitor. In case of multiple processes, use spaces to separate them.

Note

To discover the name of the process(es), use the `ps -el` rather than the `ps -ef` command. See Section 2.5.1.3, “Identifying the correct process name to monitor” on page 21 for details.

Process Owner
Specify the user who is the owner of all the processes being monitored.

Instance Count
Specify the number of instances of a process to monitor. The default value is 1. This value must be 1 if more than one process is listed in the Processes to Monitor field (each specified process is allowed one instance only).

Stabilization Interval
Specify the time in seconds to wait for the application to stabilize before monitoring begins. In most cases, when an application starts, it needs a certain amount of time to become stable. This parameter allows you to delay the monitoring until after the application start shell script has been run and the application is stable.

Note

We strongly recommend not to specify 0. See Section 2.7.1, “Examining the Stabilization Interval” on page 56 for details.

Restart Count
Specify the number of times ES will try to restart the failed application on the same cluster node where it failed. To try to restart the failed application, ES executes the script specified in the Restart Method field. If all attempts to restart the application fail, ES executes the Failure Action. The default value for this field is 3. Setting it to 0 means no attempt will be made.

Restart Interval
Specify the interval in seconds that the application must remain stable before resetting the Failure Count to 0. If this field is left empty, ES assigns the value equal to the following:

```
((Restart Count)*(Stabilization Interval)*1.1)
```

This value is also the minimum value that you can specify.

Action on Application Failure
Specify the action to be taken if the failed application has not been restarted within the Restart Count value. There are two possible values: *notify* (the default) and *fallover*. When *notify* is selected, ES runs the script defined in the Notify Method field to inform the user of the application failure. When *fallover* is selected, ES will move the Resource Group containing the failed Application Server to the next highest priority node for this Resource Group.

Note

It is important not to confuse the meaning of the word *fallover* in this context. Usually fallover is associated to the situation where one cluster node goes down and *all* resources move to another node. However, in the current context, fallover means that only the Resource Group containing the failed Application Server is moved to a backup node. If the cluster node that had control of the failed application also owns a second Resource Group, this second Resource Group is not moved to the standby node.

We often refer to this field as 'Failure Action.'

Notify Method
Specify the full path name of the script to be executed when the monitored application fails. This script is run each time an application is successfully restarted, fails completely, or is moved to a standby cluster node. Typically, this script would inform the user of the action taken by ES.

Cleanup Method
Specify the full path name of the script to be run to stop the application. When the application fails, the Cleanup Method script is executed before trying the restart using the Restart Method script. By default, this field is set to the Application Server stop script. The main objective of this script must be to return the failed application to a known state (open files, buffers, and so on.) so that a subsequent restart can succeed.

Note

The Cleanup Method script may be invoked when the application has already failed.

Restart Method
Specify the full path name of the script to run in order to try to restart the failed application. The default value is the Application Server start script. This field can be left blank in case the Restart Count parameter is equal to 0.

2.3.2 New cluster events

In order to implement Process Monitor, ES 4.4 introduces the following three new cluster events:

`server_restart`
This event is executed each time ES tries to restart the application

using the Restart Method script. ES tries to restart the application on the same cluster node where it has failed. This event handles Notify Method, Cleanup Method, and Restart Method.

server_down
This event is executed when all attempts to restart the failed application have been unsuccessful and the Failure Action field is set to “notify”. This event handles Notify Method.

rg_move
This event is executed when ES moves the resource group containing the failed application to the next highest priority node. It is only executed when the Failure Action field is set to ’fallover.’

2.3.3 Diagrams

In this section we show different diagrams of Process Monitor using three scenarios.

2.3.3.1 Successful restart with the first attempt

In this scenario ES successfully restarts the failed application. Refer to Figure 3 for the diagram of this scenario.

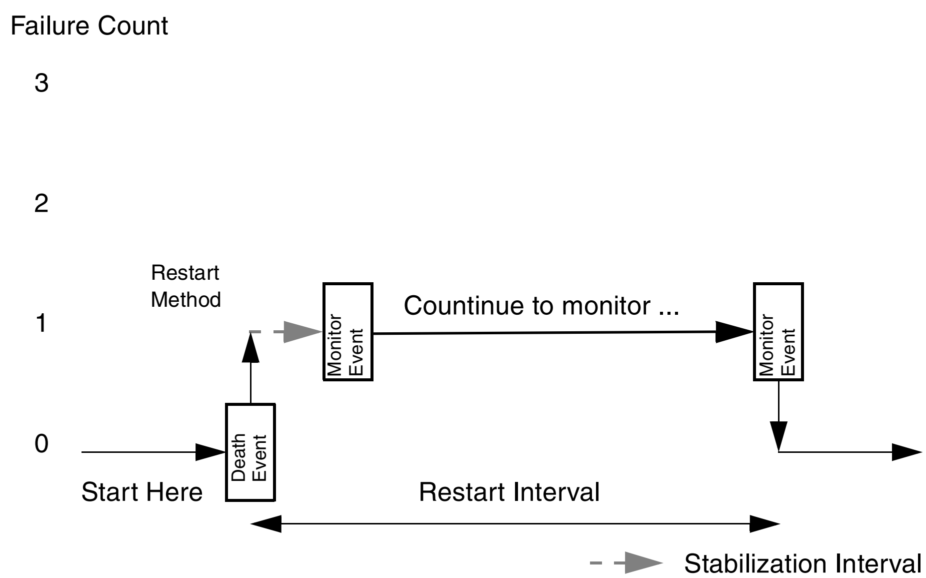

Figure 3. Process Monitor successful restart

We assume the Restart Count parameter is set to a value of 3. On the left side of the figure we have a column representing the Failure Count. This counter is incremented by 1 each time the monitored application fails.

The lower left corner of the figure represents the starting point of the diagram. ES has started, the application is running successfully, and the monitoring has begun. The Failure Count is now equal to 0. The full arrow reaches the first rectangle because EM notifies the death of the process. A Monitor Event occurs each time ES checks the process table in order to find out if the application is still running. In this case the application has failed and for this reason the Failure Count is incremented to 1. By executing the Restart Method script, ES tries to restart the application and then waits for the Stabilization Interval to expire. The Stabilization Interval is represented by the dashed arrow. When the Stabilization Interval has expired, the monitoring resumes, as shown by the rectangle. The monitoring of the application continues as represented by the full arrow. When ES reaches the Restart Interval with the application still running, the Failure Count is reset to 0. At this point the handling of this application failure is completed. ES continues monitoring.

2.3.3.2 Successful restart with the second attempt

In the next scenario ES successfully restarts the failed application on the second attempt. Refer to Figure 4 on page 14 for the diagram of this scenario.

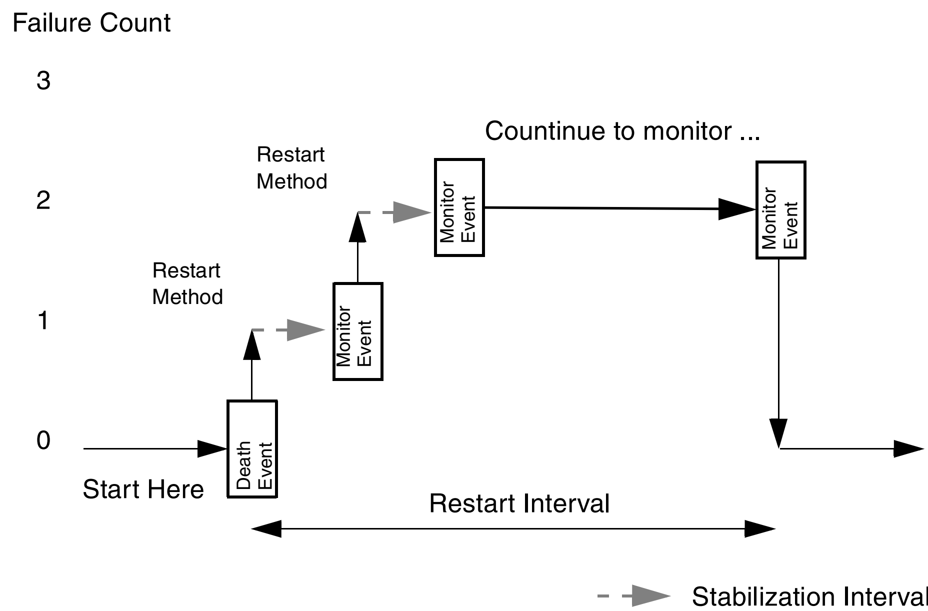

Figure 4. Process Monitor second successful restart

We again assume the Restart Count parameter is set to a value of 3. The Failure Count begins at 0. ES has started, and the monitored application is running successfully. ES reaches the first rectangle, which represents death of the process notified by EM. Failure Count is incremented to 1, and ES executes the Restart Method script to reactivate the application, then waits for the Stabilization Interval (represented by the dashed arrow) to expire. When it has expired, the monitoring resumes and we now reach next rectangle. The application is still not running, so the Failure Count is incremented to 2. ES again executes the Restart Method script and waits for the Stabilization Interval to expire. When monitoring resumes, this time the application is running so ES waits for the Restart Interval to expire, then considers that the application has been running safely long enough to reset the Failure Count to 0. The handling of these two application failures is now complete, and the monitoring continues.

2.3.3.3 Unsuccessful restart

In the last scenario ES is unsuccessful in restarting the failed application. Refer to Figure 5 on page 15 for the diagram of this scenario.

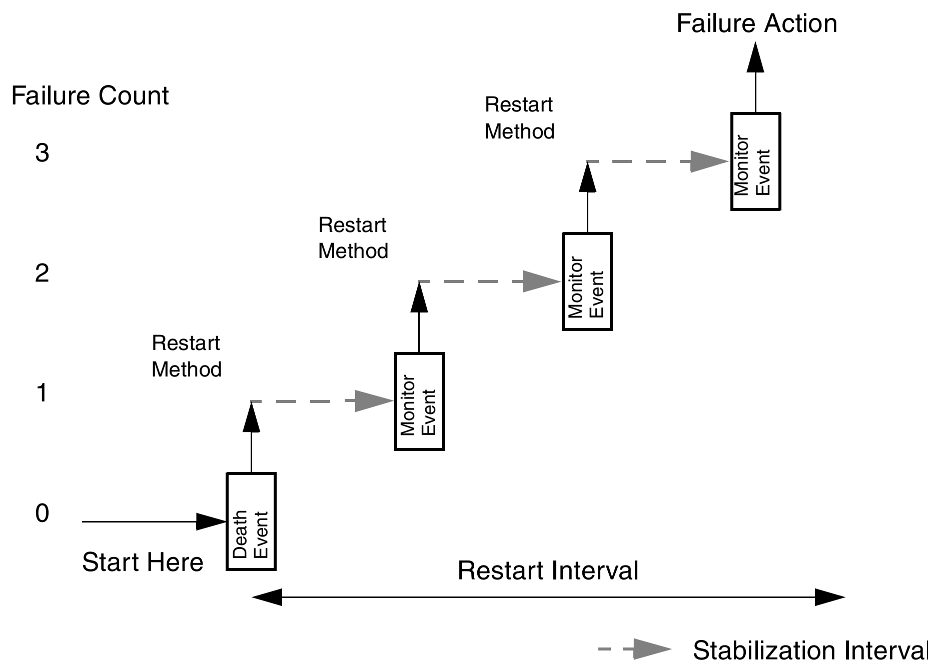

Figure 5. Process Monitor unsuccessful restart

We assume the Restart Count parameter is set to a value of 3. The Failure Count begins at 0. ES has started, and the monitored application is running successfully. ES reaches the first rectangle, the death of the process. The Failure Count is incremented to 1, and ES executes the Restart Method script to reactivate it. When the Stabilization Interval (dashed arrow) expires, the monitoring resumes and ES reaches next rectangle. The application is still not running, so the Failure Count becomes 2 and the Restart Method is run again to try to restart the application. After the Stabilization Interval has expired ES checks again if the application is running safely. It is not and so the Failure Count is incremented to 3. After waiting for the Stabilization Interval, ES checks if the application is running and it is not. This is equal to the Restart Count, so ES executes the Failure Action. If the Failure Action is set to *notify*, you are informed by the Notify Method script that your application will not run anymore. If the Failure Action is set to *fallover*, ES moves the Resource Group containing the failed application to another cluster node, where it will try again to restart the application.

2.4 Custom Application Monitor

This section provides you with information on configuration parameters, new events, and diagrams of Custom Monitor. This helps you to understand what Custom Monitor is.

2.4.1 Configuration parameters

We explain the configuration parameters available in Custom Monitor.

Figure 6 shows the SMIT menu to configure Custom Monitor reachable with the `smit clappserv_to_custom_monitor.select` fastpath.

```
                  Add Custom Application Monitor

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

                                                        [Entry Fields]
  Application Server Name                             imageappsrvr
* Monitor Method                                      []
  Monitor Interval                                    []                       #
  Hung Monitor Signal                                 []                       #
* Stabilization Interval                              []                       #
  Restart Count                                       []                       #
  Restart Interval                                    []                       #
* Action on Application Failure                       [notify]                +
  Notify Method                                       []
  Cleanup Method                                      [/usr/sbin/cluster/even>
  Restart Method                                      [/usr/sbin/cluster/even>


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
```

Figure 6. Custom Monitor SMIT menu

Most of these parameters are exactly the same as those found in the SMIT menu to configure Process Monitor. In this section we only explain the parameters that are new or have a different meaning. For the remaining parameters, refer to Section 2.3.1, “Configuration parameters” on page 8.

Note

The information typed in this SMIT menu is saved in a new ODM object class called HACMPmonitor.

Monitor Method
The full path name of a user-written shell script that examines if the monitored application is running or not. It must exit with a return code of 0 if the application is running and with a return code different from 0 if it is not running. Arguments cannot be passed to the Monitor Method script on this field.

Monitor Interval
The Monitor Method script is run periodically at this interval (expressed in seconds). Also, if the execution time of the Monitor Method script is longer than the Monitor Interval, the script is delivered the signal specified in the Hung Monitor Signal field.

Hung Monitor Signal
The signal sent to terminate the Monitor Method script if it does not complete execution within the number of seconds specified as the Monitor Interval.

Restart Interval
Specify the interval in seconds that the application must remain stable before resetting the Restart Count to 0. If this field is left empty, ES assigns a default value equal to the following:

```
((Restart Count)*(Stabilization Interval + Monitor Interval) * 1.1)
```

This value is also the minimum value that you can specify.

2.4.2 New cluster events

The three new events explained in Section 2.3.2, “New cluster events” on page 11 are also used in Custom Application Monitor, and have exactly the same meaning.

2.4.3 Diagrams

In this section we will show diagrams of Custom Monitor from two scenarios.

2.4.3.1 Successful restart

The first scenario shows a case in which ES is successful in restarting the failed application. Refer to Figure 7 on page 18.

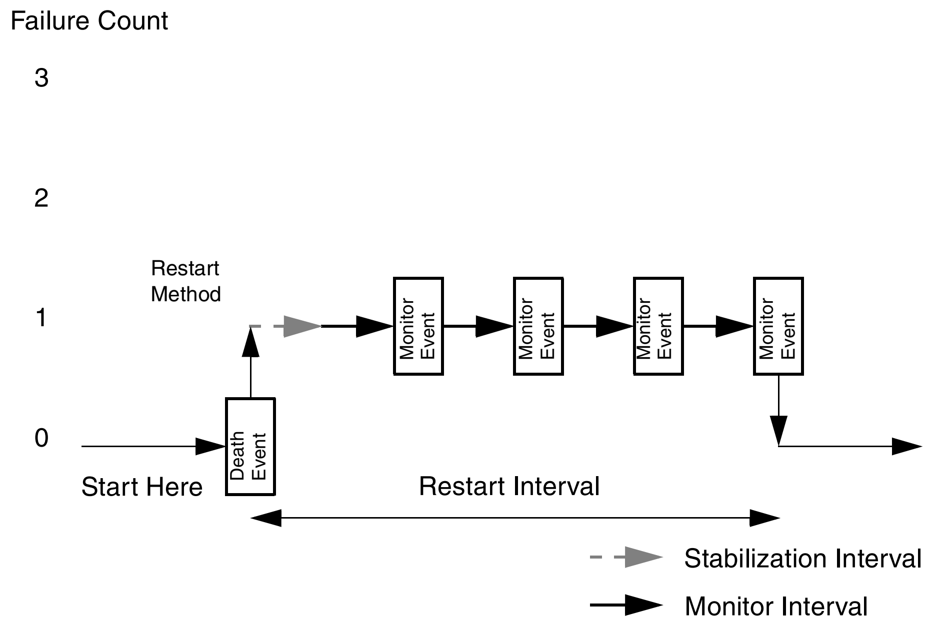

Figure 7. Custom Monitor successful restart

The difference from Process Monitor is that the monitoring is performed by a user-written script called Monitor Method, and that we have an additional interval called Monitor Interval.

We assume the Restart Count parameter has been set to a value of 3. The column on the left of the figure represents the Failure Count, which starts at 0. The lower left corner of the figure is our starting point. ES has started, the application is running successfully, and the monitoring has begun. ES reaches the first rectangle and checks if the application is running. The application has died, so the Failure Count becomes 1 and the Restart Method script is executed. ES now waits for the Stabilization Interval (dashed arrow) to expire, and then waits for the Monitor Interval (full arrow) to expire before resuming the monitoring, shown by the second rectangle. The monitoring continues periodically, as represented by the following three rectangles. The time-frame between each rectangle is determined by the Monitor Interval parameter. When ES reaches the Restart Interval with the application still running safely, the Failure Count is reset to 0. At this point the handling of this application failure is completed and the monitoring continues.

2.4.3.2 Unsuccessful restart

The second scenario shows a case in which ES cannot restart the failed application. Refer to Figure 8 for the diagram of this scenario.

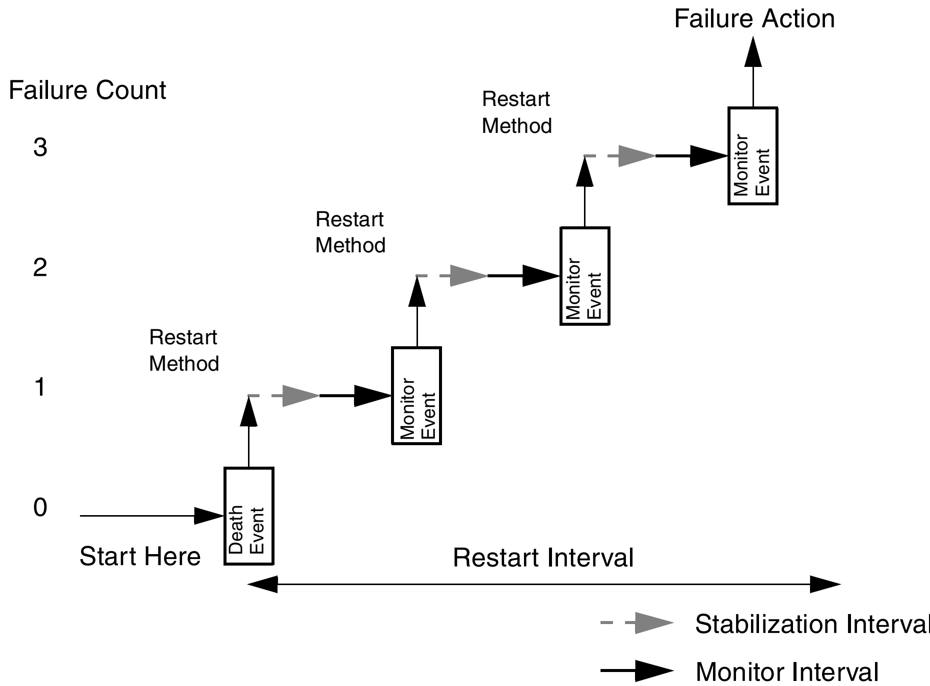

Figure 8. Custom Monitor unsuccessful restart

Again, the Restart Count parameter is presumed to be 3. The Failure Count is equal to 0. ES has started, the application is running safely, and the monitoring has begun. ES reaches the first rectangle and realizes the application has died, so the Failure Count becomes 1. After executing the Restart Method script, ES waits for the Stabilization Interval (dashed arrow) and Monitor Interval (full arrow) to expire. ES checks again if the application is running, but it is not. The Failure Count becomes 2 and ES waits for the two intervals and attempts to restart twice more, after which it has reached the Restart Count and executes the Failure Action. If the Failure Action is set to *notify*, you are informed by the Notify Method script that your application will not run anymore. If the Failure Action is set to *fallover*, ES moves the Resource Group containing the failed application to another cluster node where it will try again to restart the application.

2.5 Process Application Monitor examples

This section describes our experience of configuring and understanding Process Monitor.

2.5.1 Common configuration

As an sample application we decide to use the Image Cataloger Demo that is provided with the HACMP software. It is a very simple application that starts a server process called imserv. The objective of this Process Monitoring is to constantly monitor the life of the imserv process and to notify the user in case of failure. For information on this demo program, refer to *HACMP V4.3 AIX: Install Guide*.

2.5.1.1 Application Server definition

Figure 9 shows the configuration of the Application Server in SMIT reachable with the `smit claddserv.dialog` fastpath.

```
                  Add an Application Server

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

                                              [Entry Fields]
  Server Name                                 imageappsrvr
  New Server Name                            [imageappsrvr]
  Start Script                               [/usr/sbin/cluster/even>
  Stop Script                                [/usr/sbin/cluster/even>


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
```

Figure 9. Application Server definition

The application server start script is:

```
/usr/sbin/cluster/events/utils/start_imagedemo -d /fs1 -a risc1_svc
```

The application server stop script is:

```
/usr/sbin/cluster/events/utils/stop_imagedemo -a risc1_svc
```

Appendix B, “Application server scripts” on page 273 shows these two shell scripts.

The monitored process is an executable and its full path name is:

```
/usr/sbin/cluster/demos/image/imserv
```

2.5.1.2 Resource Group definition

Figure 10 shows the configuration of the cascading resource group in the SMIT menu reachable with the `smit cm_cfg_res.select` fastpath. Cluster node risc1 is the high priority node and risc3 is the low priority node. As cluster resources we have defined only the service IP address of risc1, the /fs1 file system, the extvg volume group, and the imageappsrvr Application Server.

```
          Change/Show Resources/Attributes for a Resource Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
[TOP]                                                    [Entry Fields]
  Resource Group Name                                  resgrp1
  Node Relationship                                    cascading
  Participating Node Names                             risc1 risc3

  Service IP label                                    [risc1_svc]      +
  Filesystems                                         [/fs1]           +
  Filesystems Consistency Check                        fsck            +
  Filesystems Recovery Method                          sequential      +
  Filesystems/Directories to Export                   []               +
  Filesystems/Directories to NFS mount                []               +
  Network For NFS Mount                               []               +
  Volume Groups                                       [extvg]          +
  Concurrent Volume groups                            []               +
  Raw Disk PVIDs                                      []               +
  Connections Services                                []               +
  Fast Connect Services                               []               +
  Application Servers                                 [imageappsrvr]   +
  Highly Available Communication Links                []               +
  Miscellaneous Data                                  []

  Inactive Takeover Activated                          false           +
  Cascading Without Fallback Enabled                   false           +
  9333 Disk Fencing Activated                          false           +
  SSA Disk Fencing Activated                           false           +
  Filesystems mounted before IP configured             false           +


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
```

Figure 10. Resource Group definition

2.5.1.3 Identifying the correct process name to monitor

A very important step in configuring Process Monitor is to correctly identify the name of the process to be monitored. So we start ES and look for imserv in the process table. It is mandatory to use the `ps -ef` command and not `ps -el` to list the running processes.

Figure 11 shows both the `ps -ef` and the `ps -el` output to see the difference.

```
risc1# ps -ef | egrep "PID|imserv|16148"
UID    PID  PPID   C    STIME    TTY  TIME CMD
root 22984 17306   1 08:26:47      -  0:00 imserv 10.10.10.1

risc1# ps -el | egrep "imserv|PID"
     F S UID   PID  PPID   C PRI NI  ADDR    SZ   WCHAN  TTY  TIME CMD
240801 A   0 22984 17306   0  60 20 13853   900           -  0:00 imserv
```

Figure 11. Identifying the imserv process

The correct process name to use in monitoring is “imserv 10.10.10.1,” not “imserv.”

2.5.2 Simulating Failure Action “notify”

In this section we provide an example that specifies *notify* to a Failure Action.

2.5.2.1 Configuring Process Application Monitor

Figure 12 on page 23 shows the definition of Process Monitoring in the SMIT menu reachable with the `smit clappserv_to_monitor_by_process.select` fastpath.

```
              Add Process Application Monitor

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

                                                      [Entry Fields]
  Application Server Name                          imageappsrvr
* Processes to Monitor                             [imserv]
* Process Owner                                    [root]
  Instance Count                                   [1]                    #
* Stabilization Interval                           [20]                   #
* Restart Count                                    [2]                    #
  Restart Interval                                 []                     #
* Action on Application Failure                    [notify]               +
  Notify Method                                    [/usr/haes44/notify.sh]
  Cleanup Method                                   [/usr/sbin/cluster/even>
  Restart Method                                   [/usr/sbin/cluster/even>


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
```

Figure 12. Process Monitor definition

For a detailed description of the fields in this SMIT menu, see Section 2.3.1, “Configuration parameters” on page 8.

Application Server Name

This is the name of the Application Server configured in Section 2.5.1.1, “Application Server definition” on page 20.

Processes to Monitor

The name of the process to monitor discovered in Section 2.5.1.3, “Identifying the correct process name to monitor” on page 21.

Process Owner

The owner of the process to monitor discovered in Section 2.5.1.3, “Identifying the correct process name to monitor” on page 21.

Instance Count

It is 1 because we have only one instance of the process shown in Section 2.5.1.3, “Identifying the correct process name to monitor” on page 21.

Stabilization Interval

The definition of this field is the number of seconds that ES waits before starting to monitor the application. To find out the correct value, we activated the application multiple times and realized 20 seconds is the proper value.

Note

See Section 2.7.1, "Examining the Stabilization Interval" on page 56 for additional details.

Restart Count

We set 2 because we want ES to make two attempts to restart the failed application.

Restart Interval

We took the default value assigned by SMIT. The default value based on the following formula:

```
((Restart Count)*(Stabilization Interval)*1.1)
```

If you try to specify a value lower than this one, SMIT rejects it and uses the default one instead.

If you set Restart Count equal to 2, Stabilization Interval equal to 20, and leave the Restart Interval field empty, SMIT assigns the default value (44 seconds) to Restart Interval, as shown in Figure 13.

```
Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

clchappmon:  Setting the ODM path to /etc/objrepos
clchappmon warning:  The parameter "RESTART_INTERVAL" was not specified.
Will use 44.


F1=Help             F2=Refresh          F3=Cancel           Esc+6=Command
Esc+8=Image         Esc+9=Shell         Esc+0=Exit          /=Find
n=Find Next
```

Figure 13. The Restart Interval default value

Action on Application Failure

We leave the default value, *notify*, so that ES will inform us when the application fails.

Notify Method

We want ES to execute the `/usr/haes44/notify.sh` shell script when the application fails. We wrote this script as shown in Figure 14 on page 25.

```
#!/bin/ksh

# Shell Script executed to notify of the failure of the
# 'imserv' process

echo "\nThe imserv process has died at " >> /tmp/NOTIFY.IMSERV
/bin/date >> /tmp/NOTIFY.IMSERV

exit 0
```

Figure 14. The notify.sh script

Cleanup Method

We accept the default value assigned by SMIT, which is the Application Server stop script `/usr/sbin/cluster/events/utils/stop_imagedemo -a risc1_svc` as defined in Section 2.5.1.1, “Application Server definition” on page 20.

Restart Method

We accept the default value assigned by SMIT, which is the Application Server start script `/usr/sbin/cluster/events/utils/start_imagedemo -d /fs1 -a risc1_svc` as defined in Section 2.5.1.1, “Application Server definition” on page 20.

2.5.2.2 Starting the cluster

The configuration is now complete. The next step is to synchronize the cluster resources.

Note

The synchronization of the cluster resources does not copy over to the other node all the user-written shell scripts like the Notify, Cleanup, and Restart Methods. These files need to be transferred manually.

After ES starts, node risc1 acquires the resources, as shown in Figure 15.

```
risc1# clfindres
GroupName     Type       State    Location   Sticky Loc
---------   ----------   ------   --------   ----------
resgrp1     cascading      UP     risc1
```

Figure 15. Node risc1 acquires the resources

ES immediately realizes that Process Monitor has been configured for the imageappsrvr Application Server contained in the resgrp1 Resource Group, and starts to monitor the imserv process.

In Figure 16 we can see both the `run_clappmond` and the `clappmond` daemons running on node risc1 and monitoring the imserv process.

```
risc1# ps -ef | egrep "imserv|app"
    root 17622 20636   2 09:25:39  pts/6  0:00 egrep imserv|app
    root 22032 24500   0 09:25:15      -  0:00 /usr/es/sbin/cluster/clappmond im
ageappsrvr
    root 22984 17306   3 09:24:55      -  0:00 imserv 10.10.10.1
    root 24500 18598   0 09:25:15      -  0:00 run_clappmond -sport 1000 -result
_node risc1 -script_id 0 -command_id 12 -command imageappsrvr -environment ?CLUS
TER_VERSION=3??GS_NODEID=1??APPLICATION_SERVER=imageappsrvr??MISC_DATA=??RESOURC
E_GROUP=resgrp1??RESTART_METHOD=/usr/sbin/cluster/events/utils/start_imagedemo -
d /fs1 -a risc1_svc??CLEANUP_METHOD=/usr/sbin/cluster/events/utils/stop_imagedem
o -a risc1_svc??NOTIFY_METHOD=/usr/haes44/notify.sh??MONITOR_METHOD=??FAILURE_AC
TION=notify??RESTART_INTERVAL=44??RESTART_COUNT=2??STABILIZATION_INTERVAL=20??MO
NITOR_INTERVAL=0??INSTANCE_COUNT=1??PROCESS_OWNER=root??PROCESSES=imserv??MONITO
R_TYPE=process??HACMP_VERSION=__PE__??PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr
/bin/X11:/sbin??ODMDIR=/etc/es/objrepos??LC_FASTMSG=true??PING_IP_ADDRESS= ??LOC
ALNODEID=risc1??LOCALNODENAME=risc1??CM_CLUSTER_NAME=haes44??CM_CLUSTER_ID=6?
```

Figure 16. The run_clappmond and clappmond daemons monitoring the application

When ES started on node risc1, the events shown in Figure 17 were logged in the cluster history file.

```
risc1# tail -f /usr/es/sbin/cluster/history/cluster.07142000
Jul 14 09:24:03 EVENT START: node_up risc1
Jul 14 09:24:05 EVENT START: node_up_local
Jul 14 09:24:06 EVENT START: acquire_service_addr risc1_svc
Jul 14 09:24:18 EVENT START: acquire_aconn_service en0 ennetwork
Jul 14 09:24:19 EVENT START: swap_aconn_protocols en0 en1
Jul 14 09:24:20 EVENT COMPLETED: swap_aconn_protocols en0 en1
Jul 14 09:24:21 EVENT COMPLETED: acquire_aconn_service en0 ennetwork
Jul 14 09:24:21 EVENT COMPLETED: acquire_service_addr risc1_svc
Jul 14 09:24:22 EVENT START: get_disk_vg_fs /fs1 extvg
Jul 14 09:24:47 EVENT COMPLETED: get_disk_vg_fs /fs1 extvg
Jul 14 09:24:48 EVENT COMPLETED: node_up_local
Jul 14 09:24:48 EVENT COMPLETED: node_up risc1
Jul 14 09:24:49 EVENT START: node_up_complete risc1
Jul 14 09:24:51 EVENT START: node_up_local_complete
Jul 14 09:24:51 EVENT START: start_server imageappsrvr
Jul 14 09:24:52 EVENT COMPLETED: start_server imageappsrvr
Jul 14 09:24:54 EVENT COMPLETED: node_up_local_complete
Jul 14 09:24:54 EVENT COMPLETED: node_up_complete risc1
```

Figure 17. The history log file of node risc1

2.5.2.3 Simulating successful restart

At this point we simulate the failure of the monitored application by intentionally killing the imserv process:

```
risc1# kill -9 22984
```

ES recognizes the application failure, and the configured Notify Method, /usr/haes44/notify.sh described in Figure 14 on page 25 is executed, as shown in Figure 18.

```
risc1# cd /tmp
risc1# ls -l NOTIFY*
-rw-rw-rw-   1 root     system        62 Jul 14 09:37 NOTIFY.IMSERV

risc1# cat NOTIFY*
The imserv process has died at
Fri Jul 14 09:37:23 CDT 2000
```

Figure 18. Execution of the Notify Method shell script

The events shown in Figure 19 are logged in the history file of node risc1 when ES tries to restart the failed application.

```
risc1# tail -f /usr/es/sbin/cluster/history/cluster.07142000
Jul 14 09:37:22 EVENT START: server_restart risc1 12
Jul 14 09:37:24 EVENT COMPLETED: server_restart risc1 12
Jul 14 09:37:27 EVENT START: server_restart_complete risc1 12
Jul 14 09:37:29 EVENT COMPLETED: server_restart_complete risc1 12
```

Figure 19. The server_restart event

As we can see in Figure 20, the server_restart event has been successful in restarting the application. In fact, we see a new imserv process running, which is again monitored by ES.

```
risc1# ps -ef | egrep "imserv|app"
    root 15688 14092   0 09:37:30      -  0:00 imserv 10.10.10.1
    root 17420 18598   0 09:37:49      -  0:00 run_clappmond -sport 1000 -result
_node risc1 -script_id 0 -command_id 12 -command imageappsrvr -environment ?CLUS
TER_VERSION=3??GS_NODEID=1??APPLICATION_SERVER=imageappsrvr??MISC_DATA=??RESOURC
E_GROUP=resgrp1??RESTART_METHOD=/usr/sbin/cluster/events/utils/start_imagedemo -
d /fs1 -a risc1_svc??CLEANUP_METHOD=/usr/sbin/cluster/events/utils/stop_imagedem
o -a risc1_svc??NOTIFY_METHOD=/usr/haes44/notify.sh??MONITOR_METHOD=??FAILURE_AC
TION=notify??RESTART_INTERVAL=44??RESTART_COUNT=2??STABILIZATION_INTERVAL=20??MO
NITOR_INTERVAL=0??INSTANCE_COUNT=1??PROCESS_OWNER=root??PROCESSES=imserv??MONITO
R_TYPE=process??HACMP_VERSION=__PE__??PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr
/bin/X11:/sbin??ODMDIR=/etc/es/objrepos??LC_FASTMSG=true??PING_IP_ADDRESS= ??LOC
ALNODEID=risc1??LOCALNODENAME=risc1??CM_CLUSTER_NAME=haes44??CM_CLUSTER_ID=6?
    root 22042 20636   1 09:38:22  pts/6  0:00 egrep imserv|app
    root 23016 17420   0 09:37:50      -  0:00 /usr/es/sbin/cluster/clappmond im
ageappsrvr
```

Figure 20. New instance of the imserv process

To understand the execution of the server_restart event, it is necessary to look at the /tmp/hacmp.out log file of cluster node risc1. Figure 21 on page 29 shows only the relevant lines of the server_restart event.

```
/usr/sbin/cluster/events/utils/start_imagedemo[141]: 22984 Killed
start_imagedemo[146] exit 0

Jul 14 09:37:22 EVENT START: server_restart risc1 12

>>>>>>>>  omitted lines  <<<<<<<<<

notify_script=/usr/haes44/notify.sh
server_restart[114] [ -x /usr/haes44/notify.sh ]
server_restart[116] /usr/haes44/notify.sh

>>>>>>>>  omitted lines  <<<<<<<<<

cleanup=/usr/sbin/cluster/events/utils/stop_imagedemo -a risc1_svc
server_restart[123] server_restart[123] echo /usr/sbin/cluster/events/utils
/stop_imagedemo -a risc1_svc

>>>>>>>>  omitted lines  <<<<<<<<<<

Jul 14 09:37:24 EVENT COMPLETED: server_restart risc1 12
Jul 14 09:37:27 EVENT START: server_restart_complete risc1 12

>>>>>>>>  omitted lines  <<<<<<<<

restart_script=/usr/sbin/cluster/events/utils/start_imagedemo
server_restart_complete[112] [ -x /usr/sbin/cluster/events/utils/start_imag
edemo ]
server_restart_complete[114] /usr/sbin/cluster/events/utils/start_imagedemo
-d /fs1 -a risc1_svc

>>>>>>>>  omitted lines  <<<<<<<<

Jul 14 09:37:29 EVENT COMPLETED: server_restart_complete risc1 12
```

Figure 21. The /tmp/hacmp.out file of node risc1

By looking closely at this partial /tmp/hacmp.out log file, we can understand the occurrence of the events and also the order of execution of the different Method scripts. First of all, the imserv process is killed. When the server_restart event initiates, the Notify Method script `/usr/haes44/notify.sh` is executed. Then the Cleanup Method script `/usr/sbin/cluster/events/utils/stop_imagedemo -a risc1_svc` is run, and finally the Restart Method script `/usr/sbin/cluster/events/utils/start_imagedemo -d /fs1 -a risc1_svc` restarts the failed application on cluster node risc1.

2.5.2.4 Simulating unsuccessful restart

In the previous section we simulated ES successfully restarting an application. In this section, we simulate a case in which ES is unable to restart a failed application.

We again simulate the failure of the application by killing the imserv process. Figure 22 shows the entries logged in the history file of node risc1.

```
risc1# tail -f /usr/es/sbin/cluster/history/cluster.07142000
Jul 14 09:47:52 EVENT START: server_restart risc1 12
Jul 14 09:47:54 EVENT COMPLETED: server_restart risc1 12
Jul 14 09:47:58 EVENT START: server_restart_complete risc1 12
Jul 14 09:47:59 EVENT COMPLETED: server_restart_complete risc1 12

Jul 14 09:48:23 EVENT START: server_restart risc1 12
Jul 14 09:48:25 EVENT COMPLETED: server_restart risc1 12
Jul 14 09:48:29 EVENT START: server_restart_complete risc1 12
Jul 14 09:48:30 EVENT COMPLETED: server_restart_complete risc1 12

Jul 14 09:48:55 EVENT START: server_down risc1 12
Jul 14 09:48:56 EVENT COMPLETED: server_down risc1 12
Jul 14 09:48:59 EVENT START: server_down_complete risc1 12
Jul 14 09:49:00 EVENT COMPLETED: server_down_complete risc1 12
```

Figure 22. Events logged on node risc1

ES makes two attempts to restart the failed application, as can be seen by the server_restart event being executed twice. Both attempts are unsuccessful, so the server_down event is run to declare the application unavailable.

Note

Why does ES make two attempts? Because when we configured Process Monitoring, we set the parameter Restart Count equal to 2. See Figure 12 on page 23 for details.

At this point ES runs the Failure Action. When we configured Process Monitor (refer to Figure 12 on page 23), we specified “notify” as Failure Action, so the Notify Method shell script `/usr/haes44/notify.sh` is executed to inform the user that the application is no longer available, as shown in Figure 23 on page 31.

```
risc1# cat /tmp/NOTIFY.IMSERV

The imserv process has died at
Fri Jul 14 09:47:53 CDT 2000

The imserv process has died at
Fri Jul 14 09:48:24 CDT 2000

The imserv process has died at
Fri Jul 14 09:48:56 CDT 2000
#
```

Figure 23. Execution of the Notify Method script

Note

In this case ES does not perform a takeover to the standby node because when we configured Process Monitor we specified “notify” as the Failure Action. For ES to perform a takeover, it is necessary to specify “fallover” as Failure Action.

In such a situation, the output of the `clfindres` command can be very misleading, as shown in Figure 24 on page 31.

```
risc1# clfindres
GroupName     Type        State    Location   Sticky Loc
---------   ----------   ------   --------   ----------
resgrp1     cascading      UP     risc1
```

Figure 24. The clfindres command

You may think all cluster resources are still available on node risc1. In reality node risc1 now only owns the service IP label risc1_svc, the /fs1 filesystem, and the extvg volume group. The imageappsrvr application server is unavailable because ES was unable to restart it.

2.5.3 Simulating Failure Action “fallover”

This example specifies *fallover* as the Failure Action.

2.5.3.1 Configuring Process Application Monitor

This time, we configured the Process Monitor as shown in Figure 25.

```
             Add Process Application Monitor

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

                                                    [Entry Fields]
  Application Server Name                         imageappsrvr
* Processes to Monitor                            [imserv]
* Process Owner                                   [root]
  Instance Count                                  [1]                   #
* Stabilization Interval                          [15]                  #
* Restart Count                                   [1]                   #
  Restart Interval                                [16]                  #
* Action on Application Failure                   [fallover]            +
  Notify Method                                   [/usr/haes44/notify.sh]
  Cleanup Method                                  [/usr/sbin/cluster/even>
  Restart Method                                  [/usr/sbin/cluster/even>


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
```

Figure 25. Process Monitor definition

The differences are that we have specified 1 as Restart Count, and “fallover” as Failure Action instead of “notify.”

2.5.3.2 Starting the cluster

After starting the cluster, node risc1 acquires the resources and ES starts monitoring the application, as shown in Figure 26 on page 33.

```
risc1# ps -ef | egrep "imserv|app"
    root 15356 25804   2 10:14:07  pts/2  0:00 egrep imserv|app
    root 15800 18930   0 10:13:14      -  0:00 imserv 10.10.10.1
    root 19470 20184   0 10:13:35      -  0:00 run_clappmond -sport 1000 -result
_node risc1 -script_id 0 -command_id 12 -command imageappsrvr -environment ?CLUS
TER_VERSION=3??GS_NODEID=1??APPLICATION_SERVER=imageappsrvr??MISC_DATA=??RESOURC
E_GROUP=resgrp1??RESTART_METHOD=/usr/sbin/cluster/events/utils/start_imagedemo -
d /fs1 -a risc1_svc??CLEANUP_METHOD=/usr/sbin/cluster/events/utils/stop_imagedem
o -a risc1_svc??NOTIFY_METHOD=/usr/haes44/notify.sh??MONITOR_METHOD=??FAILURE_AC
TION=fallover??RESTART_INTERVAL=16??RESTART_COUNT=1??STABILIZATION_INTERVAL=15??
MONITOR_INTERVAL=0??INSTANCE_COUNT=1??PROCESS_OWNER=root??PROCESSES=imserv??MONI
TOR_TYPE=process??HACMP_VERSION=__PE__??PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/u
sr/bin/X11:/sbin??ODMDIR=/etc/es/objrepos??LC_FASTMSG=true??PING_IP_ADDRESS= ??L
OCALNODEID=risc1??LOCALNODENAME=risc1??CM_CLUSTER_NAME=haes44??CM_CLUSTER_ID=6?
    root 26282 19470   0 10:13:35      -  0:00 /usr/es/sbin/cluster/clappmond im
ageappsrvr
```

Figure 26. Monitoring of the imserv process

2.5.3.3 Simulating an application failure

We now intentionally kill the imserv process to simulate an application failure:

```
risc1# kill -9 15800
```

ES reacts to the failure by successfully restarting the application and then resumes the monitoring, as shown in Figure 27.

```
risc1# ps -ef | egrep "imserv|app"
    root 14102 20300   0 10:35:11      -  0:00 imserv 10.10.10.1
    root 14860 20636   1 10:35:38  pts/6  0:00 egrep imserv|app
    root 15108 20184   0 10:35:26      -  0:00 run_clappmond -sport 1000 -result
_node risc1 -script_id 0 -command_id 12 -command imageappsrvr -environment ?CLUS
TER_VERSION=3??GS_NODEID=1??APPLICATION_SERVER=imageappsrvr??MISC_DATA=??RESOURC
E_GROUP=resgrp1??RESTART_METHOD=/usr/sbin/cluster/events/utils/start_imagedemo -
d /fs1 -a risc1_svc??CLEANUP_METHOD=/usr/sbin/cluster/events/utils/stop_imagedem
o -a risc1_svc??NOTIFY_METHOD=/usr/haes44/notify.sh??MONITOR_METHOD=??FAILURE_AC
TION=fallover??RESTART_INTERVAL=16??RESTART_COUNT=1??STABILIZATION_INTERVAL=15??
MONITOR_INTERVAL=0??INSTANCE_COUNT=1??PROCESS_OWNER=root??PROCESSES=imserv??MONI
TOR_TYPE=process??HACMP_VERSION=__PE__??PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/u
sr/bin/X11:/sbin??ODMDIR=/etc/es/objrepos??LC_FASTMSG=true??PING_IP_ADDRESS= ??L
OCALNODEID=risc1??LOCALNODENAME=risc1??CM_CLUSTER_NAME=haes44??CM_CLUSTER_ID=6?
    root 19476 15108   0 10:35:26      -  0:00 /usr/es/sbin/cluster/clappmond im
ageappsrvr
```

Figure 27. Monitoring of the imserv process has resumed

So far, ES has performed exactly the same actions outlined in Section 2.5.2, “Simulating Failure Action “notify”” on page 22.

We again kill the imserv process:

```
risc1# kill -9 14102
```

Figure 28 shows the events logged in the history log file on node risc1.

```
risc1# tail -f /usr/es/sbin/cluster/history/cluster.07142000
Jul 14 10:45:39 EVENT START: server_restart risc1 12
Jul 14 10:45:41 EVENT COMPLETED: server_restart risc1 12
Jul 14 10:45:45 EVENT START: server_restart_complete risc1 12
Jul 14 10:45:46 EVENT COMPLETED: server_restart_complete risc1 12
Jul 14 10:46:05 EVENT START: rg_move risc1 1
Jul 14 10:46:07 EVENT START: node_down_local
Jul 14 10:46:08 EVENT START: stop_server imageappsrvr
Jul 14 10:46:10 EVENT COMPLETED: stop_server imageappsrvr
Jul 14 10:46:11 EVENT START: release_vg_fs /fs1 extvg
Jul 14 10:46:17 EVENT COMPLETED: release_vg_fs /fs1 extvg
Jul 14 10:46:18 EVENT START: release_service_addr risc1_svc
Jul 14 10:46:31 EVENT COMPLETED: release_service_addr risc1_svc
Jul 14 10:46:32 EVENT COMPLETED: node_down_local
Jul 14 10:46:33 EVENT COMPLETED: rg_move risc1 1
Jul 14 10:47:29 EVENT START: rg_move_complete risc1 1
Jul 14 10:47:31 EVENT START: node_up_remote_complete risc1
Jul 14 10:47:33 EVENT COMPLETED: node_up_remote_complete risc1
Jul 14 10:47:34 EVENT COMPLETED: rg_move_complete risc1 1
```

Figure 28. Events logged on node risc1

We can see that ES makes only one attempt to restart the application, in fact the server_restart event is executed once. ES makes one attempt because when we configured Process Monitor we specified a value of 1 in the field Restart Count, as you can see in Figure 25 on page 32. This time ES is unsuccessful in restarting the application. At this point the Failure Count parameter is equal to the Restart Count parameter (both are 1), so ES executes the Failure Action.

Note

The Restart Count parameter specifies the number of attempts ES will try to restart the failed application. The Failure Count keeps track of the number of times the application has failed.

We have specified “fallover” for a Failure Action; therefore the rg_move event is run. This event moves the resource group containing the failed application to the standby cluster node risc3.

If other resource groups are currently owned by node risc1, they remain on that node. The rg_move event *only* moves the resource group containing the failed application.

Figure 29 shows the events logged in to node risc3's cluster history file as it acquired the application. The name of the resource group containing the failed application server is resgrp1.

```
risc3# tail -f /usr/es/sbin/cluster/history/cluster.07142000
Jul 14 10:45:28 EVENT START: server_restart risc1 12
Jul 14 10:45:29 EVENT COMPLETED: server_restart risc1 12
Jul 14 10:45:31 EVENT START: server_restart_complete risc1 12
Jul 14 10:45:32 EVENT COMPLETED: server_restart_complete risc1 12
Jul 14 10:46:20 EVENT START: rg_move risc1 1
Jul 14 10:46:22 EVENT START: node_up_local
Jul 14 10:46:24 EVENT START: acquire_takeover_addr risc1_svc
Jul 14 10:46:41 EVENT COMPLETED: acquire_takeover_addr risc1_svc
Jul 14 10:46:41 EVENT START: get_disk_vg_fs /fs1 extvg
Jul 14 10:47:12 EVENT COMPLETED: get_disk_vg_fs /fs1 extvg
Jul 14 10:47:13 EVENT COMPLETED: node_up_local
Jul 14 10:47:14 EVENT COMPLETED: rg_move risc1 1
Jul 14 10:47:15 EVENT START: rg_move_complete risc1 1
Jul 14 10:47:19 EVENT START: node_up_remote_complete risc1
Jul 14 10:47:21 EVENT COMPLETED: node_up_remote_complete risc1
Jul 14 10:47:22 EVENT START: node_up_local_complete
Jul 14 10:47:23 EVENT START: start_server imageappsrvr
Jul 14 10:47:25 EVENT COMPLETED: start_server imageappsrvr
Jul 14 10:47:27 EVENT COMPLETED: node_up_local_complete
Jul 14 10:47:28 EVENT COMPLETED: rg_move_complete risc1 1
```

Figure 29. Node risc3 acquiring the resgrp1 Resource Group

Node risc3 simply acquires all the resources contained in the resgrp1 resource group, including the failed application as shown in Figure 30.

```
risc3# clfindres
GroupName     Type       State    Location   Sticky Loc
---------   ----------   ------   --------   ----------
resgrp1     cascading      UP     risc3
```

Figure 30. The clfindres command

In Figure 31 on page 36 we verify that the imserv process is now running on node risc3.

```
risc3# ps -ef | egrep "imserv|app"
    root 17376 13250   2 10:48:25  pts/2  0:00 egrep imserv|app
    root 17554 16380   0 10:47:43      -  0:00 run_clappmond -sport 1000 -result
_node risc3 -script_id 0 -command_id 12 -command imageappsrvr -environment ?CLUS
TER_VERSION=3??GS_NODEID=2??APPLICATION_SERVER=imageappsrvr??MISC_DATA=??RESOURC
E_GROUP=resgrp1??RESTART_METHOD=/usr/sbin/cluster/events/utils/start_imagedemo -
d /fs1 -a risc1_svc??CLEANUP_METHOD=/usr/sbin/cluster/events/utils/stop_imagedem
o -a risc1_svc??NOTIFY_METHOD=/usr/haes44/notify.sh??MONITOR_METHOD=??FAILURE_AC
TION=fallover??RESTART_INTERVAL=16??RESTART_COUNT=1??STABILIZATION_INTERVAL=15??
MONITOR_INTERVAL=0??INSTANCE_COUNT=1??PROCESS_OWNER=root??PROCESSES=imserv??MONI
TOR_TYPE=process??HACMP_VERSION=__PE__??PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/u
sr/bin/X11:/sbin??ODMDIR=/etc/es/objrepos??LC_FASTMSG=true??PING_IP_ADDRESS= ??L
OCALNODEID=risc3??LOCALNODENAME=risc3??CM_CLUSTER_NAME=haes44??CM_CLUSTER_ID=6?
    root 18600 11760   0 10:47:29      -  0:00 imserv 10.10.10.1
    root 19222 17554   0 10:47:44      -  0:00 /usr/es/sbin/cluster/clappmond im
ageappsrvr
```

Figure 31. The imserv process running on node risc3

Note

We want to emphasize one more time that ES has only moved the resgrp1 resource group to node risc3. This is *not* a node takeover; in fact, ES remains up and running on node risc1.

ES continues to monitor the application on node risc3 as it did before on node risc1. If we kill the imserv process it is restarted again as shown in Figure 32.

```
risc3# ps -ef | egrep "imserv|app"
    root 12576 13250   5 10:54:26  pts/2  0:00 egrep imserv|app
    root 16444 16380   0 10:53:58      -  0:00 run_clappmond -sport 1000 -result
_node risc3 -script_id 0 -command_id 12 -command imageappsrvr -environment ?CLUS
TER_VERSION=3??GS_NODEID=2??APPLICATION_SERVER=imageappsrvr??MISC_DATA=??RESOURC
E_GROUP=resgrp1??RESTART_METHOD=/usr/sbin/cluster/events/utils/start_imagedemo -
d /fs1 -a risc1_svc??CLEANUP_METHOD=/usr/sbin/cluster/events/utils/stop_imagedem
o -a risc1_svc??NOTIFY_METHOD=/usr/haes44/notify.sh??MONITOR_METHOD=??FAILURE_AC
TION=fallover??RESTART_INTERVAL=16??RESTART_COUNT=1??STABILIZATION_INTERVAL=15??
MONITOR_INTERVAL=0??INSTANCE_COUNT=1??PROCESS_OWNER=root??PROCESSES=imserv??MONI
TOR_TYPE=process??HACMP_VERSION=__PE__??PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/u
sr/bin/X11:/sbin??ODMDIR=/etc/es/objrepos??LC_FASTMSG=true??PING_IP_ADDRESS= ??L
OCALNODEID=risc3??LOCALNODENAME=risc3??CM_CLUSTER_NAME=haes44??CM_CLUSTER_ID=6?
    root 17188 13930   0 10:53:44      -  0:00 imserv 10.10.10.1
    root 17562 16444   0 10:53:58      -  0:00 /usr/es/sbin/cluster/clappmond im
ageappsrvr
```

Figure 32. The new instance of the imserv process

Figure 33 shows the events logged in the cluster history file on risc3 when the imserv process has been restarted.

```
risc3# tail -f /usr/es/sbin/cluster/history/cluster.07142000
Jul 14 10:53:33 EVENT START: server_restart risc3 12
Jul 14 10:53:36 EVENT COMPLETED: server_restart risc3 12
Jul 14 10:53:40 EVENT START: server_restart_complete risc3 12
Jul 14 10:53:42 EVENT COMPLETED: server_restart_complete risc3 12
```

Figure 33. The server_restart event

2.5.3.4 Moving Resource Group back

At any time we can move the resgrp1 resource group back to node risc1 by running the `cldare` command as shown in Figure 34. When this Dynamic Automatic Reconfiguration Event (DARE) completes, the monitoring of the application resumes on node risc1.

```
risc3# cldare -M resgrp1:default
risc3#
>>>>>>>  omitted lines  <<<<<<<
risc3#
risc3# clfindres
GroupName     Type       State    Location   Sticky Loc
---------   ----------   ------   --------   ----------
resgrp1     cascading      UP     risc1
```

Figure 34. The cldare command

2.5.4 Simulating Instance Count failure

In this section we concentrate on the Instance Count parameter through an example configuration of Process Monitor.

2.5.4.1 Configuring Process Application Monitor

Figure 35 on page 38 shows the Process Monitor configuration parameters. This is exactly the same as Figure 12 on page 23. For the purpose of this example, we must remember that the “Instance Count” value is equal to 1. This means we expect to have only one instance of the imserv process running at any time.

```
          Change/Show Process Application Monitor

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

                                                        [Entry Fields]
  Application Server Name                              imageappsrvr
* Processes to Monitor                                 [imserv]
* Process Owner                                        [root]
  Instance Count                                       [1]                  #
* Stabilization Interval                               [20]                 #
* Restart Count                                        [2]                  #
  Restart Interval                                     []                   #
* Action on Application Failure                        [notify]            +
  Notify Method                                        [/usr/haes44/notify.sh]
  Cleanup Method                                       [/usr/sbin/cluster/even>
  Restart Method                                       [/usr/sbin/cluster/even>


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
```

Figure 35. Process Monitor definition

2.5.4.2 Starting the cluster

After starting ES, we can see that the monitoring of the imserv process has begun, as shown in Figure 36.

```
risc1# ps -ef | egrep "imserv|appmon"
    root 18096 19316   0 16:07:08      -  0:00 imserv 10.10.10.1
    root 21350  8524   0 16:07:18      -  0:00 run_clappmond -sport 1000 -result
_node risc1 -script_id 0 -command_id 6 -command imageappsrvr -environment ?CLUST
ER_VERSION=3??GS_NODEID=1??APPLICATION_SERVER=imageappsrvr??MISC_DATA=??RESOURCE
_GROUP=resgrp1??RESTART_METHOD=/usr/sbin/cluster/events/utils/start_imagedemo -d
 /fs1 -a risc1_svc??CLEANUP_METHOD=/usr/sbin/cluster/events/utils/stop_imagedemo
 -a risc1_svc??NOTIFY_METHOD=/usr/haes44/notify.sh??MONITOR_METHOD=??FAILURE_ACT
ION=notify??RESTART_INTERVAL=11??RESTART_COUNT=1??STABILIZATION_INTERVAL=10??MON
ITOR_INTERVAL=0??INSTANCE_COUNT=1??PROCESS_OWNER=root??PROCESSES=imserv??MONITOR
_TYPE=process??HACMP_VERSION=__PE__??PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/
bin/X11:/sbin??ODMDIR=/etc/es/objrepos??LC_FASTMSG=true??PING_IP_ADDRESS= ??LOCA
LNODEID=risc1??LOCALNODENAME=risc1??CM_CLUSTER_NAME=haes44??CM_CLUSTER_ID=6?
    root 23478 21350   0 16:07:18      -  0:00 /usr/es/sbin/cluster/clappmond im
ageappsrvr
    root 23584 12574   3 16:08:54  pts/2  0:00 egrep imserv|appmon
```

Figure 36. The monitoring of the imserv process

2.5.4.3 Simulating application failure

To see the reaction of ES, we intentionally start a new instance of the imserv process as follows:

```
risc1# /usr/es/sbin/cluster/demos/image/imserv
```

As soon as ES realizes there are 2 running instances of the monitored process, it reacts as if it were an application failure, as shown in Figure 37.

```
risc1# tail -f /usr/sbin/cluster/history/cluster.08032000
Aug  3 16:15:03 EVENT START: server_restart risc1 6
Aug  3 16:15:04 EVENT COMPLETED: server_restart risc1 6
Aug  3 16:15:05 EVENT START: server_restart_complete risc1 6
Aug  3 16:15:07 EVENT COMPLETED: server_restart_complete risc1 6
```

Figure 37. Execution of the server_restart event

ES registers an application failure when the number of running instances of the monitored process is *less* than the Instance Count value, and also when the number is *greater*.

2.5.5 Configuring Process Application Monitor dynamically

In Section 2.5.2, “Simulating Failure Action “notify”” on page 22 and Section 2.5.3, “Simulating Failure Action “fallover”” on page 32, we have shown two examples of configuring Process Monitor when ES has not yet started. In this section instead, we will look at defining Process Monitor through a DARE while ES is running.

2.5.5.1 Configuring Process Application Monitor

With ES already running on all cluster nodes, we configure Process Monitor as shown in Figure 38 on page 40.

```
           Add Process Application Monitor

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

                                                    [Entry Fields]
  Application Server Name                         imageappsrvr
* Processes to Monitor                            [imserv]
* Process Owner                                   [root]
  Instance Count                                  [1]                     #
* Stabilization Interval                          [15]                    #
* Restart Count                                   [1]                     #
  Restart Interval                                [16]                    #
* Action on Application Failure                   [fallover]             +
  Notify Method                                   [/usr/haes44/notify.sh]
  Cleanup Method                                  [/usr/sbin/cluster/even>
  Restart Method                                  [/usr/sbin/cluster/even>



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
```

Figure 38. Process Monitor definition

2.5.5.2 Synchronizing the cluster resources

As soon as we synchronize the cluster resources, ES starts the DARE operation and executes the events shown in Figure 39.

```
risc1# tail -f /usr/es/sbin/cluster/history/cluster.07142000
Jul 14 14:23:54 EVENT START: reconfig_resource_release
Jul 14 14:24:09 EVENT COMPLETED: reconfig_resource_release
Jul 14 14:24:17 EVENT START: reconfig_resource_acquire
Jul 14 14:24:20 EVENT COMPLETED: reconfig_resource_acquire
Jul 14 14:24:24 EVENT START: reconfig_resource_complete
Jul 14 14:24:29 EVENT COMPLETED: reconfig_resource_complete
```

Figure 39. Events executed during the DARE

At the end of the DARE, both the `run_clappmond` and `clappmond` daemons are monitoring the imserv process on node risc1.

2.5.6 Configuring multiple processes monitoring

In this section we provide information about the relationship between the *Processes to Monitor* and the *Instance Count* fields.

2.5.6.1 Configuring Process Monitor

Figure 40 shows a case where we have specified two different processes to monitor, myappl1 and myappl2. They are binary modules we created. The field Instance Count has been set to two.

```
            Add Process Application Monitor

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

                                                        [Entry Fields]
  Application Server Name                             application2
* Processes to Monitor                               [myappl1 myappl2]
* Process Owner                                      [root]
  Instance Count                                     [2]                    #
* Stabilization Interval                             [10]                   #
* Restart Count                                      [1]                    #
  Restart Interval                                   []                   #
* Action on Application Failure                      [notify]              +
  Notify Method                                      [/tmp/notifyappl]
  Cleanup Method                                     [/tmp/stopappl]
  Restart Method                                     [/tmp/startappl]


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
```

Figure 40. Multiple processes to monitor

SMIT does not allow such a configuration. When more than one process is specified in the field Processes to Monitor, the Instance Count field *must* be set to one. As shown in Figure 41 on page 42, SMIT has reset the field to 1.

```
COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

claddappmon:  Setting the ODM path to /etc/objrepos
claddappmon warning:  Since multiple processes are to be monitored, the paramete
r "INSTANCE_COUNT" must be 1.
claddappmon warning:  The parameter "RESTART_INTERVAL" was not specified.  Will
use 11.




F1=Help             F2=Refresh          F3=Cancel           Esc+6=Command
Esc+8=Image         Esc+9=Shell         Esc+0=Exit          /=Find
n=Find Next
```

Figure 41. The Instance Count parameter reset to 1

2.5.7 Understanding Event Management

This section explains how Process Monitor relies on the Event Management (EM) component of the RS/6000 Cluster Technology (RSCT).

2.5.7.1 Configuring Process Application Monitor

As an example, we use the Process Monitor definition shown in Figure 42 on page 43. The process to monitor is called “myappl1”, which is a binary module we created. Then we defined 3 as the Instance Count because our Application Server starts a script that launches three instances of this process.

```
              Add Process Application Monitor

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

                                                      [Entry Fields]
* Application Server Name                           application1
* Processes to Monitor                             [myappl1]
* Process Owner                                    [root]
  Instance Count                                   [3]                   #
* Stabilization Interval                           [10]                  #
* Restart Count                                    [1]                   #
  Restart Interval                                 []                    #
* Action on Application Failure                    [notify]             +
  Notify Method                                    [/tmp/notifyappl1]
  Cleanup Method                                   [/tmp/stopappl1]
  Restart Method                                   [/tmp/startappl1 >


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
```

Figure 42. Process Monitor definition

2.5.7.2 Starting the cluster

After starting ES, we see the three running instances of the myappl1 process being monitored, as shown in Figure 43. The Process IDs (PIDs) are 16548, 22938, and 23494.

```
risc1# ps -ef | egrep "myappl1|appmon"
    root 11840 18130   0 14:28:08      -  0:00 /usr/es/sbin/cluster/clappmond ap
plication1
    root 13424     1   0 14:27:55      -  0:00 /tmp/myappl1 hello
    root 16548 14716   2 14:31:50  pts/0  0:00 egrep myappl1|appmon
    root 18130 23228   0 14:28:08      -  0:00 run_clappmond -sport 1000 -result
_node risc1 -script_id 0 -command_id 3 -command application1 -environment ?CLUST
ER_VERSION=3??GS_NODEID=1??APPLICATION_SERVER=application1??MISC_DATA=??RESOURCE
_GROUP=resgrpappl1??RESTART_METHOD=/tmp/startappl1 3??CLEANUP_METHOD=/tmp/
yoshi/stopappl1??NOTIFY_METHOD=/tmp/notifyappl1??MONITOR_METHOD=??FAILURE_
ACTION=notify??RESTART_INTERVAL=11??RESTART_COUNT=1??STABILIZATION_INTERVAL=10??
MONITOR_INTERVAL=0??INSTANCE_COUNT=3??PROCESS_OWNER=root??PROCESSES=myappl1??MON
ITOR_TYPE=process??HACMP_VERSION=__PE__??PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/
usr/bin/X11:/sbin??ODMDIR=/etc/es/objrepos??LC_FASTMSG=true??PING_IP_ADDRESS= ??
LOCALNODEID=risc1??LOCALNODENAME=risc1??CM_CLUSTER_NAME=haes44??CM_CLUSTER_ID=6?

    root 22938     1   0 14:27:55      -  0:00 /tmp/myappl1 hello
    root 23494     1   0 14:27:55      -  0:00 /tmp/myappl1 hello
```

Figure 43. The three running myappl1 processes

As explained in Figure 1 on page 7, Process Application Monitor registers a resource variable name with EM to monitor the life of a process. The `haemqvar` command is part of RSCT and allows you to display information about all the resource variable names. Figure 44 shows the syntax of this command. For additional information, refer to *HACMP V.4.3 AIX: Enhanced Scalability & Administration Guide, Vol. 2*.

```
risc1# cd /usr/sbin/rsct/bin
risc1# ./haemqvar -?
Usage:
        haemqvar [-S domain | -H domain] [ -c | -d | -i ] [ -f file ] [ -h ]
                 [ class  var  rsrcID  [ ... ] ]
                -S      Get definitions for the specified SP domain
                -H      Get definitions for the specified HACMP domain
                -c      Query current resource variable values
                -d      Query definitions, but output short form
                -i      Query instances of resource variable values
                -f      File containing lines of class var rsrcID
                -h      Only display this usage statement
                class   Name of resource variable class or quoted null string
                var     Name of resource variable or quoted null string
                rsrcID  Resource ID or "*"
```

Figure 44. The haemqvar command

We now use the `haemqvar` command to query which resource variable ES is using to monitor the myappl1 processes, as shown in Figure 45.

```
risc1# ./haemqvar -c -H haes44 | grep myappl1

1  IBM.PSSP.Prog.pcount  ProgName=myappl1;UserName=root;NodeNum=1  SBS: 3 0 "134
24,22938,23494"
```

Figure 45. The IBM.PSSP.Prog.pcount resource variable

The `-c` flag allows you to list all the resource variables being used by EM, while the `-H` flag is necessary to identify the cluster name, haes44 in this case. You can see that the resource variable called IBM.PSSP.Prog.pcount is used to monitor the myappl1 processes. You can also see the process owner is root and the PIDs of the monitored instances are 16548, 22938, and 23494. These PIDs are of course the same as those shown by the `ps -ef` command in Figure 43 on page 43.

The haemqvar command also provides a detailed description of each resource variable it supports. The following is the command to get an explanation of all resource variables:

```
risc1# ./haemqvar -H haes44 > /tmp/haemqvar.out
```

The /tmp/haemqvar.out file generated is very long. Figure 46 only shows the first few lines documenting the IBM.PSSP.Prog.pcount resource variable.

```
Variable Name:  IBM.PSSP.Prog.pcount
Value Type:     State
Data Type:      Structured Byte String
SBS Format:     CurPIDCount=long,PrevPIDCount=long,CurPIDList=cstring
Initial Value:  CurPIDCount=0,PrevPIDCount=0,CurPIDList=
Class:          IBM.PSSP.Prog
Locator:        NodeNum
Variable Description:
    A count of, and list of, processes running a program for a user.

    IBM.PSSP.Prog.pcount represents processes running a specified program
    on behalf of a specified user.  The resource variable's resource
    ID specifies the program name (ProgName), user name (UserName),
    and node number (NodeNum) of interest.  The ProgName value specifies
    the base name of the file containing the program.  The UserName value
    specifies the real user name, not the effective user name, associated
    with the process.  The NodeNum value specifies the node or nodes on
    which the processes are running.
```

Figure 46. Description of the IBM.PSSP.Prog.pcount resource variable

2.6 Custom Application Monitor examples

In this section we describe our experience of configuring and understanding Custom Monitor.

2.6.1 Common configuration

We use the same configuration described in Section 2.5.1, “Common configuration” on page 20.

2.6.2 Simulating successful restart

In this example, ES successfully restarts a failed application on the same node.

2.6.2.1 Configuring Custom Monitor

Figure 47 on page 46 shows the definition of Custom Monitor in the SMIT menu reachable with the `smit clappserv_to_custom_monitor.select` fastpath.

```
            Add Custom Application Monitor

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

                                                       [Entry Fields]
  Application Server Name                             imageappsrvr
* Monitor Method                                     [/usr/haes44/monmeth.sh]
  Monitor Interval                                   [20]                    #
  Hung Monitor Signal                                [9]                     #
* Stabilization Interval                             [15]                    #
  Restart Count                                      [1]                     #
  Restart Interval                                   []                      #
* Action on Application Failure                      [fallover]               +
  Notify Method                                      [/usr/haes44/notify.sh]
  Cleanup Method                                     [/usr/sbin/cluster/even>
  Restart Method                                     [/usr/sbin/cluster/even>


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
```

Figure 47. Custom Monitor definition

For a detailed description of each field of the SMIT menu, see Section 2.4.1, “Configuration parameters” on page 16.

Application Server Name

This is the name of the application server configured in Section 2.5.1.1, “Application Server definition” on page 20.

Monitor Method

This is the full-path name of the Monitor Method script, /usr/haes44/monmeth.sh, we created. This is shown in Figure 48.

```
risc1# cat /usr/haes44/monmeth.sh
#!/bin/ksh

# program that calls 'polling.sh' passing as an argument
# the name of the process to monitor, 'imserv' in our case

/usr/haes44/polling.sh imserv
```

Figure 48. The /usr/haes44/monmeth.sh script

The Monitor Method script invokes another user-written shell script called /usr/haes44/polling.sh we created. This passes an argument of imserv, the name of the process to monitor. The /usr/haes44/polling.sh script is shown in

Figure 49. We use this script to scan the process table looking for the imserv process. If it is running we exit with a return code of 0, whereas if it is not running we exit with a return code of 1. This return code is then passed back to the Monitor Method script /usr/haes44/monmeth.sh.

```
risc1# cat /usr/haes44/polling.sh
#!/bin/ksh

# Program that monitors if the 'imserv' process is running
#
# exit 0 means imserv is running
# exit 1 means imserv is not running

myself=$0

if [ $# -lt 1 ]
then
  echo "Specify the name of the process to monitor !\n"
  exit 0
fi

proc_to_monitor=$1

/bin/ps -ef | egrep $proc_to_monitor | grep -v clappmond | egrep -vq "egrep|$mys

if [ $? -eq 0 ]
then
  echo `date`
  echo "$proc_to_monitor is running !\n"
  exit 0
else
  echo `date`
  echo "$proc_to_monitor is not running !\n"
  exit 1
fi
```

Figure 49. The /usr/haes44/polling.sh script

Monitor Interval

We want the Monitor Method script /usr/haes44/monmeth.sh to run every 20 seconds. Also, if the execution time of the Monitor Method script is longer than 20 seconds, the script is delivered the signal specified in the field Hung Monitor Signal.

Hung Monitor Signal

If the Monitor Method script takes more than 20 seconds to complete, the signal 9 is delivered to the Monitor Method script.

Stabilization Interval

This field is the number of seconds that ES waits before starting to monitor the application. To find out the correct value, we activated the application several times and determined that 15 seconds is an appropriate value.

Restart Count

In case the application fails, we want ES to make one attempt to restart it.

Restart Interval

We left this field blank and accepted the default value assigned by SMIT. The default value based on the following formula:

```
((Restart Count)*(Stabilization Interval + Monitor Interval)*1.1)
```

If you try to specify a value lower than the default one, SMIT rejects it and uses the default instead.

If we set Restart Count equal to 2, Stabilization Interval equal to 15, and Monitor Interval equal to 20, SMIT assigns to Restart Interval the default value shown in Figure 50.

```
 COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

claddappmon:  Setting the ODM path to /etc/objrepos
claddappmon warning:  The parameter "RESTART_INTERVAL" was not specified.  Will
use 77.



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
```

Figure 50. The Restart Interval default value

Action on Application Failure

We specified fallover so that ES will move the Resource Group containing the failed Application Server to the next highest priority node for this Resource Group.

Notify Method

We want ES to execute the /usr/haes44/notify.sh shell script when the application fails. This script is shown in Figure 14 on page 25.

Cleanup Method

We accept the default value assigned by SMIT, which is the Application Server stop script /usr/sbin/cluster/events/utils/stop_imagedemo -a risc1_svc as defined in Section 2.5.1.1, “Application Server definition” on page 20.

Restart Method

We accept the default value assigned by SMIT, which is the Application Server start script /usr/sbin/cluster/events/utils/start_imagedemo -d /fs1 -a risc1_svc as defined in Section 2.5.1.1, “Application Server definition” on page 20.

2.6.2.2 Starting the cluster

The configuration is now complete, the last step is to synchronize the cluster resources.

Note

The synchronization of the cluster resources does not copy over to the other cluster nodes all the user-written shell scripts like the Monitor, Notify, Cleanup, and Restart Methods. These files need to be transferred manually. However, the `clverify` command reports an error if these scripts do not exist or are not executable on all cluster nodes.

When ES starts, the events shown in Figure 51 on page 50 are logged in the cluster history file of node risc1.

```
risc1# tail -f /usr/sbin/cluster/history/cluster.08012000
Aug  1 09:26:46 EVENT START: node_up risc1
Aug  1 09:26:48 EVENT START: node_up_local
Aug  1 09:26:49 EVENT START: acquire_service_addr risc1_svc
Aug  1 09:27:02 EVENT START: acquire_aconn_service en0 ennetwork
Aug  1 09:27:03 EVENT START: swap_aconn_protocols en0 en1
Aug  1 09:27:03 EVENT COMPLETED: swap_aconn_protocols en0 en1
Aug  1 09:27:04 EVENT COMPLETED: acquire_aconn_service en0 ennetwork
Aug  1 09:27:04 EVENT COMPLETED: acquire_service_addr risc1_svc
Aug  1 09:27:05 EVENT START: get_disk_vg_fs /fs1 extvg
Aug  1 09:27:29 EVENT COMPLETED: get_disk_vg_fs /fs1 extvg
Aug  1 09:27:29 EVENT COMPLETED: node_up_local
Aug  1 09:27:30 EVENT COMPLETED: node_up risc1
Aug  1 09:27:31 EVENT START: node_up_complete risc1
Aug  1 09:27:33 EVENT START: node_up_local_complete
Aug  1 09:27:33 EVENT START: start_server imageappsrvr
Aug  1 09:27:34 EVENT COMPLETED: start_server imageappsrvr
Aug  1 09:27:36 EVENT COMPLETED: node_up_local_complete
Aug  1 09:27:37 EVENT COMPLETED: node_up_complete risc1
```

Figure 51. The events logged at cluster startup

Monitoring of the imserv process begins as shown in Figure 52.

```
risc1# ps -ef | egrep "imserv|app"
    root 13926 14716   3 09:30:18  pts/0  0:00 egrep imserv|app
    root 14094 20112   0 09:27:51      -  0:00 run_clappmond -sport 1000 -result
_node risc1 -script_id 0 -command_id 12 -command imageappsrvr -environment ?CLUS
TER_VERSION=3??GS_NODEID=1??APPLICATION_SERVER=imageappsrvr??MISC_DATA=??RESOURC
E_GROUP=resgrp1??RESTART_METHOD=/usr/sbin/cluster/events/utils/start_imagedemo -
d /fs1 -a risc1_svc??CLEANUP_METHOD=/usr/sbin/cluster/events/utils/stop_imagedem
o -a risc1_svc??NOTIFY_METHOD=/usr/haes44/notify.sh??MONITOR_METHOD=/usr/haes44/
monmeth.sh??FAILURE_ACTION=fallover??RESTART_INTERVAL=77??RESTART_COUNT=1??STABILI
ZATION_INTERVAL=15??MONITOR_INTERVAL=20??INSTANCE_COUNT=0??PROCESS_OWNER=??PROCE
SSES=??MONITOR_TYPE=user??HACMP_VERSION=__PE__??PATH=/usr/bin:/etc:/usr/sbin:/us
r/ucb:/usr/bin/X11:/sbin??ODMDIR=/etc/es/objrepos??LC_FASTMSG=true??PING_IP_ADDR
ESS= ??LOCALNODEID=risc1??LOCALNODENAME=risc1??CM_CLUSTER_NAME=haes44??CM_CLUSTE
R_ID=6?
    root 14988 14094   0 09:27:52      -  0:00 /usr/es/sbin/cluster/clappmond im
ageappsrvr
    root 21406 22282   0 09:27:37      -  0:00 imserv 10.10.10.1
```

Figure 52. Monitoring of the imserv process

2.6.2.3 Simulating an application failure

We now intentionally kill the imserv process to simulate an application failure as follows:

```
risc1# kill -9 21406
```

ES recognizes the application failure immediately. The configured Notify Method script `/usr/haes44/notify.sh` is executed, as shown in Figure 53.

```
risc1# cd /tmp
risc1# ls -l NOTIFY.IMSERV
-rw-rw-rw-   1 root     system        62 Aug 01 09:50 NOTIFY.IMSERV
risc1# cat NOTIFY.IMSERV

The imserv process has died at
Tue Aug  1 09:50:41 CDT 2000
```

Figure 53. The execution of the Notify Method script

The events shown in Figure 54 are logged in the cluster history file of node risc1 when ES tries to restart the imserv process.

```
risc1# tail -f /usr/sbin/cluster/history/cluster.08012000
Aug  1 09:50:40 EVENT START: server_restart risc1 12
Aug  1 09:50:42 EVENT COMPLETED: server_restart risc1 12
Aug  1 09:50:46 EVENT START: server_restart_complete risc1 12
Aug  1 09:50:47 EVENT COMPLETED: server_restart_complete risc1 12
```

Figure 54. The restart of the failed application

As we can see in Figure 55, the server_restart event successfully started a new instance of the imserv process, which is again monitored by ES.

```
risc1# ps -ef | egrep "imserv|app"
    root 11926 14034   0 09:50:48      -  0:00 imserv 10.10.10.1
    root 14376 20112   0 09:51:03      -  0:00 run_clappmond -sport 1000 -result
_node risc1 -script_id 0 -command_id 12 -command imageappsrvr -environment ?CLUS
TER_VERSION=3??GS_NODEID=1??APPLICATION_SERVER=imageappsrvr??MISC_DATA=??RESOURC
E_GROUP=resgrp1??RESTART_METHOD=/usr/sbin/cluster/events/utils/start_imagedemo -
d /fs1 -a risc1_svc??CLEANUP_METHOD=/usr/sbin/cluster/events/utils/stop_imagedem
o -a risc1_svc??NOTIFY_METHOD=/usr/haes44/notify.sh??MONITOR_METHOD=/usr/haes44/
monmeth.sh??FAILURE_ACTION=notify??RESTART_INTERVAL=77??RESTART_COUNT=2??STABILI
ZATION_INTERVAL=15??MONITOR_INTERVAL=20??INSTANCE_COUNT=0??PROCESS_OWNER=??PROCE
SSES=??MONITOR_TYPE=user??HACMP_VERSION=__PE__??PATH=/usr/bin:/etc:/usr/sbin:/us
r/ucb:/usr/bin/X11:/sbin??ODMDIR=/etc/es/objrepos??LC_FASTMSG=true??PING_IP_ADDR
ESS= ??LOCALNODEID=risc1??LOCALNODENAME=risc1??CM_CLUSTER_NAME=haes44??CM_CLUSTE
R_ID=6?
    root 19866 14376   0 09:51:03      -  0:00 /usr/es/sbin/cluster/clappmond im
ageappsrvr
    root 22290 14716   1 09:52:49  pts/0  0:00 egrep imserv|app
```

Figure 55. The imserv process successfully restarted

To better understand the execution of the server_restart event it is necessary to look at the /tmp/hacmp.out log file of node risc1. Figure 56 only shows the relevant lines.

```
/usr/sbin/cluster/events/utils/start_imagedemo[142]: 21406 Killed
start_imagedemo[147] exit 0

Aug  1 09:50:40 EVENT START: server_restart risc1 12

>>>>>>  omitted lines  <<<<<<<

notify_script=/usr/haes44/notify.sh
server_restart[114] [ -x /usr/haes44/notify.sh ]
server_restart[116] /usr/haes44/notify.sh

>>>>>>  omitted lines  <<<<<<<

cleanup_script=/usr/sbin/cluster/events/utils/stop_imagedemo
server_restart[125] [ -x /usr/sbin/cluster/events/utils/stop_imagedemo ]
server_restart[127] /usr/sbin/cluster/events/utils/stop_imagedemo -a risc1_svc

>>>>>>  omitted lines  <<<<<<<

Aug  1 09:50:42 EVENT COMPLETED: server_restart risc1 12
Aug  1 09:50:46 EVENT START: server_restart_complete risc1 12

>>>>>>  omitted lines  <<<<<<<

restart_script=/usr/sbin/cluster/events/utils/start_imagedemo
server_restart_complete[112] [ -x /usr/sbin/cluster/events/utils/start_imagedemo
 ]
server_restart_complete[114] /usr/sbin/cluster/events/utils/start_imagedemo -d /
fs1 -a risc1_svc

>>>>>>  omitted lines  <<<<<<<

Aug  1 09:50:47 EVENT COMPLETED: server_restart_complete risc1 12
```

Figure 56. The /tmp/hacmp.out log file of node risc1

By looking closely at this partial /tmp/hacmp.out file, we can understand the order of execution of the different Method scripts. When the server_restart event begins, the Notify Method script `/usr/haes44/notify.sh` is run to inform you of the application failure, as shown in Figure 53. Then ES runs the Cleanup Method script `/usr/sbin/cluster/events/utils/stop_imagedemo -a risc1_svc`, then finally executes the Restart Method script `/usr/sbin/cluster/events/utils/start_imagedemo -d /fs1 -a risc1_svc` to restart the imserv process.

2.6.3 Simulating unsuccessful restart

In Section 2.6.2, “Simulating successful restart” on page 45, ES successfully restarted a failed application. In this example ES is unsuccessful.

2.6.3.1 Configuring Custom Application Monitor

The configuration is exactly the same as shown in Figure 47 on page 46.

2.6.3.2 Starting the cluster

After starting ES, the monitoring of the imserv process begins on node risc1, as shown in Figure 57.

```
risc1# ps -ef | egrep "imserv|app"
    root 12250 14716   2 08:20:55  pts/0  0:00 egrep imserv|app
    root 14338  7454   0 08:20:00      -  0:00 run_clappmond -sport 1000 -result
_node risc1 -script_id 0 -command_id 12 -command imageappsrvr -environment ?CLUS
TER_VERSION=3??GS_NODEID=1??APPLICATION_SERVER=imageappsrvr??MISC_DATA=??RESOURC
E_GROUP=resgrp1??RESTART_METHOD=/usr/sbin/cluster/events/utils/start_imagedemo -
d /fs1 -a risc1_svc??CLEANUP_METHOD=/usr/sbin/cluster/events/utils/stop_imagedem
o -a risc1_svc??NOTIFY_METHOD=/usr/haes44/notify.sh??MONITOR_METHOD=/usr/haes44/
monmeth.sh??FAILURE_ACTION=fallover??RESTART_INTERVAL=77??RESTART_COUNT=1??STABI
LIZATION_INTERVAL=15??MONITOR_INTERVAL=20??INSTANCE_COUNT=0??PROCESS_OWNER=??PRO
CESSES=??MONITOR_TYPE=user??HACMP_VERSION=__PE__??PATH=/usr/bin:/etc:/usr/sbin:/
usr/ucb:/usr/bin/X11:/sbin??ODMDIR=/etc/es/objrepos??LC_FASTMSG=true??PING_IP_AD
DRESS= ??LOCALNODEID=risc1??LOCALNODENAME=risc1??CM_CLUSTER_NAME=haes44??CM_CLUS
TER_ID=6?
    root 22474 14338   0 08:20:00      -  0:00 /usr/es/sbin/cluster/clappmond im
ageappsrvr
    root 23200 16578   0 08:19:43      -  0:00 imserv 10.10.10.1
```

Figure 57. Monitoring of imserv has begun

2.6.3.3 Simulating the application failure

To force a failure to restart the application, we killed imserv daemon twice using the `kill` command.

ES immediately recognizes the application failure and executes the events shown in Figure 58 on page 54. The server_restart event is run once, then the rg_move event is executed.

Note

The server_restart event is executed only one time because we specified the value of 1 as the Restart Count. Because the attempt to restart the application was unsuccessful, ES runs the rg_move event because we specified fallover as Failure Action. Refer to Figure 47 on page 46 for details.

```
risc1# tail -f /usr/sbin/cluster/history/cluster.08032000
Aug  3 08:27:12 EVENT START: server_restart risc1 12
Aug  3 08:27:14 EVENT COMPLETED: server_restart risc1 12
Aug  3 08:27:17 EVENT START: server_restart_complete risc1 12
Aug  3 08:27:19 EVENT COMPLETED: server_restart_complete risc1 12
Aug  3 08:27:38 EVENT START: rg_move risc1 1
Aug  3 08:27:40 EVENT START: node_down_local
Aug  3 08:27:41 EVENT START: stop_server imageappsrvr
Aug  3 08:27:43 EVENT COMPLETED: stop_server imageappsrvr
Aug  3 08:27:44 EVENT START: release_vg_fs /fs1 extvg
Aug  3 08:27:51 EVENT COMPLETED: release_vg_fs /fs1 extvg
Aug  3 08:27:52 EVENT START: release_service_addr risc1_svc
Aug  3 08:28:03 EVENT COMPLETED: release_service_addr risc1_svc
Aug  3 08:28:05 EVENT COMPLETED: node_down_local
Aug  3 08:28:05 EVENT COMPLETED: rg_move risc1 1
Aug  3 08:29:04 EVENT START: rg_move_complete risc1 1
Aug  3 08:29:06 EVENT START: node_up_remote_complete risc1
Aug  3 08:29:08 EVENT COMPLETED: node_up_remote_complete risc1
Aug  3 08:29:08 EVENT COMPLETED: rg_move_complete risc1 1
```

Figure 58. Events logged on risc1

As a consequence of the rg_move event, the resource group containing the failed application is moved to the standby node risc3. Figure 59 on page 55 shows the events logged in the cluster history file of node risc3 when ES tries to restart the imserv process.

```
risc3# tail -f /usr/sbin/cluster/history/cluster.08032000
Aug  3 08:27:19 EVENT START: server_restart risc1 12
Aug  3 08:27:20 EVENT COMPLETED: server_restart risc1 12
Aug  3 08:27:22 EVENT START: server_restart_complete risc1 12
Aug  3 08:27:23 EVENT COMPLETED: server_restart_complete risc1 12
Aug  3 08:28:11 EVENT START: rg_move risc1 1
Aug  3 08:28:14 EVENT START: node_up_local
Aug  3 08:28:16 EVENT START: acquire_takeover_addr risc1_svc
Aug  3 08:28:33 EVENT COMPLETED: acquire_takeover_addr risc1_svc
Aug  3 08:28:35 EVENT START: get_disk_vg_fs /fs1 extvg
Aug  3 08:29:05 EVENT COMPLETED: get_disk_vg_fs /fs1 extvg
Aug  3 08:29:06 EVENT COMPLETED: node_up_local
Aug  3 08:29:07 EVENT COMPLETED: rg_move risc1 1
Aug  3 08:29:09 EVENT START: rg_move_complete risc1 1
Aug  3 08:29:13 EVENT START: node_up_remote_complete risc1
Aug  3 08:29:15 EVENT COMPLETED: node_up_remote_complete risc1
Aug  3 08:29:16 EVENT START: node_up_local_complete
Aug  3 08:29:17 EVENT START: start_server imageappsrvr
```

Figure 59. Events logged on risc3

This time ES is successful in restarting the failed application on node risc3, and monitoring is resumed as shown in Figure 60.

```
risc3# ps -ef | egrep "imserv|app"
    root  7924 16366   3 08:33:17  pts/1  0:00 egrep imserv|app
    root 11192 13100   0 08:29:38      -  0:00 /usr/es/sbin/cluster/clappmond im
ageappsrvr
    root 12896 20320   0 08:29:24      -  0:00 imserv 10.10.10.1
    root 13100 17916   0 08:29:38      -  0:00 run_clappmond -sport 1000 -result
_node risc3 -script_id 0 -command_id 12 -command imageappsrvr -environment ?CLUS
TER_VERSION=3??GS_NODEID=2??APPLICATION_SERVER=imageappsrvr??MISC_DATA=??RESOURC
E_GROUP=resgrp1??RESTART_METHOD=/usr/sbin/cluster/events/utils/start_imagedemo -
d /fs1 -a risc1_svc??CLEANUP_METHOD=/usr/sbin/cluster/events/utils/stop_imagedem
o -a risc1_svc??NOTIFY_METHOD=/usr/haes44/notify.sh??MONITOR_METHOD=/usr/haes44/
monmeth.sh??FAILURE_ACTION=fallover??RESTART_INTERVAL=77??RESTART_COUNT=1??STABI
LIZATION_INTERVAL=15??MONITOR_INTERVAL=20??INSTANCE_COUNT=0??PROCESS_OWNER=??PRO
CESSES=??MONITOR_TYPE=user??HACMP_VERSION=__PE__??PATH=/usr/bin:/etc:/usr/sbin:/
usr/ucb:/usr/bin/X11:/sbin??ODMDIR=/etc/es/objrepos??LC_FASTMSG=true??PING_IP_AD
DRESS= ??LOCALNODEID=risc3??LOCALNODENAME=risc3??CM_CLUSTER_NAME=haes44??CM_CLUS
TER_ID=6?
```

Figure 60. The imserv process running on node risc3

2.7 Other examples

This section provides examples that apply to both Process and Custom Monitors.

Note

The examples shown here use Process Monitor. However the concepts of these examples apply to both Process and Custom Monitors. The configuration steps are also the same.

2.7.1 Examining the Stabilization Interval

In this section we want to clarify the meaning of the Stabilization Interval with an example.

We use the same configuration as described in Section 2.5.1, “Common configuration” on page 20. As shown in Figure 61, we modified the start script `/usr/sbin/cluster/events/utils/start_imagedemo` by adding the following line:

```
/bin/date >> /tmp/imserv.date
```

in order to save the exact time when the imserv process is started.

```
if [ -z "$LINE" ]
then
        /bin/date >> /tmp/imserv.date
        imserv $SERVICE_ADDRESS 2>&1 > /tmp/imserv
else
        cl_echo 51 "$PROGNAME: imserv already running." $PROGNAME
fi
```

Figure 61. The start_imagedemo script

2.7.1.1 Configuring Process Monitor

Figure 62 on page 57 shows the definition of our Process Monitor. We have specified 60 seconds as Stabilization Interval.

```
              Change/Show Process Application Monitor

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

                                                    [Entry Fields]
  Application Server Name                          imageappsrvr
* Processes to Monitor                             [imserv]
* Process Owner                                    [root]
  Instance Count                                   [1]                    #
* Stabilization Interval                           [60]                   #
* Restart Count                                    [1]                    #
  Restart Interval                                 [66]                   #
* Action on Application Failure                    [fallover]             +
  Notify Method                                    [/usr/haes44/notify.sh]
  Cleanup Method                                   [/usr/sbin/cluster/even>
  Restart Method                                   [/usr/sbin/cluster/even>


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
```

Figure 62. The Process Application Monitor definition

2.7.1.2 Starting the cluster

After starting ES, the events shown in Figure 63 are executed. The start_server event is responsible to start the imserv process, and it completes execution at 15:16:54.

```
risc1# tail -f /usr/es/sbin/cluster/history/cluster.07142000
Jul 14 15:16:52 EVENT START: node_up_local_complete
Jul 14 15:16:53 EVENT START: start_server imageappsrvr
Jul 14 15:16:54 EVENT COMPLETED: start_server imageappsrvr
Jul 14 15:16:55 EVENT COMPLETED: node_up_local_complete
```

Figure 63. The start_server event

By looking at Figure 64 we see that the imserv process was started at 15:16:56.

```
risc1# cat /tmp/imserv.date
Fri Jul 14 15:16:56 CDT 2000
```

Figure 64. The /tmp/imserv.date file

Note

The slight difference in time between the completion of the start_server event and the time shown in the /tmp/imserv.date file is due to the fact that the imserv process is started in the background.

When ES starts to monitor an application, the `clappmond` daemon is invoked and creates its own log file called /tmp/clappmond.imageappsrvr.log. Figure 65 shows the first few lines of this file.

```
risc1# head /tmp/clappmond.imageappsrvr.log
Jul 14 15:17:57: clappmond starting on "imageappsrvr"
Jul 14 15:17:57: MONITOR_TYPE="process"
Jul 14 15:17:57: PROCESSES="imserv"
Jul 14 15:17:57: PROCESS_OWNER="root"
```

Figure 65. The /tmp/clappmond.imageappsrvr.log file

We can see that the daemon was started at 15:17:57. If we compare this value with the one we found in Figure 64 on page 57, we see that approximately 60 seconds have elapsed from the time the imserv process was started to when ES starts the monitoring. These 60 seconds are the stabilization interval we specified in the process application monitor configuration shown in Figure 62 on page 57.

2.7.2 Suspending and resuming Application Monitoring

After configuring and activating Application Monitoring, at some point there may be the need to momentarily suspend the monitoring. For example, suspending could become useful when the application must be stopped for maintenance reasons, but the entire cluster needs to remain running. At any time we can resume the monitoring.

2.7.2.1 Configuring Process Monitor

We use as an example the Process Monitor configuration shown in Figure 66 on page 59.

```
              Change/Show Process Application Monitor

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

                                              [Entry Fields]
  Application Server Name                        imageappsrvr
* Processes to Monitor                          [imserv]
* Process Owner                                 [root]
  Instance Count                                [1]                 #
* Stabilization Interval                        [15]                #
* Restart Count                                 [1]                 #
  Restart Interval                              [16]                #
* Action on Application Failure                 [fallover]          +
  Notify Method                                 [/usr/haes44/notify.sh]
  Cleanup Method                                [/usr/sbin/cluster/even>
  Restart Method                                [/usr/sbin/cluster/even>


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
```

Figure 66. Process Monitor definition

2.7.2.2 Starting the cluster

After starting ES on all cluster nodes, node risc1 has taken control of the application server and monitoring is active, as shown in Figure 67. Both the `run_clappmond` and `clappmond` daemons are running.

```
risc1# ps -ef | egrep "imserv|app"
    root 11832 25804   2 11:39:53  pts/2  0:00 egrep imserv|app
    root 19656 21712   0 11:31:11      -  0:00 run_clappmond -sport 1000 -result
_node risc1 -script_id 0 -command_id 12 -command imageappsrvr -environment ?CLUS
TER_VERSION=3??GS_NODEID=1??APPLICATION_SERVER=imageappsrvr??MISC_DATA=??RESOURC
E_GROUP=resgrp1??RESTART_METHOD=/usr/sbin/cluster/events/utils/start_imagedemo -
d /fs1 -a risc1_svc??CLEANUP_METHOD=/usr/sbin/cluster/events/utils/stop_imagedem
o -a risc1_svc??NOTIFY_METHOD=/usr/haes44/notify.sh??MONITOR_METHOD=??FAILURE_AC
TION=fallover??RESTART_INTERVAL=16??RESTART_COUNT=1??STABILIZATION_INTERVAL=15??
MONITOR_INTERVAL=0??INSTANCE_COUNT=1??PROCESS_OWNER=root??PROCESSES=imserv??MONI
TOR_TYPE=process??HACMP_VERSION=__PE__??PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/u
sr/bin/X11:/sbin??ODMDIR=/etc/es/objrepos??LC_FASTMSG=true??PING_IP_ADDRESS= ??L
OCALNODEID=risc1??LOCALNODENAME=risc1??CM_CLUSTER_NAME=haes44??CM_CLUSTER_ID=6?
    root 22422 20404   0 11:30:56      -  0:00 imserv 10.10.10.1
    root 26122 19656   0 11:31:11      -  0:00 /usr/es/sbin/cluster/clappmond im
ageappsrvr
```

Figure 67. Monitoring of the imserv process

Note

At this moment the only way to discover if Application Monitoring is active is to scan the process table and make sure both `run_clappmond` and `clappmond` are running.

2.7.2.3 Suspending Application Monitoring

At this point we want to temporarily suspend the monitoring of our application. The suspend operation is performed by the Cluster Single Point of Control (C-SPOC) SMIT menus, and like all C-SPOC commands. This operation can be executed from any cluster node, not necessarily the node where the application is running.

First of all, we must select which application server to suspend. In our case we only have one, imageappsrvr, as shown in Figure 68. This SMIT menu is reachable with the `smit cm_suspend_resume_menu` fastpath.

```
                 Suspend/Resume Application Monitoring

Move cursor to desired item and press Enter.

  Suspend Application Monitoring
  Resume Application Monitoring

  lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
  x                      Application Monitor to Suspend                      x
  x                                                                          x
  x Move cursor to desired item and press Enter.                             x
  x                                                                          x
  x   imageappsrvr                                                           x
  x                                                                          x
  x F1=Help                 F2=Refresh              F3=Cancel                x
  x Esc+8=Image             Esc+0=Exit              Enter=Do                 x
F1x /=Find                  n=Find Next                                      x
Esmqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

Figure 68. Select the imageappsrvr Application Server

When the command completes, SMIT shows the output shown in Figure 69 on page 61.

```
                                COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

executing clRMupdate
clRMupdate: checking operation suspend_appmon
clRMupdate: found operation in table
clRMupdate: operating on 12
clRMupdate: sending operation to resource manager
clRMupdate completed successfully



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
```

Figure 69. The suspend operation completes

In order to find out if monitoring is active or not, look for `run_clappmond` and `clappmond` in the process table. After the suspend we only find the application server as seen in Figure 70:

```
risc1# ps -ef | egrep "imserv|app"
    root 19662 25804   4 11:46:50  pts/2  0:00 egrep imserv|app
    root 22422 20404   0 11:30:56      -  0:00 imserv 10.10.10.1
risc1#
```

Figure 70. Process Table after suspend operation

2.7.2.4 Resuming Application Monitoring

After suspending the monitoring, it can be resumed at any time. Resuming means ES starts to monitor the life of the application server again. Resuming is performed from the SMIT menu shown in Figure 71 on page 62.

```
          Suspend/Resume Application Monitoring

Move cursor to desired item and press Enter.

  Suspend Application Monitoring
  Resume Application Monitoring


  lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
  x                     Application Monitor to Resume                        x
  x                                                                          x
  x Move cursor to desired item and press Enter.                             x
  x                                                                          x
  x   imageappsrvr                                                           x
  x                                                                          x
  x F1=Help                F2=Refresh               F3=Cancel                x
  x Esc+8=Image            Esc+0=Exit               Enter=Do                 x
F1x /=Find                 n=Find Next                                       x
Esmqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

Figure 71. Resuming the monitoring

As soon as the command completes, both `run_clappmond` and `clappmond` are running again as shown in Figure 72.

```
risc1# ps -ef | egrep "imserv|app"
    root 11862 15202   0 11:49:24      -  0:00 /usr/es/sbin/cluster/clappmond im
ageappsrvr
    root 15202 21712   0 11:49:24      -  0:00 run_clappmond -sport 1000 -result
_node risc1 -script_id 0 -command_id 12 -command imageappsrvr -environment ?CLUS
TER_VERSION=3??GS_NODEID=1??APPLICATION_SERVER=imageappsrvr??MISC_DATA=??RESOURC
E_GROUP=resgrp1??RESTART_METHOD=/usr/sbin/cluster/events/utils/start_imagedemo -
d /fs1 -a risc1_svc??CLEANUP_METHOD=/usr/sbin/cluster/events/utils/stop_imagedem
o -a risc1_svc??NOTIFY_METHOD=/usr/haes44/notify.sh??MONITOR_METHOD=??FAILURE_AC
TION=fallover??RESTART_INTERVAL=16??RESTART_COUNT=1??STABILIZATION_INTERVAL=15??
MONITOR_INTERVAL=0??INSTANCE_COUNT=1??PROCESS_OWNER=root??PROCESSES=imserv??MONI
TOR_TYPE=process??HACMP_VERSION=__PE__??PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/u
sr/bin/X11:/sbin??ODMDIR=/etc/es/objrepos??LC_FASTMSG=true??PING_IP_ADDRESS= ??L
OCALNODEID=risc1??LOCALNODENAME=risc1??CM_CLUSTER_NAME=haes44??CM_CLUSTER_ID=6?
    root 22422 20404   0 11:30:56      -  0:00 imserv 10.10.10.1
    root 26144 25804   2 11:49:42  pts/2  0:00 egrep imserv|app
```

Figure 72. Monitoring has resumed

2.7.2.5 Considerations of suspended Application Monitoring

This section outlines important aspects to remember when the application monitor has been suspended.

Reminder 1
You cannot make changes to the application monitor configuration while it is in the suspended state.

Reminder 2
When monitoring is suspended, if a cluster event occurs that results in the resource group containing a monitored (but currently suspended) application being moved to another cluster node, the monitoring resumes automatically on the new node.

As an example, node risc1 has a monitored application running while node risc3 is in standby. Application Monitoring has been suspended. In fact there is no `run_clappmond` and `clappmond` running on node risc1 as follows:

```
risc1# ps -ef | egrep "imserv|app"
    root 19662 25804   4 11:46:50  pts/2  0:00 egrep imserv|app
    root 22422 20404   0 11:30:56      -  0:00 imserv 10.10.10.1
risc1#
```

Figure 73. Monitoring disabled on risc1

Node risc1 crashes, and ES performs a takeover over to node risc3. As soon as the takeover completes, Application Monitoring starts automatically on risc3 as shown in Figure 74.

```
risc3# ps -ef | egrep "imserv|app"
    root 11240 15706   0 11:56:40      -  0:00 /usr/es/sbin/cluster/clappmond im
ageappsrvr
    root 15706 16048   0 11:56:38      -  0:00 run_clappmond -sport 1000 -result
_node risc3 -script_id 0 -command_id 12 -command imageappsrvr -environment ?CLUS
TER_VERSION=3??GS_NODEID=2??APPLICATION_SERVER=imageappsrvr??MISC_DATA=??RESOURC
E_GROUP=resgrp1??RESTART_METHOD=/usr/sbin/cluster/events/utils/start_imagedemo -
d /fs1 -a risc1_svc??CLEANUP_METHOD=/usr/sbin/cluster/events/utils/stop_imagedem
o -a risc1_svc??NOTIFY_METHOD=/usr/haes44/notify.sh??MONITOR_METHOD=??FAILURE_AC
TION=fallover??RESTART_INTERVAL=16??RESTART_COUNT=1??STABILIZATION_INTERVAL=15??
MONITOR_INTERVAL=0??INSTANCE_COUNT=1??PROCESS_OWNER=root??PROCESSES=imserv??MONI
TOR_TYPE=process??HACMP_VERSION=__PE__??PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/u
sr/bin/X11:/sbin??ODMDIR=/etc/es/objrepos??LC_FASTMSG=true??PING_IP_ADDRESS= ??L
OCALNODEID=risc3??LOCALNODENAME=risc3??CM_CLUSTER_NAME=haes44??CM_CLUSTER_ID=6?
    root 16510 17580   4 11:59:03  pts/2  0:00 egrep imserv|app
    root 18080 13140   0 11:56:23      -  0:00 imserv 10.10.10.1
```

Figure 74. Monitoring restarted on node risc3

Reminder 3
When Application Monitoring is suspended on a node, if ES is stopped gracefully without takeover on this node, when ES is restarted the monitoring is resumed automatically.

2.7.3 Stopping the Application Monitoring

While suspending is a temporary action, stopping the Application Monitoring is a permanent change.

In order to stop the Application Monitoring definitely, use the SMIT menu shown in Figure 75, reachable with the `smit cm_cfg_process_appmon` fastpath.

```
              Define Process Application Monitor

Move cursor to desired item and press Enter.

  Add Process Application Monitor
  Change/Show Process Application Monitor
  Remove a Process Application Monitor







F1=Help              F2=Refresh           F3=Cancel            Esc+8=Image
Esc+9=Shell          Esc+0=Exit           Enter=Do
```

Figure 75. Removing the monitoring definition

Select **Remove a Process Application Monitor** and then synchronize the cluster resources.

If ES is currently active in the cluster, the monitoring is stopped dynamically through a DARE.

Chapter 3. Tivoli cluster monitoring

Tivoli cluster monitoring allows you to monitor the state of an HACMP cluster and its components on a Tivoli Desktop window.

This provides a centralized GUI to monitor:

- Cluster state and substate
- Configured networks and network state
- Participating nodes and node state
- Configured resource group location and state (ES only)
- Individual resource location (ES only)

> **Note**
>
> TME 10 has recently been renamed Tivoli Enterprise. This change has no effect on the information contained in this chapter.

3.1 Tivoli basics

Tivoli Management Environment (TME 10) is an integrated set of applications for network computing management, application management, and centralized control. It has been designed from the ground up using a client/server architecture. TME 10 is an environment designed for multi-platform applications, running on mainframes as well as on PCs.

TME 10 applications cover four network management areas:

- Deployment Management
- Availability Management
- Security Management
- Operations and Administration

3.1.1 Tivoli concepts

Tivoli Management Region (TMR) is the basic unit of Tivoli functionality. A TMR is a partition of your network. This partition depends on the following factors:

- Network topology
- Number of systems to be managed

- Security rules and concerns
- Organizational structure
- Availability

The Tivoli Enterprise environment is a three-tiered architecture. It consists of a variety of node types, which include:

TMR server contains the TME 10 Framework that is the base component for the TME 10 product line. It provides a set of common features and services that are used by TME 10 applications installed on the Framework. The services provided by the Framework include, but are not limited to, the following:

- An object oriented database to store the information about TME.
- A Relational Database Management System (RDBMS) Interface Module (RIM) that enables some TME 10 applications to write specific information to relational databases.
- A Task Library on which users can create tasks and execute the tasks on multiple TME 10 resources, including installation of TME 10 applications and other software.
- A query facility that enables users to search and retrieve information from a relational database located on the TMR server or on other servers.
- A scheduler that enables users to schedule all TME 10 operations including tasks created in the Task Library.

A *managed node* runs the full TME 10 Framework software and can perform the same functions performed by the TMR server. This is the machine from which the system administrators will manage other systems in the network. A managed node maintains a client database, which is smaller than the TMR server database. A managed node can also be a proxy system for a PC managed node, an endpoint gateway, and a NetWare managed site. Managed nodes can receive distributions, execute tasks, run monitors, send events, and store information in the local database. For more information on gateways and NetWare managed sites, see the *TME-10 Framework 3.6 Planning & Installation Guide.*

3.1.2 Tivoli Framework Components

Tivoli Framework provides the basic system management services such as communications, security, and presentations, creating an environment for the integration of all Tivoli applications.

Tivoli Framework is the main software component that must be installed on the TMR server. It consists of the following components:

- **Management Database** - to store the information about Tivoli environment. The database can be distributed between TMR server and all systems defined as managed nodes in the same TMR. The TMR server is the "master" of the region but is also a managed node.
- **oserv daemon** - a service that runs continuously on all managed nodes and provides communication coordination between all the systems in the Tivoli environment. It is a CORBA compliant Object Request Broker.
- **Tivoli Desktop** - a graphical user interface (GUI) that allows administrators to control and observe the TMR environment.
- **Command Line Interface (CLI)** - all the functions that can be performed through the GUI, can also be performed through a shell command line. You can use it for writing shell scripts to perform management functions on remote systems, or on systems that do not have a graphic terminal.
- **Tivoli Web Interface** - Tivoli Framework provides management capabilities through a Web browser. The basic function allowed through this Web interface is the management for the TMA endpoints.
- **Installation services** - A system that has installed the Tivoli Management Framework can be used to install appropriate Tivoli components and applications on other systems throughout the network. Tivoli Software Installation Services (SIS) provide a Java-based front end for the installation services.
- **Application services** - these services include task libraries, schedulers, a notification mechanism, and profile distribution. Tivoli Management Framework also supports Multiplex Distribution, which provides the fan-out mechanism for the profile distribution. Multiplex Distribution is especially useful in gateways to provide efficient communication.

3.1.3 Tivoli Distributed Monitoring

Distributed Monitoring is an application for remotely monitoring a wide range of systems and applications, including resources there are not part of the Tivoli environment.

A *monitor* is a program that controls specific aspects of resources (paging space, number of users, process existence, error condition, and so on). Its response actions are also part of the monitor definition (trigger, when, where, who, and so on). Monitors are defined in the TMR database and grouped in profiles that are distributed to and activated on the target systems (client nodes).

Distributed Monitoring provides the following functions to the TME 10 environment:

- Centralized resource monitoring
- Predefined monitoring collections (Universal, UNIX, SNMP, and so on)
- Mechanism to generate events
- Automated decisions and action in response to events
- Custom scripts for specific applications - as in HATivoli
- Data collection for analysis and capacity planning

The four major components of Distributed Monitoring are:

- Distributed Monitoring Engine
- Distributed Monitoring Gateway
- Distributed Monitoring Proxies (for monitoring non Tivoli resources)
- Distributed Monitoring Scheduler

3.2 Tivoli installation

This section uses the following HACMP and Tivoli environment:

- The *Tivoli Management Region (TMR)* (risc78-region), which consists of one TMR server and three managed nodes.
- The *TMR server* (risc78) manages one HACMP cluster.
- The HACMP cluster (cluster1) contains three *managed nodes* (arthur, merlin, and camelot). The TMR uses a separate network for Tivoli monitoring, which is outside the cluster networks.

This environment is illustrated in Figure 76 on page 69.

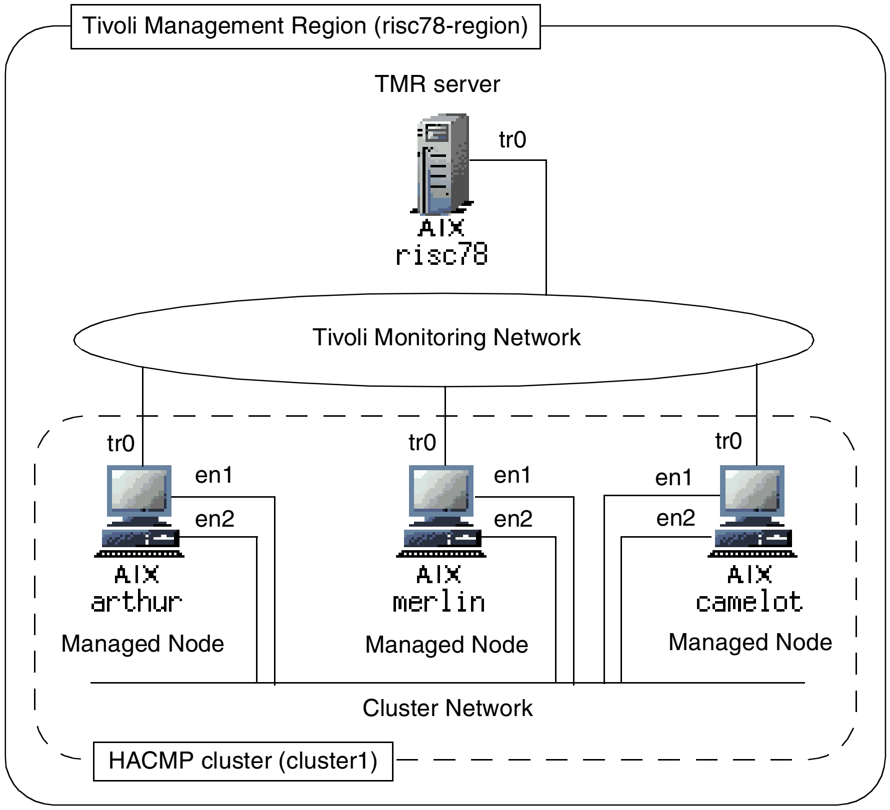

Figure 76. Tivoli Management Region (risc78-region)

We assume that AIX 4.3.3 and HACMP 4.4 (without HATivoli LPPs) are already installed on these machines. In addition, we need to install the software listed in Table 1 on each machine.

Table 1. Required software for TMR server and managed nodes

Software	TMR server	Managed Nodes
AIX 4.3.3	Installed	Installed
HACMP 4.4 (without HATivoli)	No need	Installed
TME 10 Framework	Yes	Yes
TME 10 Distributed Monitoring	Yes	Yes
TME 10 Application Extension Facility	Yes	No need
HATivoli (cluster.hativoli)	Yes	Yes

TME 10 Application Extension Facility (AEF) is a part of TME 10 Framework software product. You need a software license key for TME 10 Framework, but not for TME 10 Distributed Monitoring.

Attention

All the operations in this chapter performed as *root* user on the TMR server unless otherwise specified.

3.2.1 Preparations

Before starting Framework installation, you need to prepare the network and file system environments.

Network environment

You need to perform network planning for the TMR: hosts, addresses, and IP aliases (if applicable). You also need to add the hostnames and the IP addresses into your /etc/hosts file or in your nameserver database. We added the names and the addresses for risc78, arthur, merlin, and camelot in our /etc/hosts file on all machines in the TMR.

File system environment

You need to create the disk space for TME 10 software on all machines in the TMR. We created a separate file system mounted under /tivoli as follows:

```
# lsvg -p rootvg
rootvg:
PV_NAME           PV STATE    TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk1            active      94          0           00..00..00..00..00
hdisk0            active      94          0           00..00..00..00..00
hdisk4            active      479         471         96..96..87..96..96
# mklv -y tivolilv rootvg 60
tivolilv
# crfs -v jfs -p rw -d /dev/tivolilv -m /tivoli -A yes
Based on the parameters chosen, the new /tivoli JFS file system
is limited to a maximum size of 134217728 (512 byte blocks)

New File System size is 491520
# mount /tivoli
#
```

This simplifies space maintenance, future software modifications, and installations. The disk space needed for TME 10 Framework and TME10 Distributed Monitoring is approximately 240 MBytes.

If a cdrom file system in not yet created under /cdrom directory, do this as follows:

```
# crfs -v cdrfs -p ro -d /dev/cd0 -m /cdrom -A no
#
```

3.2.2 Install Framework on TMR server

Insert the TME 10 Framework 3.6 CD in the cdrom drive and issue the following commands:

```
# mount /cdrom
# cd /tivoli
# mkdir inst_dir
# cd inst_dir
# /cdrom/wpreinst.sh
to install, type ./wserver -c /cdrom
#
```

This operation creates a link in the current directory, /tivoli/inst_dir, to a file located on the CD, so you must run the `wpreinst.sh` preinstall program from the /tivoli/inst_dir directory.

To start installation, issue the `wserver` command. It opens two windows; the TME 10 Framework Server Install Options window shown in Figure 77 on page 72 and the TME 10 Framework Server Installation window shown in Figure 78 on page 73.

Because we use the /tivoli directory for installation, you need to modify all the fields in the TME 10 Framework Server Install Options window to use this directory except the *X11 Resource Files* field. We recommend you select all the *Server Install Options* for easier installation, as shown in Figure 77 on page 72.

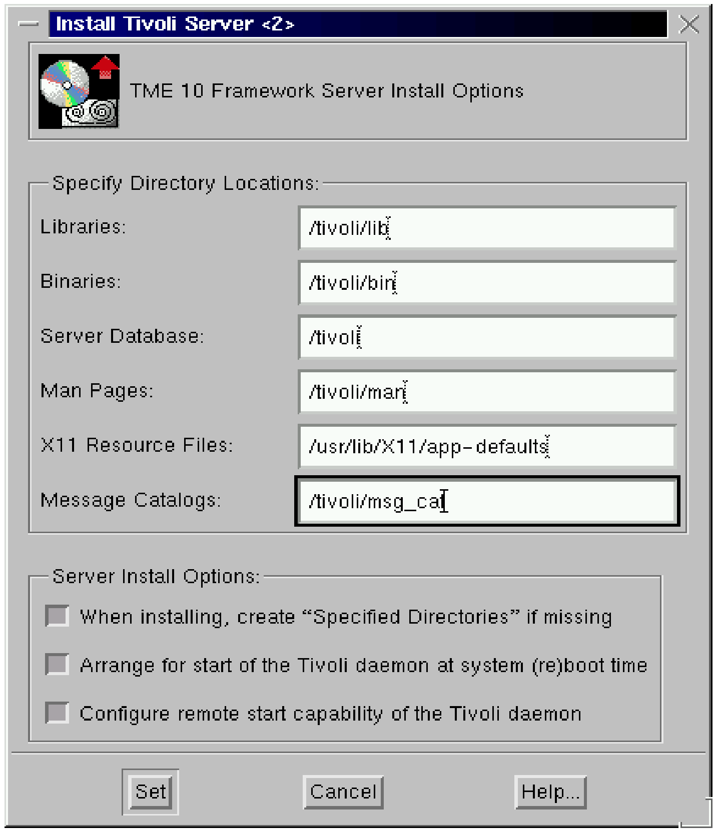

Figure 77. TME 10 Framework Server Install Options window

After entering the pertinent information, click on **Set** button.

This will open the TME 10 Framework Server Installation window. In this window you need to register the license key. Enter your license key and click on **Install** button (see Figure 78 on page 73).

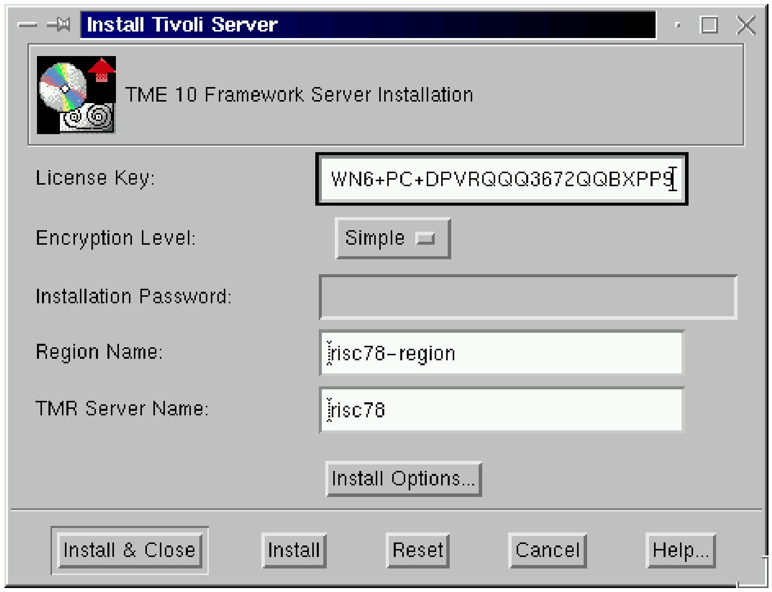

Figure 78. TME 10 Framework Server Installation window

Before the installation process starts, you will see a TME Install confirmation window as shown in Figure 79 on page 74. Click on **Continue Install** button to proceed.

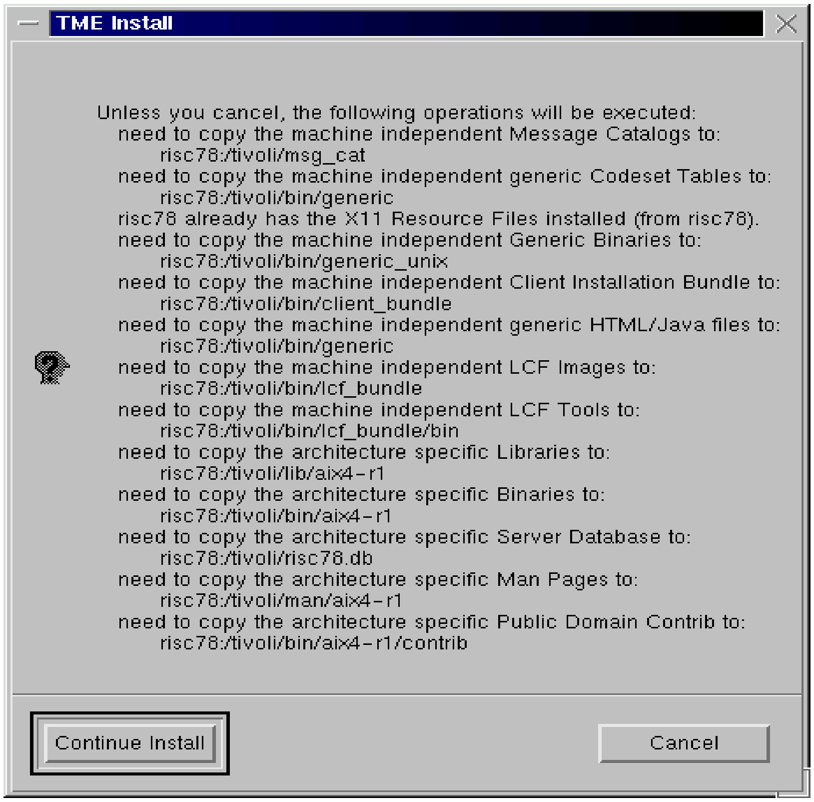

Figure 79. Confirmation window for Framework server installation

During the installation you can observe the installation progress in the TME Install window shown in Figure 80 on page 75.

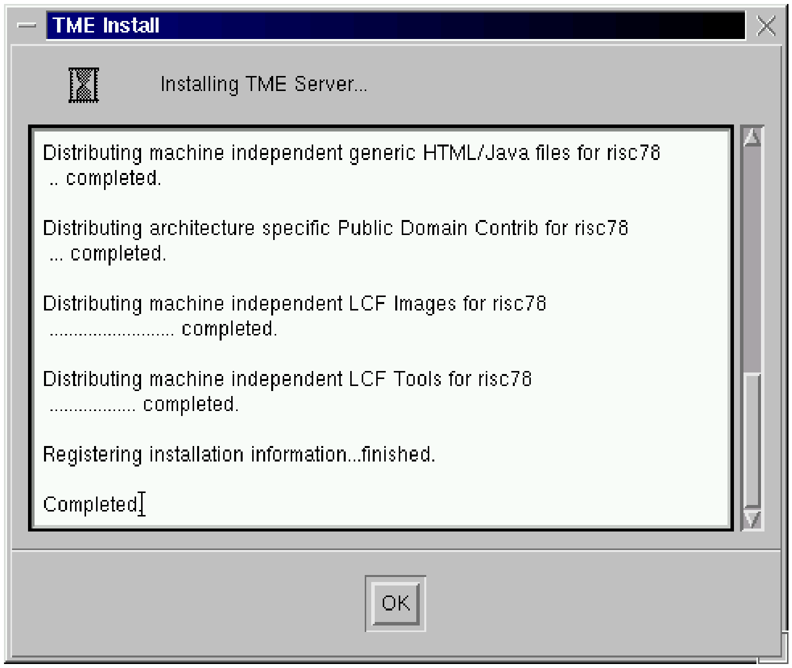

Figure 80. TME Install window

When installation is completed, click on **OK** button to close the window, then click on **Cancel** button to close the window shown in Figure 78 on page 73. The TME Desktop window shown in Figure 81 on page 76 is started automatically at the end of the installation process.

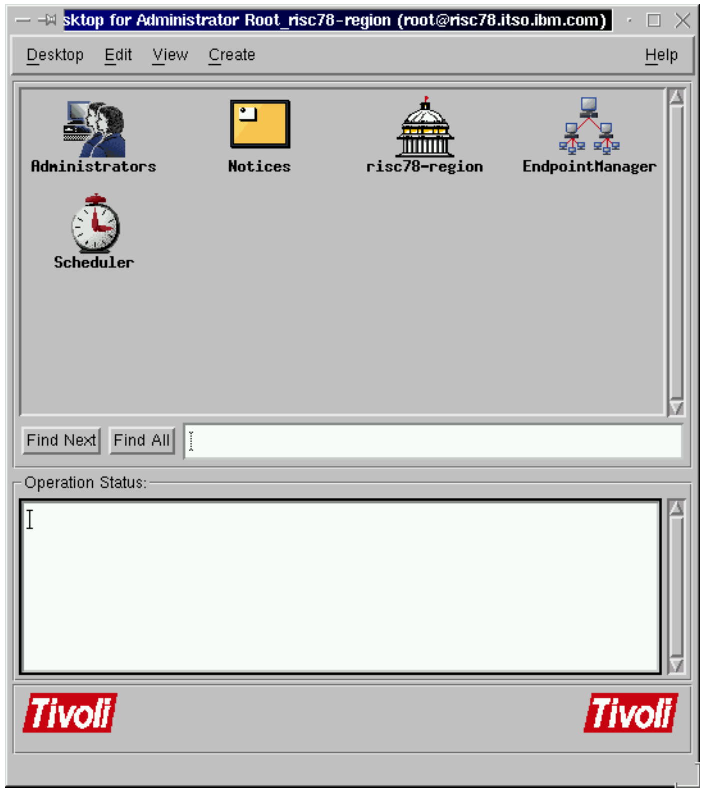

Figure 81. TME Desktop window

The TME 10 Framework installation program adds the following lines at the end of /etc/rc.nfs file:

```
if [ -f /etc/Tivoli/oserv.rc ]; then
    /etc/Tivoli/oserv.rc start
    echo "Tivoli daemon started."
fi
```

Therefore, when the system is next rebooted, the `oserv` daemon will be started automatically. You also need to add the `setup_env.sh` script to be run at logon time by adding the following line at the end of /.profile file:

```
. /etc/Tivoli/setup_env.sh
```

To get a clean environment after completing the TME 10 Framework installation on the TMR server, we recommend you reboot the system.

Registering the license key later

If you do not have the license key at the installation time, you can still continue the installation. However, the TME Desktop window will not start.

You can register the license key later by using the following command:

1. Set up Tivoli environment (you need the dot <.> in front of the command).

```
# . /etc/Tivoli/setup_env.sh
#
```

2. Start the oserv daemon.

```
# /etc/Tivoli/oserv.rc start
#
```

3. Register your license key.

```
# odadmin set_platform_license 123-ABC-D345678AAAAAAA
#
```

4. Make sure that your license key is registered.

```
# odadmin get_platform_license
123-ABC-D345678AAAAAAA#
```

3.2.3 Adding managed nodes to the TMR

Adding managed nodes to the TMR consists of two steps:

1. Defining managed nodes to the TMR
2. Installing TME 10 Framework on the managed nodes

3.2.3.1 Defining managed nodes to the TMR

This step defines managed nodes to the TMR. They are arthur, merlin, and camelot.

To start the TME 10 Desktop window shown in Figure 81 on page 76, use the `tivoli` command:

```
# tivoli
```

To define the clients (managed nodes), you need to open the default Policy Region window by double clicking on **risc78-region** icon in the TME Desktop window:

You will see the *Policy Region: risc78-region* window as shown in Figure 82.

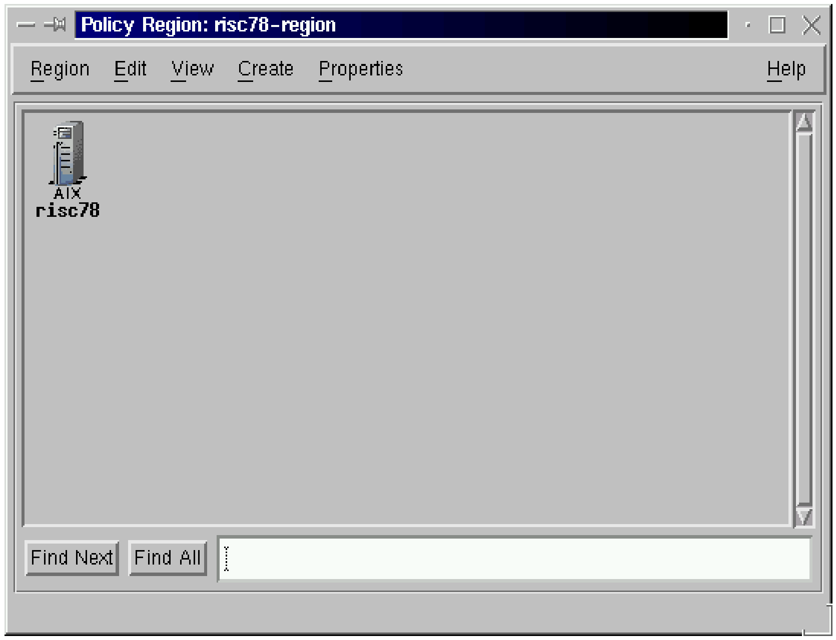

Figure 82. Policy Region: risc78-region window

To open *Client Install* window, select **Create** > **ManageNode...** in this window as shown in Figure 83 on page 79.

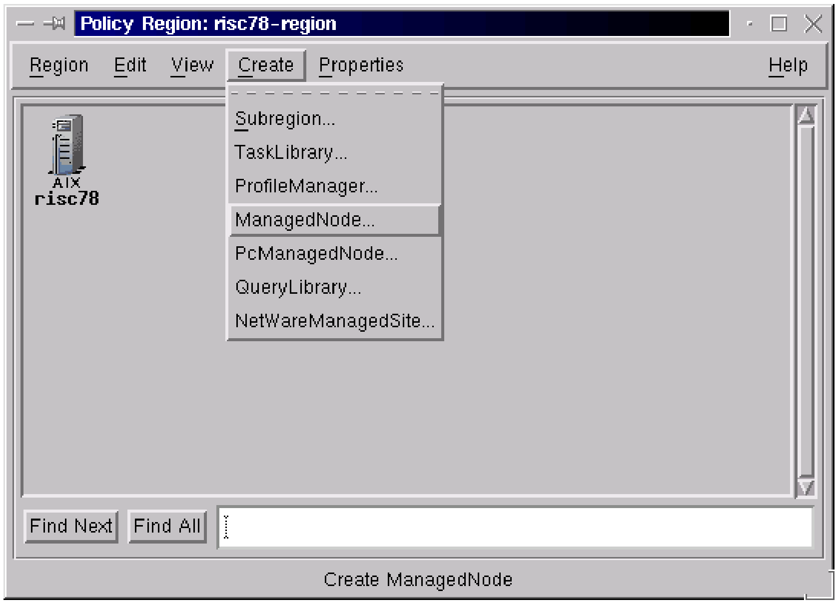

Figure 83. Opening Client Install window

You will see the *Client Install* window as shown in Figure 84 on page 80.

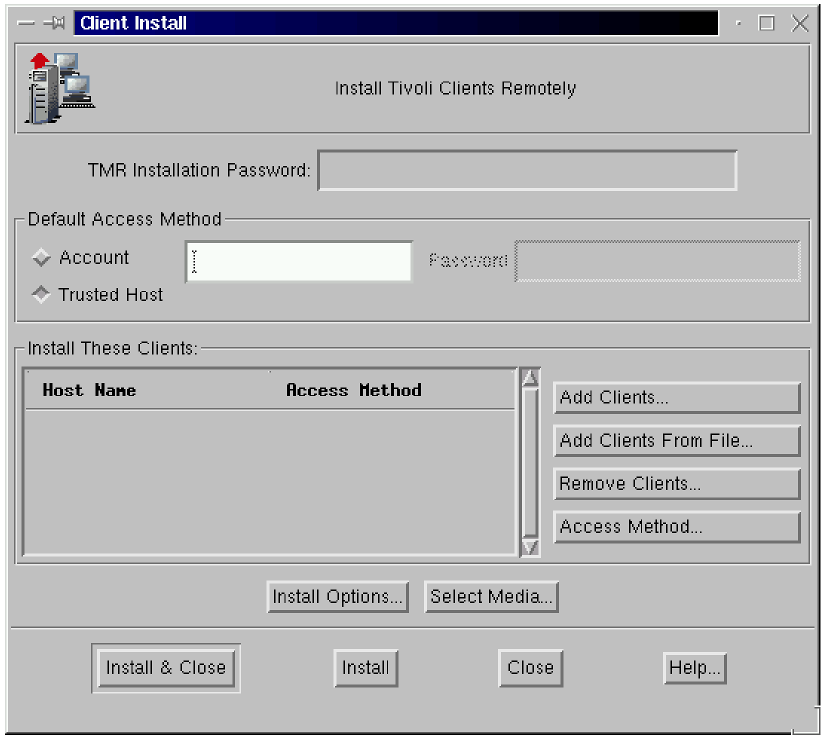

Figure 84. Client Install window

To define the clients, click on **Add Clients...** button. You will access the *Add Clients* window as shown in Figure 85 on page 81.

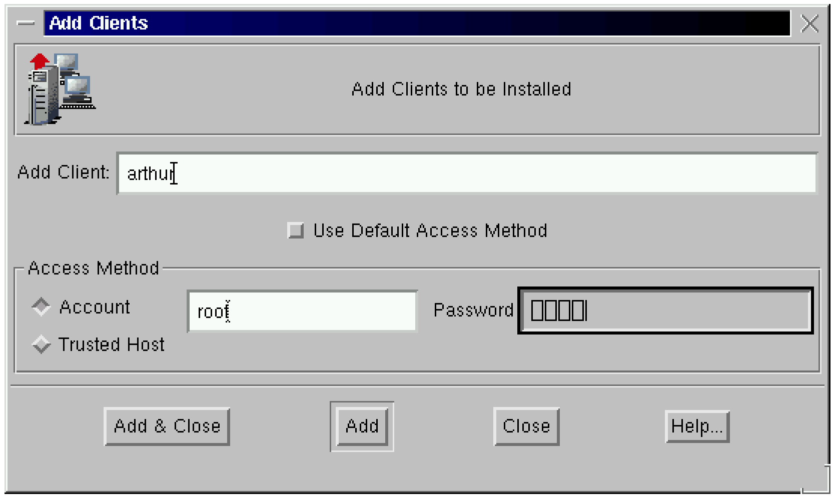

Figure 85. Add Clients window

In this window, enter the hostnames of the clients you want to install, one at a time, in the *Add Client* field. In the *Access Method* area, select **Account**, type *root* and its password, and then click on the **Add** button. Perform the same operation for the rest of the clients (merlin and camelot). In our case we added arthur, merlin, and camelot. When you finished adding clients, click on **Close** button.

The clients are checked by the installation program at the time you enter their data (hostname, username, and password). Make sure the clients are up and running, and the supplied data is correct. Otherwise, the clients will not be added.

Now *Client Install* window has three clients; arthur, merlin, and camelot as shown in Figure 86.

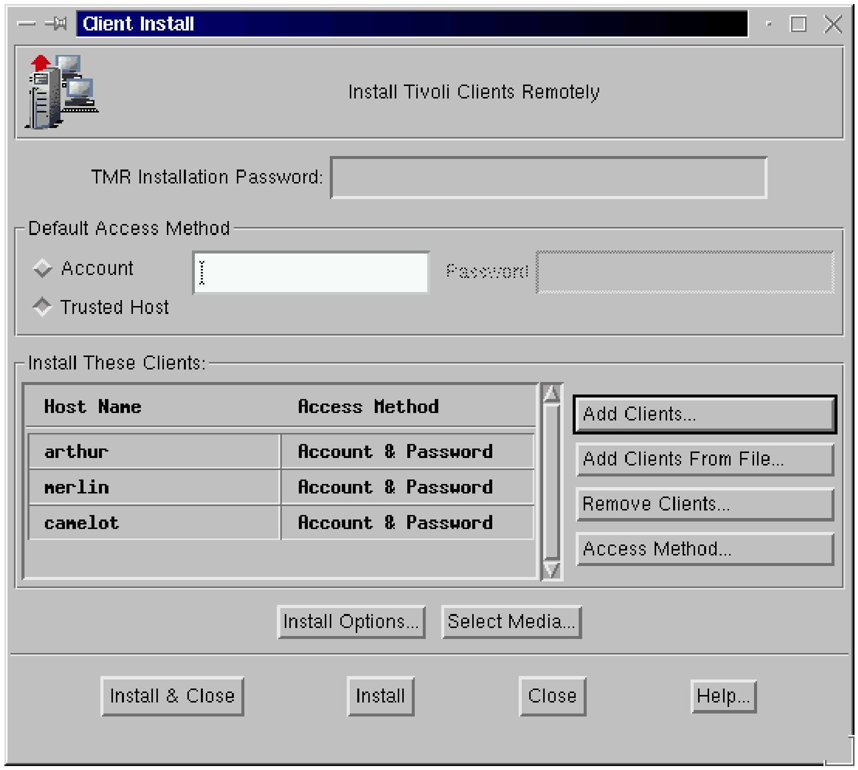

Figure 86. Client Install window with three clients

3.2.3.2 Installing Framework to the managed nodes

This step describes how to install the TME 10 Framework to the previously defined managed nodes; arthur, merlin, and camelot.

The TMR server will act as a file server for the clients to be installed.

To install the TME 10 Framework to the managed nodes, you need to make the installation files available to the clients. Click on **Select Media...** button in the *Client Install* window. You will see the *File Browser* window as in Figure 87 on page 83.

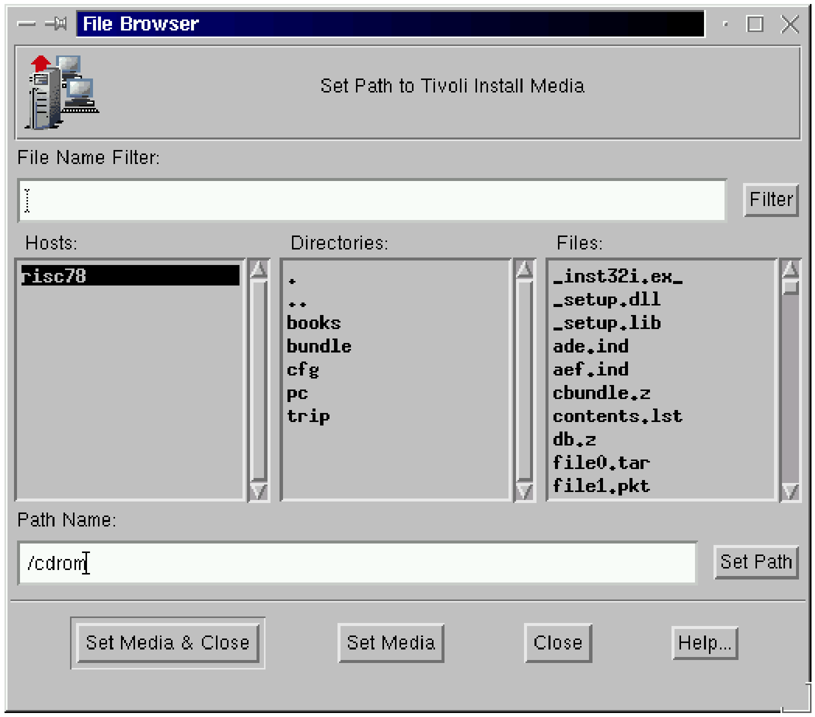

Figure 87. File Browser window

In this window, select the host that has the installation files (in our case, it is risc78) and type the path name (in our case, it is /cdrom). Then click on **Set Media & Close** button. The *Install Options* window shown in Figure 88 on page 84 will pop up.

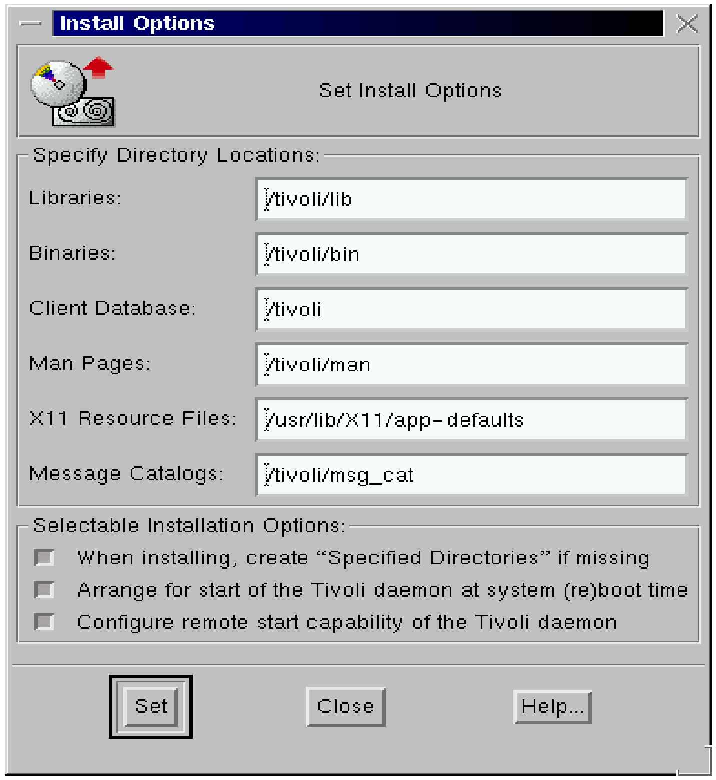

Figure 88. Install Options window

We recommend to select all three options in the *Selectable Installation Options* field.

The install program uses the path names that you specified in Figure 77 on page 72 as defaults. If you want to change the defaults, modify this window, then click on the **Set** button.

At this point preparation for installation is completed. Click on **Install** button in the *Client Install* window in Figure 86 on page 82.

You will see the confirmation window as shown in Figure 89 on page 85. To proceed with the installation, click on the **Continue Install** button.

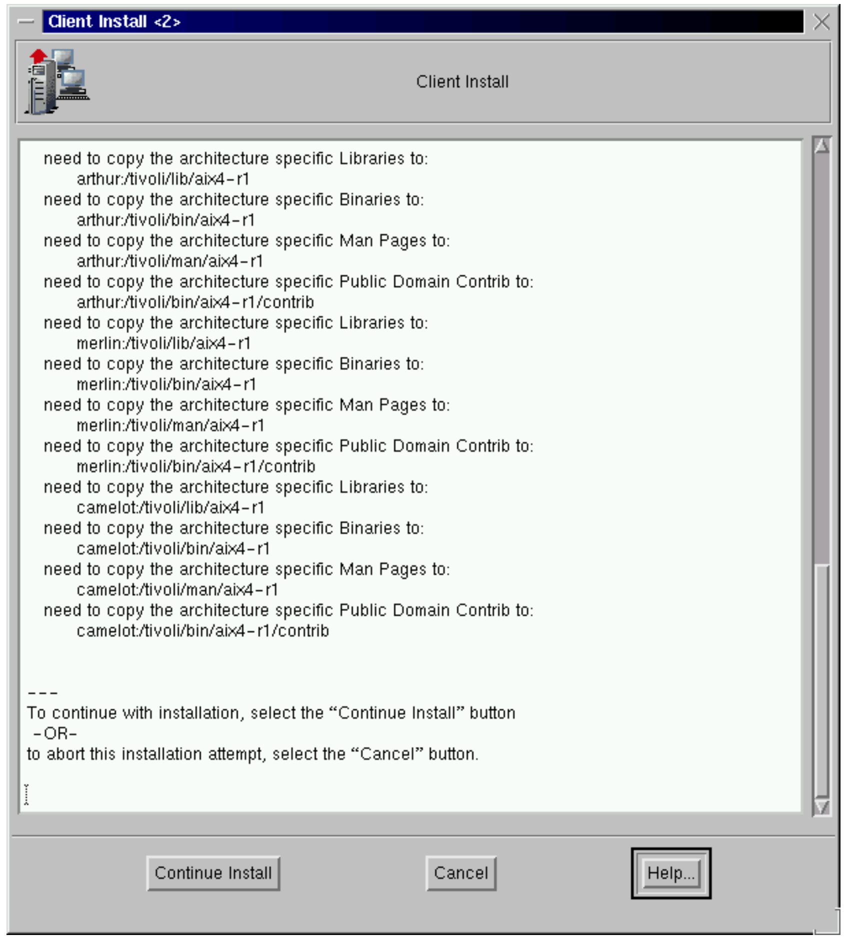

Figure 89. Client Install confirmation window

When the installation is completed, click on **Close** button (this button was labeled **Cancel** in Figure 89 on page 85), then click on **Close** button on the *Client Install* window shown in Figure 86 on page 82.

You will see the *Policy Region: risc78-region* window containing all the clients you defined in Section 3.2.3.1, “Defining managed nodes to the TMR” on page 77 as shown in Figure 90 on page 86.

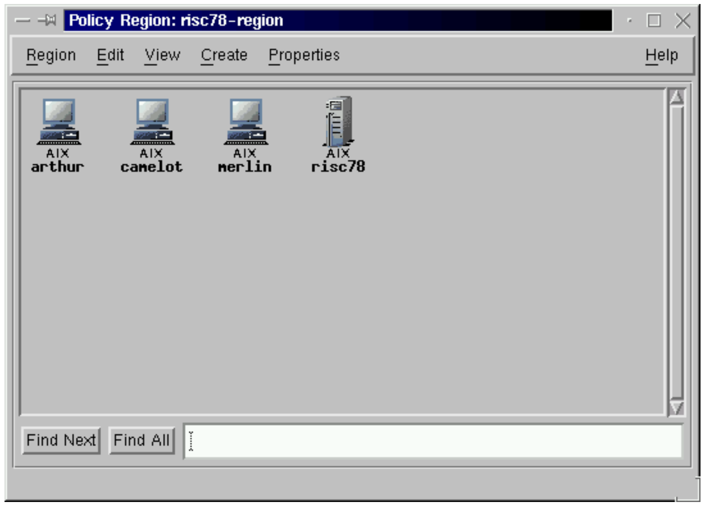

Figure 90. Policy Region window with the added clients

To close the *Policy Region: risc78-region* window, select **Region** > **Close** as shown in Figure 91.

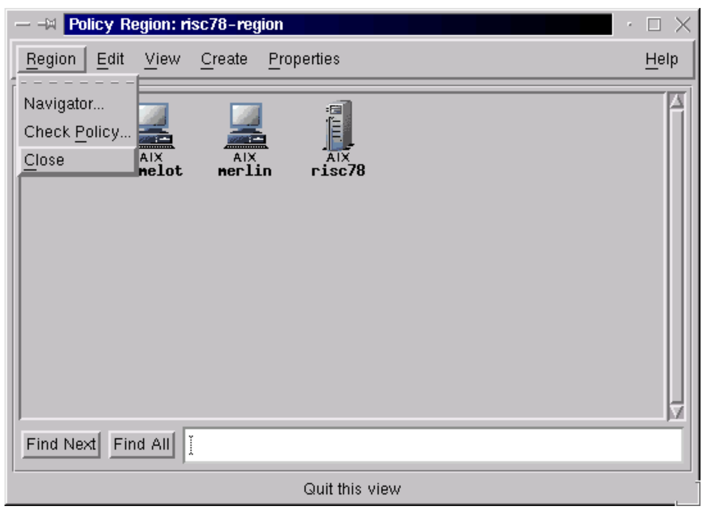

Figure 91. Closing Policy Region window

3.2.4 Verifying Tivoli installation

After the clients installation has been completed, the oserv daemon (remote object dispatcher daemon) is supposed be up and running. You can check if the oserv daemon is running on the managed nodes using the `odadmin` and `wping` commands.

The odadmin command

Use the `odadmin odlist` command as follows:

```
# odadmin odlist
Region      Disp  Flags  Port           IPaddr   Hostname(s)
1232341272     1    ct-    94        9.12.0.78   risc78.itso.ibm.com,risc78
              12    ct-    94        9.12.0.18   arthur
              13    ct-    94        9.12.0.50   merlin
              14    ct-    94        9.12.0.19   camelot
#
```

Figure 92. The odadmin odlist command output

Looking at the Flags column, notice the three flags <ct->. If the first flag is <c>, this means the installation was successful and the remote oserv daemon is up and connected to the oserv daemon running on the TMR server. If it is a question <?> mark, this means the installation was probably not completed and the client is in an inconsistent state. If it is a minus <-> flag, this indicates that remote oserv daemon is down. If it is down, but software installation was successful, you can start the oserv daemon by using the `odadmin` command:

```
# odadmin start <Disp_#>
```

The wping command

You can also use the `wping` command to check if the oserv daemon is running on the managed nodes as follows:

```
# wping arthur
object dispatcher on arhter is alive
# wping merlin
object dispatcher on merlin is alive
# wping camelot
object dispatcher on camelot is alive
#
```

The oserv daemon on all managed nodes is listening on port 94 TCP and UDP by default.

3.2.4.1 Back up the TMR database

We recommend you back up the TMR database on the TMR server and managed nodes by issuing the `wbkupdb` command:

```
# wbkupdb

Starting the snapshot of the database files for risc78...
............................................................
............................................................
........
Starting the snapshot of the database files for arthur...
........
Starting the snapshot of the database files for merlin...
...
Starting the snapshot of the database files for camelot...
...

Backup Complete.
#
```

We recommend that you back up the TMR database after each installation and configuration step. This may help you avoid repeating the entire installation from the beginning in case of failure.

3.2.4.2 Recovering from failed client installation

Depending on the client status, you may have to manually delete all Tivoli files already copied on the client in the /tivoli directory.

To delete the client object from the TMR database, use the `wrmnode` command:

```
# wrmnode -f arthur -d 12
#
```

This example deletes the client arthur from the TMR database. The dispatcher number is 12. You can get this number from the output of the `odadmin odlist` command as shown in Figure 92 on page 87.

Then check the TMR database using the `wchkdb` command as follows:

```
# wchkdb -u

wchkdb: Preparing object lists:
wchkdb: Checking object database:
................................................................................
................................................................................
................................................................................
................................................................................
....................................
wchkdb: Done checking object database.
#
```

Note

You may need to execute the `wchkdb -u` command several times before you no longer receive any error messages.

When the TMR database is checked without errors, repeat the client installation procedure from Section 3.2.3, “Adding managed nodes to the TMR” on page 77.

3.2.5 Installing TME 10 Distributed Monitoring

This step describes how to install the TME 10 Distributed Monitoring software to the TMR server and the managed nodes: arthur, merlin, and camelot. You also need to install additional Monitoring Collections to the TMR server.

3.2.5.1 Installing TME 10 Distributed Monitoring

Insert the TME 10 Distributed Monitoring CD in the cdrom drive on the TMR server, and access the /cdrom file system.

To install the TME10 Distributed Monitoring, on the TME Desktop window select **Desktop** > **Install** > **Install Product...** as shown in Figure 93 on page 90.

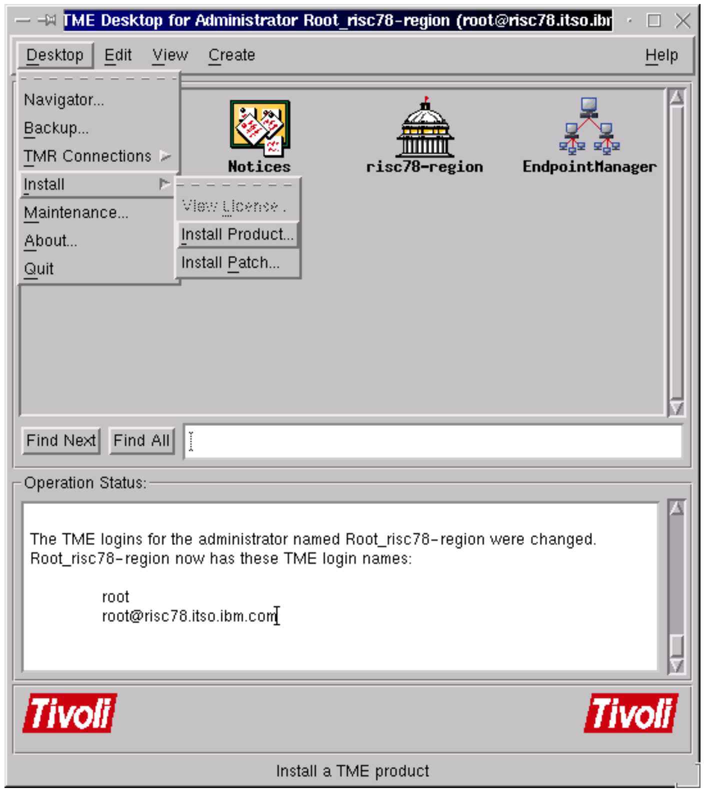

Figure 93. Start Install Product

You will see the *Install Product* window shown in Figure 94 on page 91.

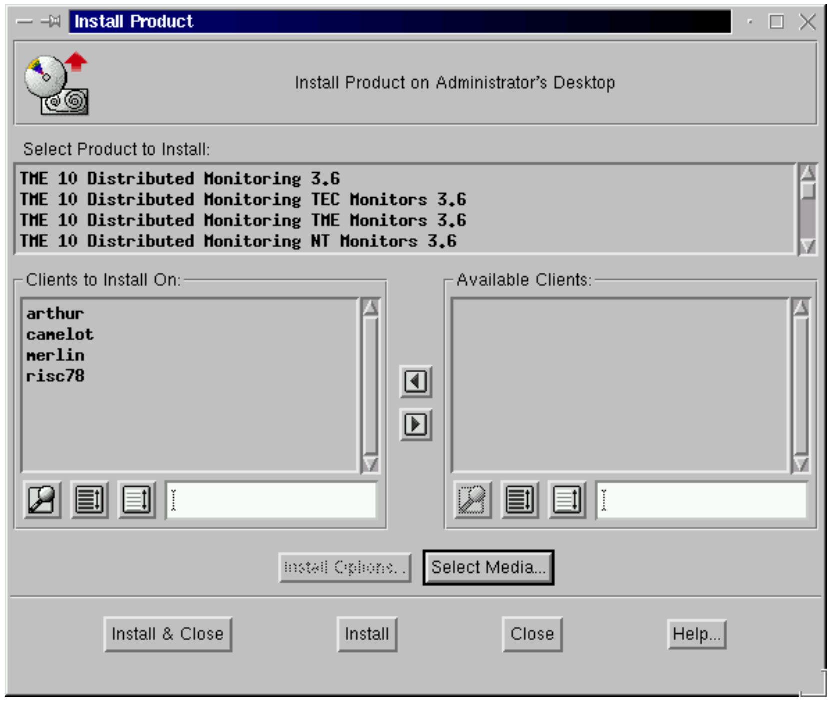

Figure 94. Install Product window

In this window click on the line **TMR 10 Distributed Monitoring 3.6** in the *Select Product to Install* area. You will see the *Install Options* window shown in Figure 95.

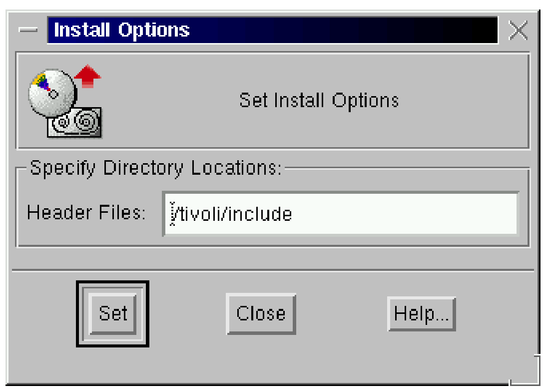

Figure 95. Install Options window

To set the install options, click on **Set** button. Then click on **Install & Close** button in the *Install Product* window shown in Figure 94 on page 91.

You will see *Product Install* window similar to Figure 89 on page 85. Click on the **Continue Install** button to proceed. When the installation process is completed, click on the **Close** button.

3.2.5.2 Installing additional Monitoring Collections

After the Distributed Monitoring installation is complete, you need to install the following additional Monitoring Collections only on the TMR server:

- TME 10 Distributed Monitoring Universal Monitors 3.6
- TME 10 Distributed Monitoring Unix Monitors 3.6
- TME 10 Distributed Monitoring SNMP Monitors 3.6

To install these Monitoring Collections, select only the TMR server (risc78) in *Clients to Install On* area in the *Install Product* window shown in Figure 94 on page 91.

Because the *Install Product* window does not allow you to select more than one Monitoring Collection at a time, you need to perform this operation for each Monitoring Collection.

3.2.6 Installing AEF on TMR server

AEF provides the TME 10 Framework with a set of graphic tools. HATivoli uses these tools to generate its own monitoring windows and menus for the Extended Node Properties application described in Section 3.4.1, “Monitoring cluster using HATivoli GUI” on page 114.

You need to insert the TME 10 Framework CD in the cdrom drive and mount the /cdrom file system.

Installing the AEF on the TMR server uses the same procedure as described in Section 3.2.5.1, “Installing TME 10 Distributed Monitoring” on page 89, with the following differences:

- Select TME 10 AEF, Version 3.6 in *Select Product to Install* area.
- Select only the TMR server (risc78) in *Clients to Install On* area.

See Figure 96 on page 93.

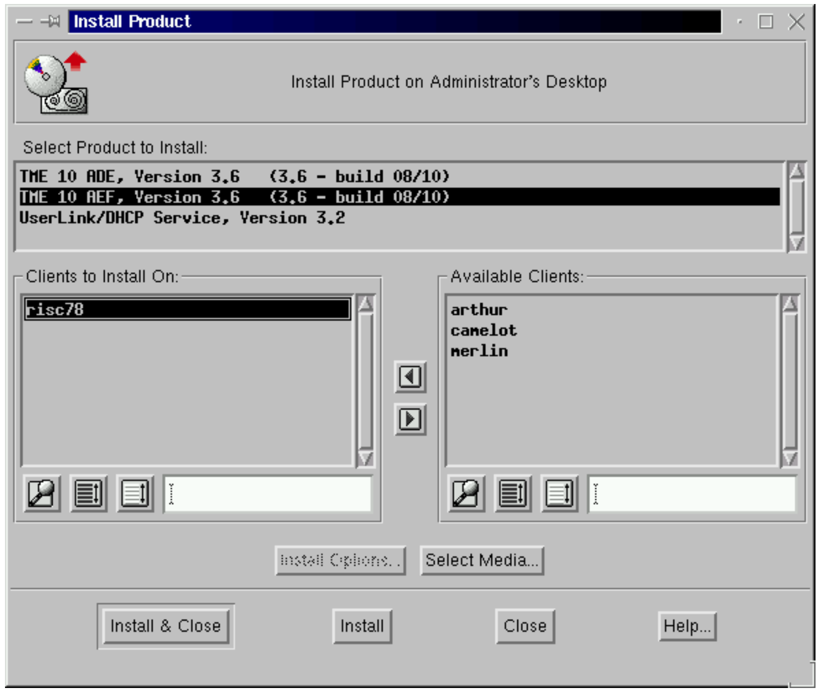

Figure 96. Install Product window for AEF installation

3.2.7 Installing patches on the Tivoli products

According to Tivoli recommendation, we updated Tivoli products to 3.6.1 level. This section describes the Tivoli software update procedure we used in our environment.

You need to insert the CD containing the TME 10 Framework patches in the cdrom drive and mount the /cdrom file system.

To install the patches, in the TME Desktop window select **Desktop** > **Install** > **Install Patch...** as shown in Figure 97 on page 94.

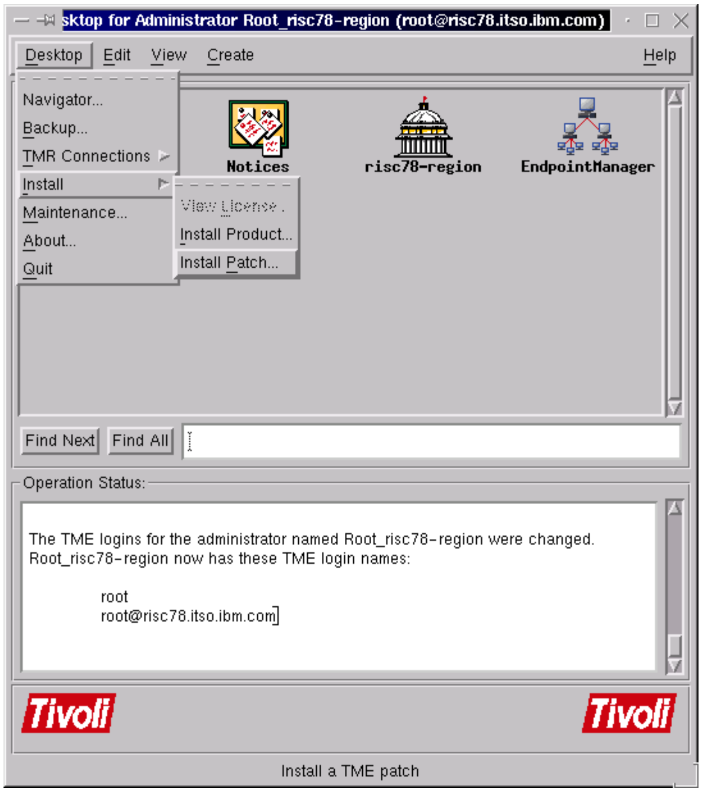

Figure 97. Opening Install Patch window

You will see the Install Patch window as shown in Figure 98 on page 95.

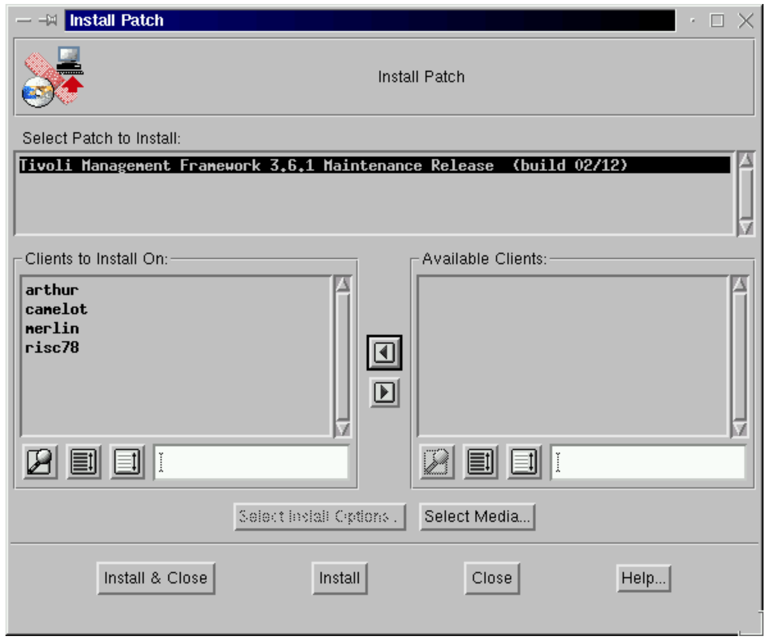

Figure 98. Install Patch window

Select the patches you want to install. Also select the clients on which you want to install the selected patches. We selected **TIvoli Management Framework 3.6.1 Maintenance Release** in the *Select Patch to Install* area, and arthur, merlin, camelot, and risc78 in the *Clients to Install On* area. Then click on the **Install & Close** button.

You will see the Install Patch window. This window is similar to the Client Install window shown in Figure 89 on page 85. Click on the **Continue Install** button. When you get the message “Finished patch installation,” click on the **Close** button.

Attention

After each patch installation you *must* refresh the `oserv` daemon on the TMR server and all the managed nodes. You can do this by issuing the `odadmin` command on the TMR server:

```
# odadmin reexec all
Waiting for clients to finish reexec before doing TME server..
Clients have resumed operation. Reexec'ing TME server.
#
```

This operation will restart `oserv` daemon on all clients first, then on the TMR server. This will cause TME Desktop GUI to close, so you will need to reopen it to proceed patch installation.

Repeat patch installation procedure for Distributed Monitoring on both TMR server and managed nodes. The same procedure is applied to AEF on TMR server.

After the software update procedure is finished, backup the TMR database as described in Section 3.2.4.1, "Back up the TMR database" on page 87.

3.3 HATivoli Installation and Configuration

This section describes the TMR customization and the HATivoli installation and configuration. HATivoli comes with HACMP 4.4, and is packaged in AIX LPP format.

3.3.1 Configuring the TMR for HATivoli

Before installing the HATivoli, you need to perform administrative tasks on the TMR server. These tasks consist of defining and creating the necessary containers (objects) that will be used by the HATivoli installation scripts to store the objects they create.

Configuring the TMR for HATivoli consists of the following steps:

1. Creating a new policy region (named hacmp44)
2. Adding the necessary resources in the policy region
3. Creating the profile manager
4. Creating the indicator collection
5. Creating the subscribers

3.3.1.1 Creating policy region hacmp44

A *policy region* is an administrative object that contains a set of resources. It also establishes the relationships (policies) between these resources.

When you installed TME 10 Framework on the TMR server, risc78-region policy region was created in the TMR automatically. This is represented by the following icon:

We can use this policy region to manage our HACMP cluster. However, for ease of management purposes, we have decided to create a new policy region containing the necessary managed resource types for managing the HACMP cluster.

The policy region created in this section is used by the configuring HATivoli step described in Section 3.3.4.1, “Creating monitoring profiles” on page 110.

To create a policy region for the HACMP cluster, select **Create > Region**... in TME Desktop window, as shown in Figure 99 on page 98.

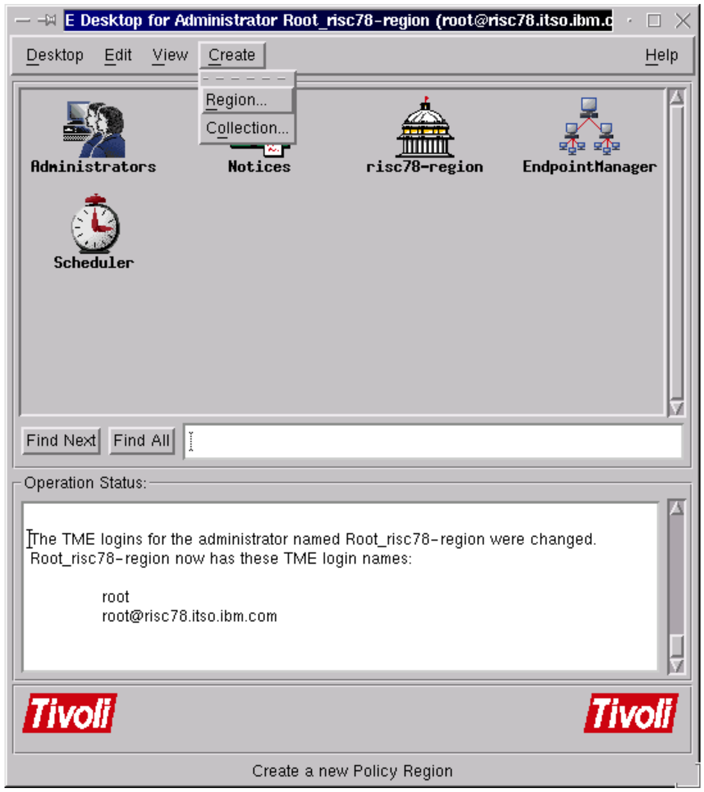

Figure 99. Creating a new region

You will see the *Create Policy Region* window shown in Figure 100. Enter the region name, and click on the **Create & Close** button. We named this policy region hacmp44.

Figure 100. Create Policy Region window

3.3.1.2 Setting the resource types to the policy region

The *profile manager* is one of resources managed by a policy region. HATivoli requires a profile manager as a container of its own monitoring profiles and subscribers. Therefore, you need to set the *profile manager* resource type to the policy region in advance.

Similarly, HATivoli requires five more resource types set to the policy region in advance. The following are the required resource types for the policy region:

- IndicatorCollection
- ManagedNode
- ProfileManager
- SentryProfile
- TaskLibrary

To set the required resource types, select **Properties** > **Managed Resources...** as shown in Figure 101.

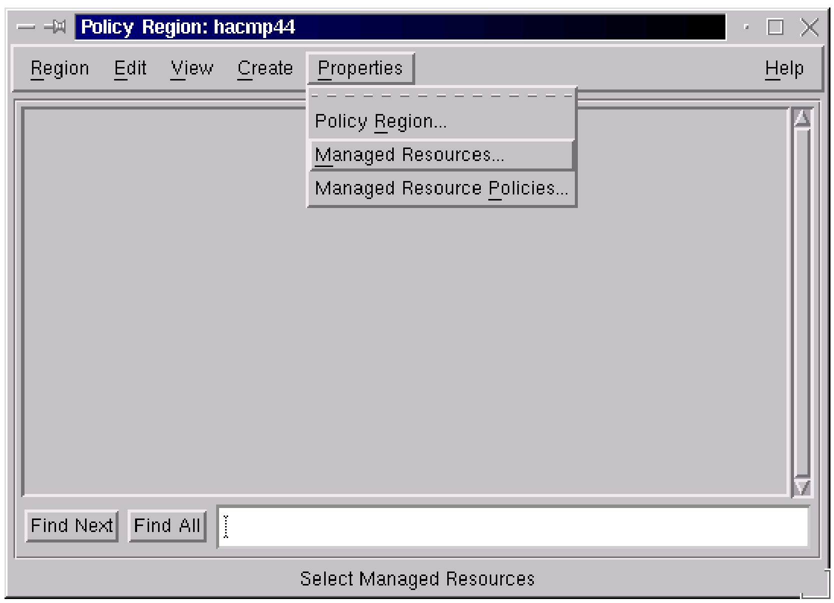

Figure 101. Opening the Set Managed Resources window

You will see the *Set Managed Resources* window as shown in Figure 102 on page 100.

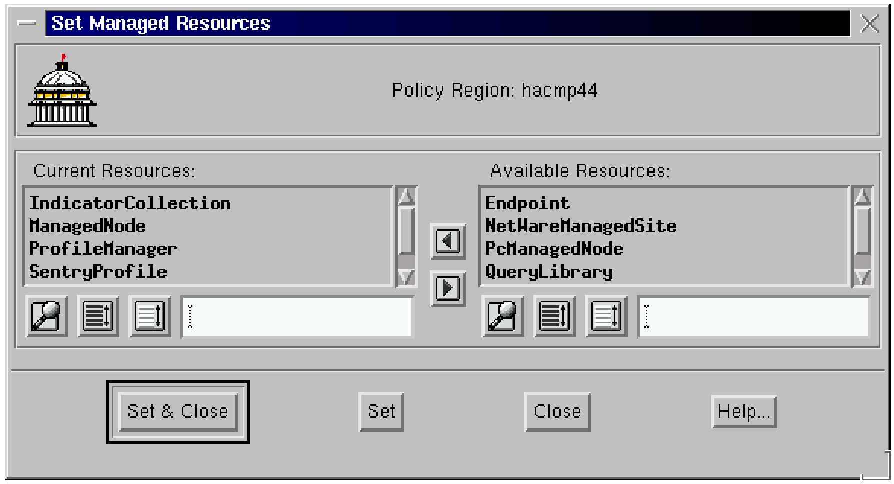

Figure 102. Set Managed Resources for policy region hacmp44 window

In this window, move the required five resource types from the *Available Resources* area to the *Current Resources* area, then click on the **Set & Close** button.

You will see a new policy region icon named hacmp44 on the TME Desktop window as shown in Figure 103 on page 101.

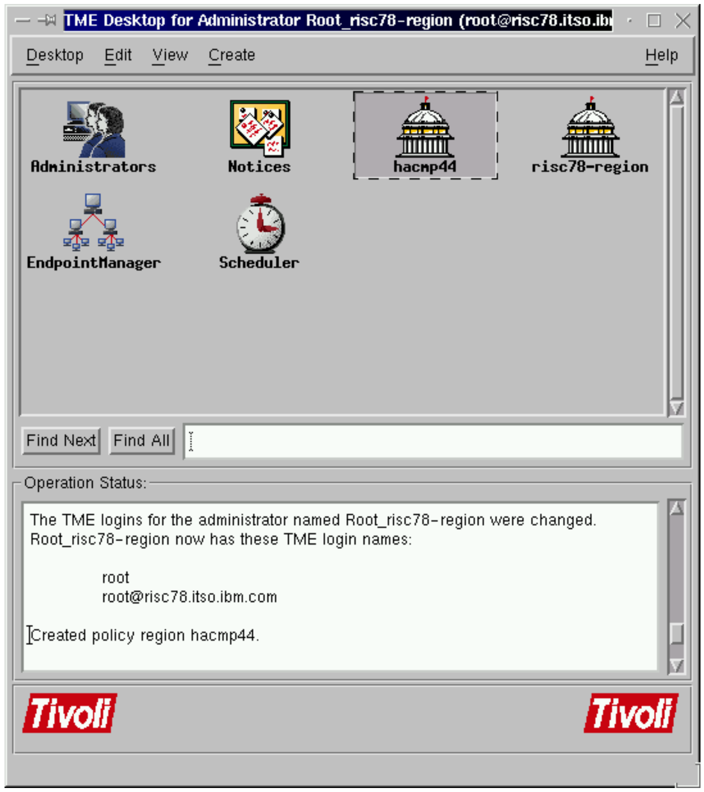

Figure 103. TME Desktop window with hacmp44 icon

3.3.1.3 Creating the profile manager

The profile manager groups the monitoring profiles that will be created and used by HATivoli. The profile manager also links the monitoring profiles to the cluster nodes. In our case, the subscriber nodes are arthur, merlin, and camelot.

The profile manager created in this section is used by the configuring HATivoli step described in Section 3.3.4.1, “Creating monitoring profiles” on page 110.

To create a profile manager in the hacmp44 policy region, double click on **hacmp44** icon in the TME Desktop window, as shown in Figure 103. You will see *Policy Region: hacmp44* window as shown in Figure 104 on page 102.

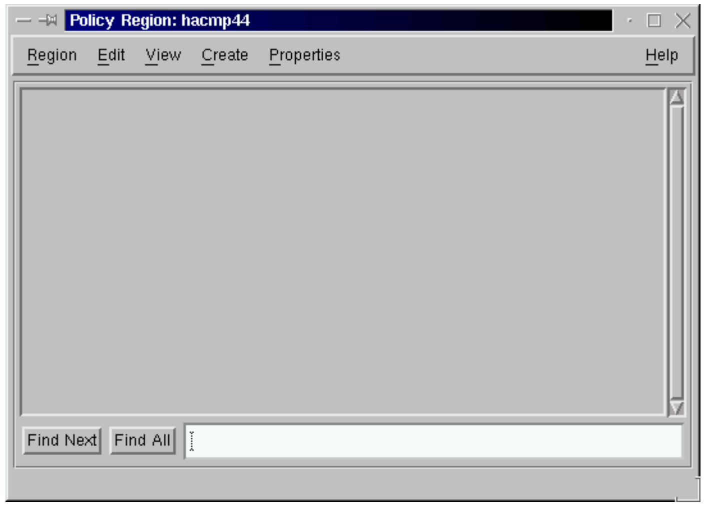

Figure 104. Policy Region hacmp44 window

To create an empty profile manager in the hacmp44 policy region, select **Create** > **ProfileManger...** as shown in Figure 105.

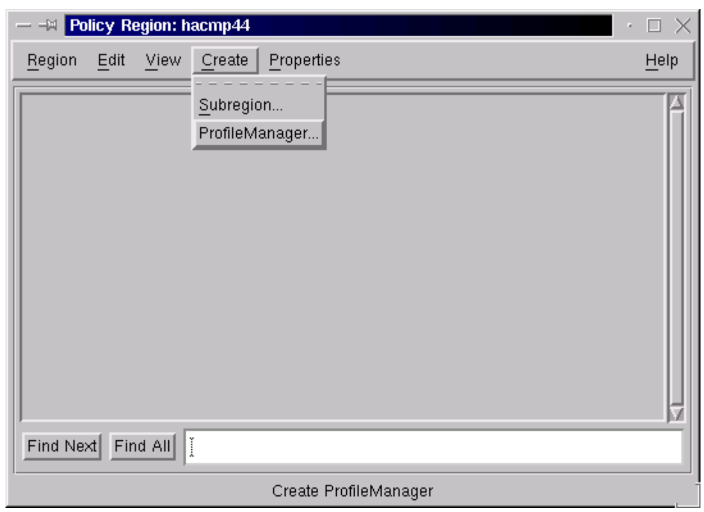

Figure 105. Opening the Create Profile Manger window

You will see the Create Profile Manager window shown in Figure 106 on page 103. Specify the name of profile manager icon. We named it “ha_cluster.” Then click on the **Create & Close** button.

Figure 106. Create ProfileManager window

A new icon named ha_cluster is added in the Policy Region: hacmp44 window as shown in Figure 107.

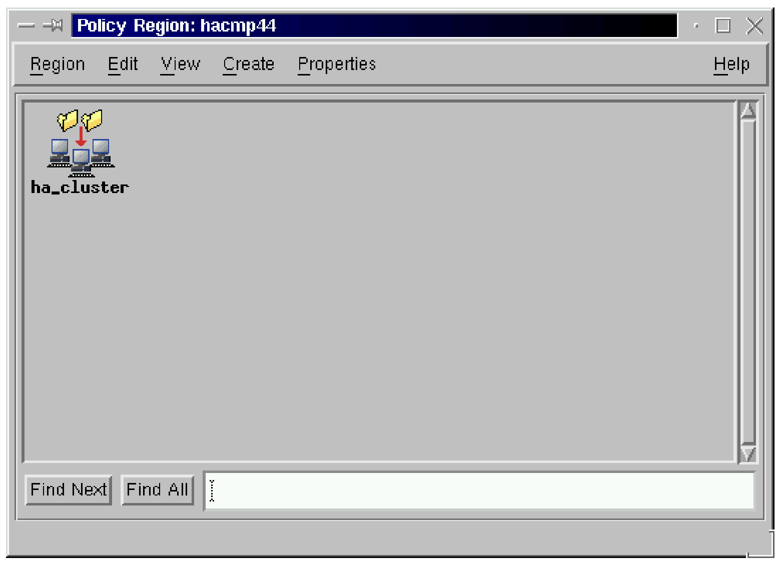

Figure 107. Policy Region hacmp44 window with ha_cluster profile manager icon

3.3.1.4 Creating the indicator collection

After creating the ha_cluster profile manager, you need to create an empty indicator collection in the hacmp44 policy region. The HATivoli installation will create the indicators under this indicator collection. The indicators collect the output data from the monitors executed on the subscriber nodes arthur, merlin, and camelot.

The indicator collection created in this section is used by the configuring HATivoli step described in Section 3.3.4.1, “Creating monitoring profiles” on page 110.

To create an empty indicator collection in the hacmp44 policy region, select **Create > IndicatorCollection...** , in the Policy Region: hacmp44 window shown in Figure 108.

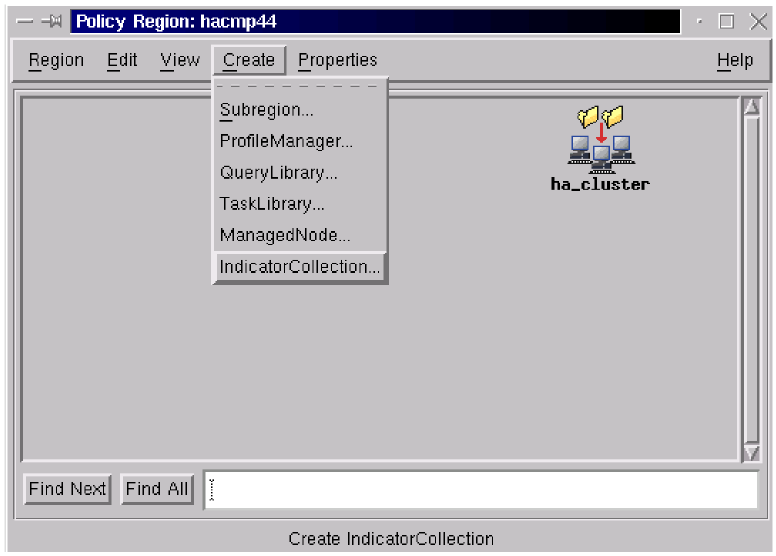

Figure 108. Opening the Create Indicator Collection window

In the *Create Indicator Collection* window, type the name of indicator collection in the *Name* field. We named it “cluster1_collection” as shown in Figure 109 on page 105. Then click on **Create & Close** button.

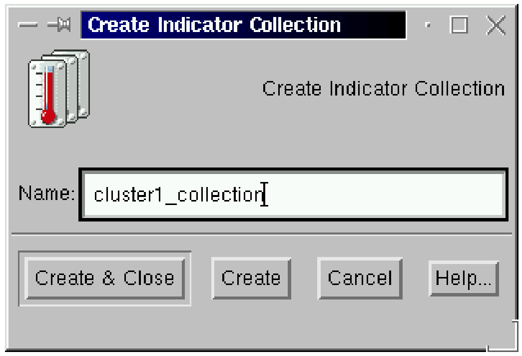

Figure 109. Create Indicator Collection window

3.3.1.5 Creating the subscribers

Cluster nodes must subscribe to the profile manager because they are targets for the monitors in ha_cluster. To subscribe the cluster nodes (arthur, merlin, and camelot) to the profile manager (ha_cluster), click on **ha_cluster** icon using the right mouse button. Then select **Subscribers...** as shown in Figure 110.

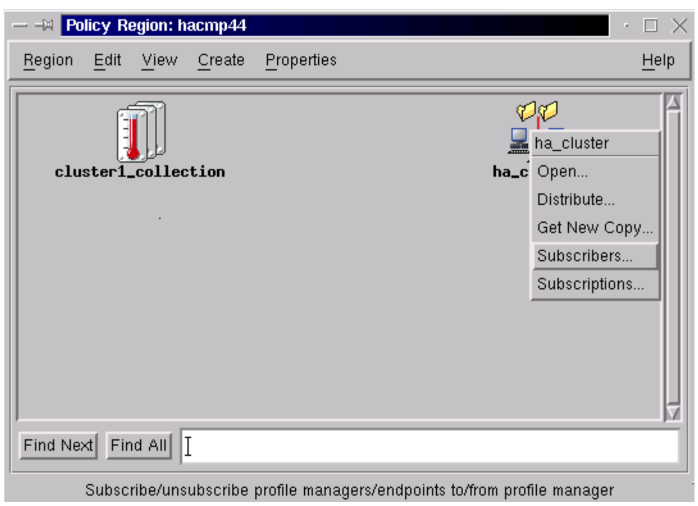

Figure 110. Opening the Subscribers window

Move the managed nodes (arthur, merlin, and camelot) from the *Available to become Subscribers* area to the *Current Subscribers* area, as shown in Figure 111 on page 106. Then click on the **Set Subscriptions & Close** button.

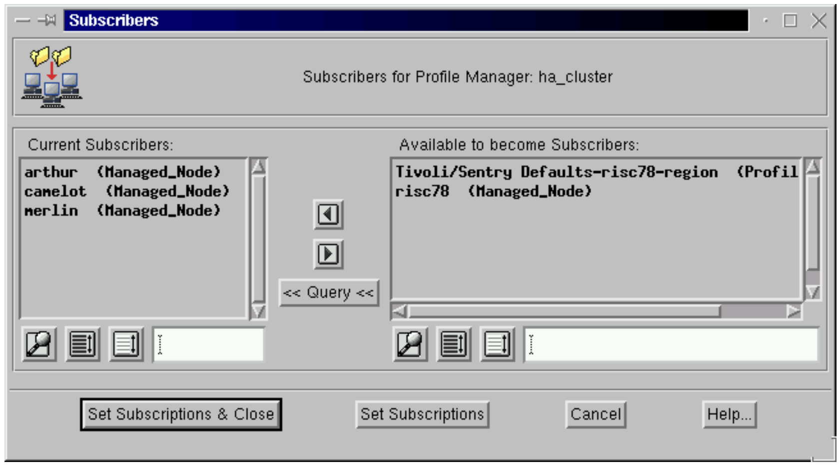

Figure 111. Subscribers window

3.3.2 Defining the administrative role for the managed nodes

Before running the HATivoli installation scripts on the cluster nodes, you have to assign the root user on cluster nodes to a Tivoli administrator.

The root user on the cluster nodes needs administrative role in the TMR environment to perform Tivoli administrative tasks; running Tivoli commands or executing Tivoli programs. The HATivoli installation scripts will perform several tasks in the TMR, but some of these tasks require administrative role.

To assign the root user to a Tivoli administrative role, double click on the Administrators icon on the TME Desktop window shown in Figure 103 on page 101:

You will see *Administrators* window shown in Figure 112 on page 107.

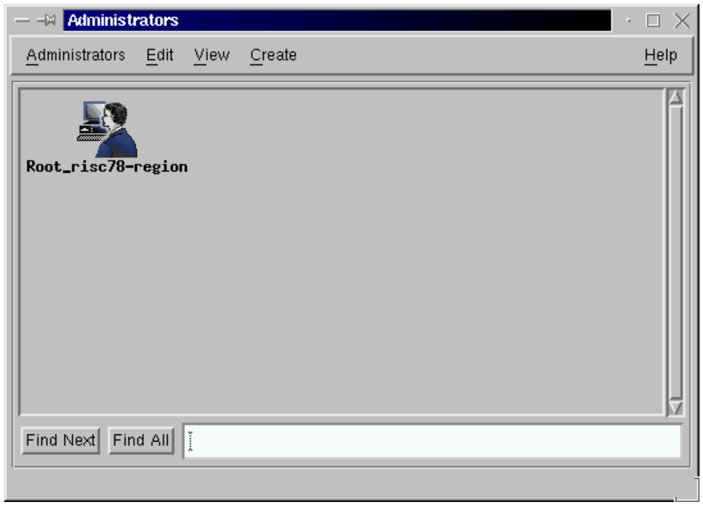

Figure 112. Administrators window

In this window there is one Tivoli administrator icon named Root_risc78-region. This Tivoli administrator is created at the installation time.

Click the right mouse button on the Root_risc78-region icon, select **Edit Logins...** as shown in Figure 113.

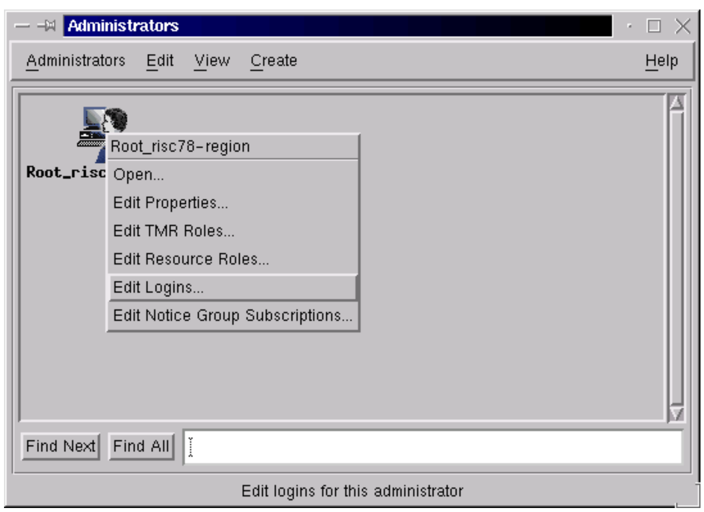

Figure 113. Administrators window (2)

You will see the *Set Login Names* window as shown in Figure 114.

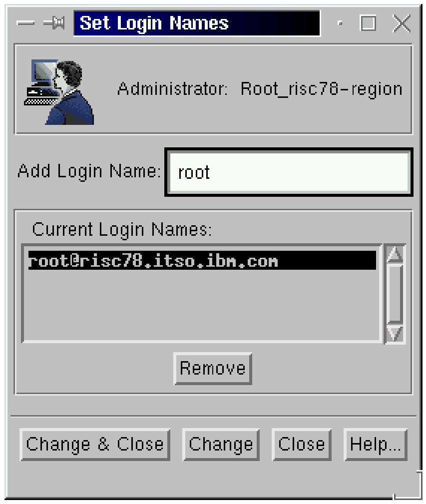

Figure 114. Set Login Names window

There is already one login name (root@risc78.itso.ibm.com) in *Current Login Names* area. When you installed TME 10 Framework on the TMR server (this is risc78.itso.ibm.com), installation program requires Tivoli administrative role for the installation operations. Therefore the program automatically assigned the root user on risc78.itso.ibm.com node to a Tivoli administrative role.

Now you need the root user on cluster nodes also assigned a Tivoli administrative role. To assign the role to the root user on cluster nodes, type "root" in the *Add Login Name* field, then press the enter key. If you do not specify a node name (for example, @arthur.itso.ibm.com, @merlin.itso.ibm.com, and so on), all the root user on cluster nodes[1] are able to perform a Tivoli administrative role.

The root user is added in the *Current Login Names* area in this window as shown in Figure 115 on page 109.

[1] In fact, specifying root without node name covers all the root users (include risc78) in the region. Therefore, removing login name "root@risc78.itso.ibm.com" does not cause any problem. This is the reason you do not need the step described in Section 3.3.2, "Defining the administrative role for the managed nodes" on page 106 in the procedure described in Section 3.5, "Adding a new cluster to an existing TMR" on page 129.

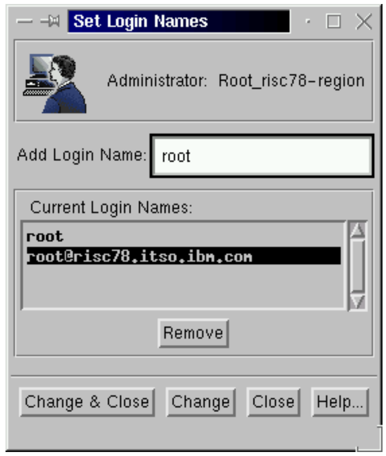

Figure 115. Set Login Names window (2)

Then click on the **Change & Close** button. To close Administrators window, select **Administrators** > **Close**.

3.3.3 installing HATivoli on TMR server and managed nodes

HATivoli LPP installation procedure is the same on both the TMR server and the managed nodes.

Insert the HACMP 4.4 CD in the cdrom drive and run the `installp` command on each node (risc78, arthur, merlin, and camelot) as follows:

```
# installp -acXd /dev/cd0 cluster.hativoli cluster.msg.?n_US.hativoli
>>>>>>> omitted lines <<<<<<<<<
. . . . . << End of copyright notice for cluster.msg.En_US.hativoli >>. . . .

Finished processing all filesets.  (Total time:  21 secs).

+-----------------------------------------------------------------------------+
                                Summaries:
+-----------------------------------------------------------------------------+

Installation Summary
--------------------
Name                        Level           Part        Event       Result
-------------------------------------------------------------------------------
cluster.hativoli.server     4.4.0.0         USR         APPLY       SUCCESS
cluster.hativoli.client     4.4.0.0         USR         APPLY       SUCCESS
cluster.msg.en_US.hativoli  4.4.0.0         USR         APPLY       SUCCESS
cluster.msg.En_US.hativoli  4.4.0.0         USR         APPLY       SUCCESS
#
```

3.3.4 Configuring HATivoli on cluster nodes

Configuration of HATivoli consists of two steps:

1. Creating the monitoring profiles and the corresponding indicators in the TMR
2. Creating and configuring the Extended Node Properties application

3.3.4.1 Creating monitoring profiles

This step performs the following tasks:

- Creates the monitoring profiles in the profile manager
- Creates the indicators in the indicator collection
- Links the monitoring profiles to all subscriber nodes
- Distributes the monitoring profiles to all subscriber nodes

To see if the `oserv` daemon is running on all managed nodes, issue the `odadmin` command as follows:

```
# odadmin odlist
```

Check if the status of each client node is <c> connected as described in Section 3.2.4, “Verifying Tivoli installation” on page 86.

Running install script

Login on one of the cluster nodes, arthur in our case, and execute the following command:

```
. /etc/Tivoli/setup_env.sh
```

This will set up the Tivoli environment for the installation script. Then issue the `install` command. It asks you a policy region name first as follows:

```
# /usr/sbin/hativoli/bin/install

Select Region
-------------

1...risc78-region
2...hacmp44


(Type 'quit' to abort installation)
Enter Selection:2
```

To select the policy region created in Section 3.3.1.1, “Creating policy region hacmp44” on page 97, type `2` and press enter key. Then it asks you a profile manager name as follows:

```
Select Profile Manager
----------------------

1...ha_cluster


(Type 'quit' to abort installation)
Enter Selection: 1
```

To select the ha_cluster profile manager created in Section 3.3.1.3, “Creating the profile manager” on page 101, type `1` and press the enter key. Then it asks you a indicator collection name as follows:

```
Select Indicator Collection
---------------------------

1...cluster1_collection


(Type 'quit' to abort installation)
Enter Selection: 1
```

To select the cluster1_collection indicator collection created in Section 3.3.1.4, “Creating the indicator collection” on page 103, type `1` and press enter key. Then it shows the following message and completes the tasks:

```
cat: 0652-050 Cannot open /usr/sbin/hativoli/ipaliases.conf.

Please wait... This operation can take several minutes.

#
```

Because we have a separate administrative network for Tivoli and do not use IP aliasing for this configuration, you can ignore the “0652-050” message.

In Section 3.5, “Adding a new cluster to an existing TMR” on page 129, we will show an example with IP aliasing.

Synchronizing cluster configuration

After creating monitoring profiles on one node, you need to synchronize cluster resources from this node by using the `smit clsyncnode.dialog` fastpath:

```
                      Synchronize Cluster Resources

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

[TOP]                                                    [Entry Fields]
  Ignore Cluster Verification Errors?                 [No]                +
  Un/Configure Cluster Resources?                     [Yes]               +
* Emulate or Actual?                                  [Actual]            +
* Skip Cluster Verification                           [No]                +

  Note:
  Only the local node's default configuration files
  keep the changes you make for resource DARE
  emulation. Once you run your emulation, to
  restore the original configuration rather than
  running an actual DARE, run the SMIT command,
  "Restore System Default Configuration from Active
  Configuration."
[MORE...3]

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
```

You can also perform cluster synchronization using the `cldare` command as follows:

```
# /usr/sbin/cluster/utilities/cldare -r
```

3.3.4.2 Creating the Extended Node Properties application

This step creates the Extended Node Properties application described in Section 3.4.1, “Monitoring cluster using HATivoli GUI” on page 114.

First of all, make sure that the perl package is installed on both the TMR server and the cluster nodes by issuing the `file` command:

```
# file /usr/bin/perl
/usr/bin/perl:  symbolic link to /usr/opt/perl5/bin/perl5.00503.
#
```

If you get the following message:

```
“/usr/bin/perl: 0653-901 Cannot get file status.”
```

This means you do not have perl package installed, or the link is missing. If this is the case, install the package as follows:

On the TMR server (rsic78)

The `/usr/sbin/hativoli/AEF/install` script will try to write in the `/usr/local/Tivoli` directory instead of `/tivoli` directory that we created. Therefore, you need to create a symbolic link from `/tivoli` directory to `/usr/local/Tivoli` as follows:

```
# ln -s /tivoli /usr/local/Tivoli
#
```

To finish this step on the TMR server, you need to execute the `/usr/sbin/hativoli/AEF/install` script as follows:

```
# /usr/sbin/hativoli/AEF/install
Creating resource-wide customization for dialog parent_dialog and resource Manag
edNode.
Creating resource-wide customization for dialog cl_node_cluster_wide_dialog and
resource ManagedNode.
Creating resource-wide customization for dialog cl_node_resource_group_dialog an
d resource ManagedNode.
Creating resource-wide customization for dialog cl_node_cluster_mgmt_dialog and
resource ManagedNode.
Creating resource-wide customization for dialog cl_node_specific_dialog and reso
urce ManagedNode.
Creating resource-wide customization for dialog cl_hativoli_msgbox and resource
ManagedNode.
#
```

On cluster nodes (arthur, merlin, and camelot)

Execute the `/usr/sbin/hativoli/AEF/install_aef_client` script on *each* cluster node:

```
# /usr/sbin/hativoli/AEF/install_aef_client
#
```

To complete this step, start the cluster services on *each* cluster node (if not already started).

3.3.5 Configuration verification

To verify configuration, double click on the hacmp44 policy region icon on the TME Desktop window.

You will see the *Policy Region:hacmp44* window shown in Figure 116 on page 114.

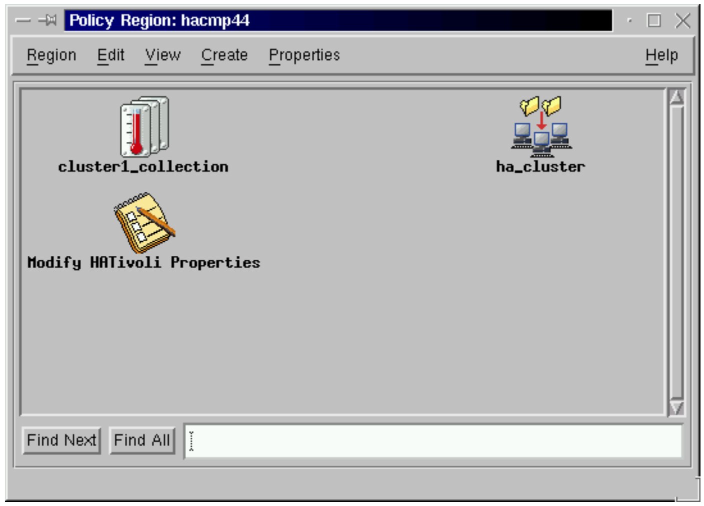

Figure 116. Policy Region:hacmp44 window with new Task Library icon

If you find the new Task Library icon named *Modify HATivoli Properties*, the installation and the configuration of HATivoli has been completed successfully.

3.4 Monitoring cluster with HATivoli

With HATivoli installed environment, there are two ways to monitor clusters. One way is using GUI provided by HATivoli and the other is using GUI provided by Tivoli. This section discusses the HATivoli GUI first, then the Tivoli GUI.

3.4.1 Monitoring cluster using HATivoli GUI

The monitoring GUI provided by HATivoli has been designed as a centralized interface for monitoring HACMP clusters. In addition, you can control HACMP clusters in the same manner using the SMIT menu.

When you install HATivoli, the HATivoli GUI called *Extended Node Properties* application is installed. This application uses the TME 10 AEF. The TME 10 AEF is a set of tools that enables you to extend the capabilities of TME 10 applications by adding fields to a dialog window, creating custom attributes and methods for application resources, and creating custom icons and bitmaps.

HATivoli GUI uses the information stored in the TMR database. The information is collected on the managed nodes by the monitors created at the installation time. These monitors are stored inside the indicator collections on the TMR server. In addition to the data retrieved from the TMR database, HATivoli GUI can run independent monitors (probes) and use some HACMP commands (for example `/usr/sbin/cluster/utilities/cllscf`) on the managed nodes.

To access the HATivoli Extended Node Properties window, double click on **ha_cluster** profile manager icon first. The *profile manager* window will appear as shown in Figure 117.

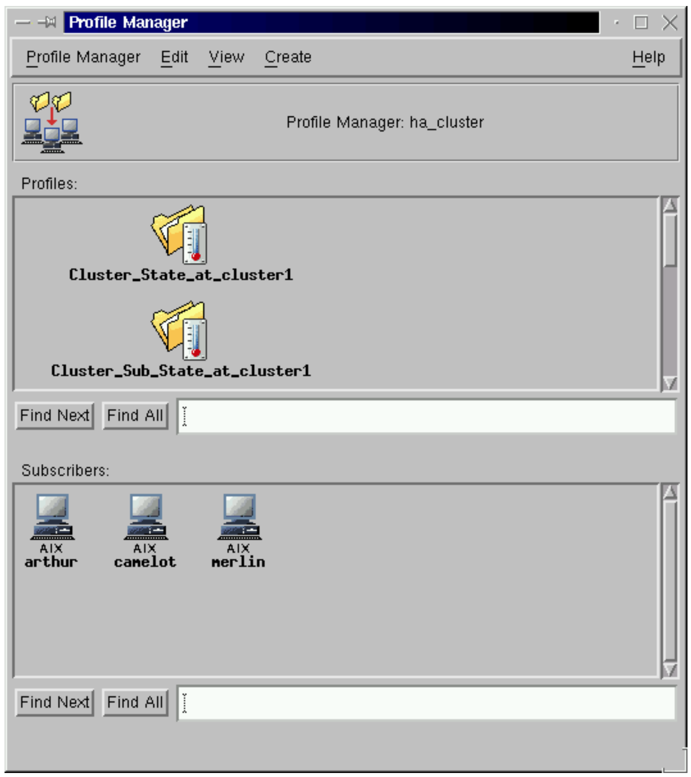

Figure 117. Profile Manager ha_cluster window

Then click and hold the right mouse button on one of the subscriber icons (for example, arthur) in the *Subscribers* area in the window as shown in Figure 118 on page 116, then select **Properties...**

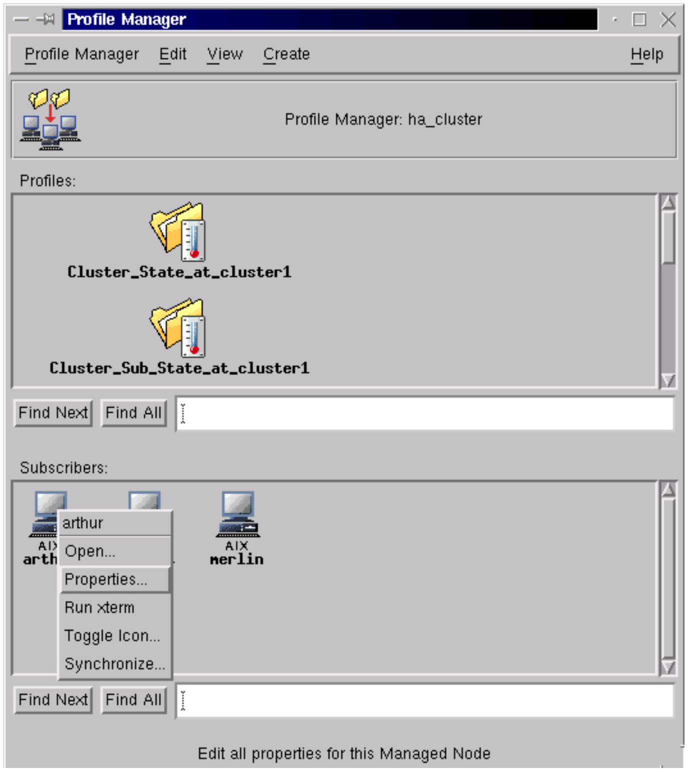

Figure 118. Opening Extended Node Properties window

The Cluster Managed Node window for the selected subscriber (arthur) appears as shown in Figure 119 on page 117.

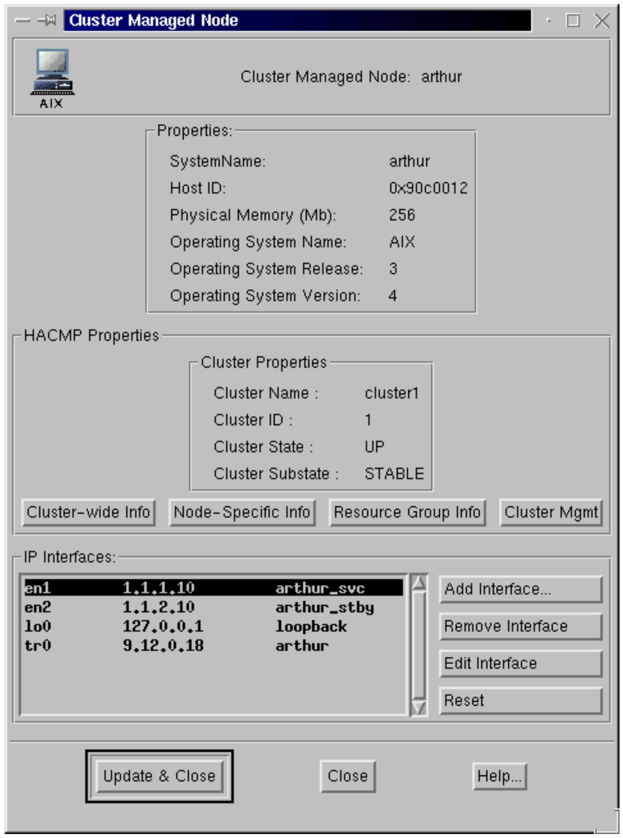

Figure 119. Extended Node Properties on node arthur

Using this window you can access useful information, such as cluster-wide, node-specific, resource group, and cluster management.

Cluster-wide information

To retrieve the cluster-wide information, click on **Cluster-wide info** button shown in Figure 119. This opens the Cluster Information Command window shown in the center of Figure 120 on page 118.

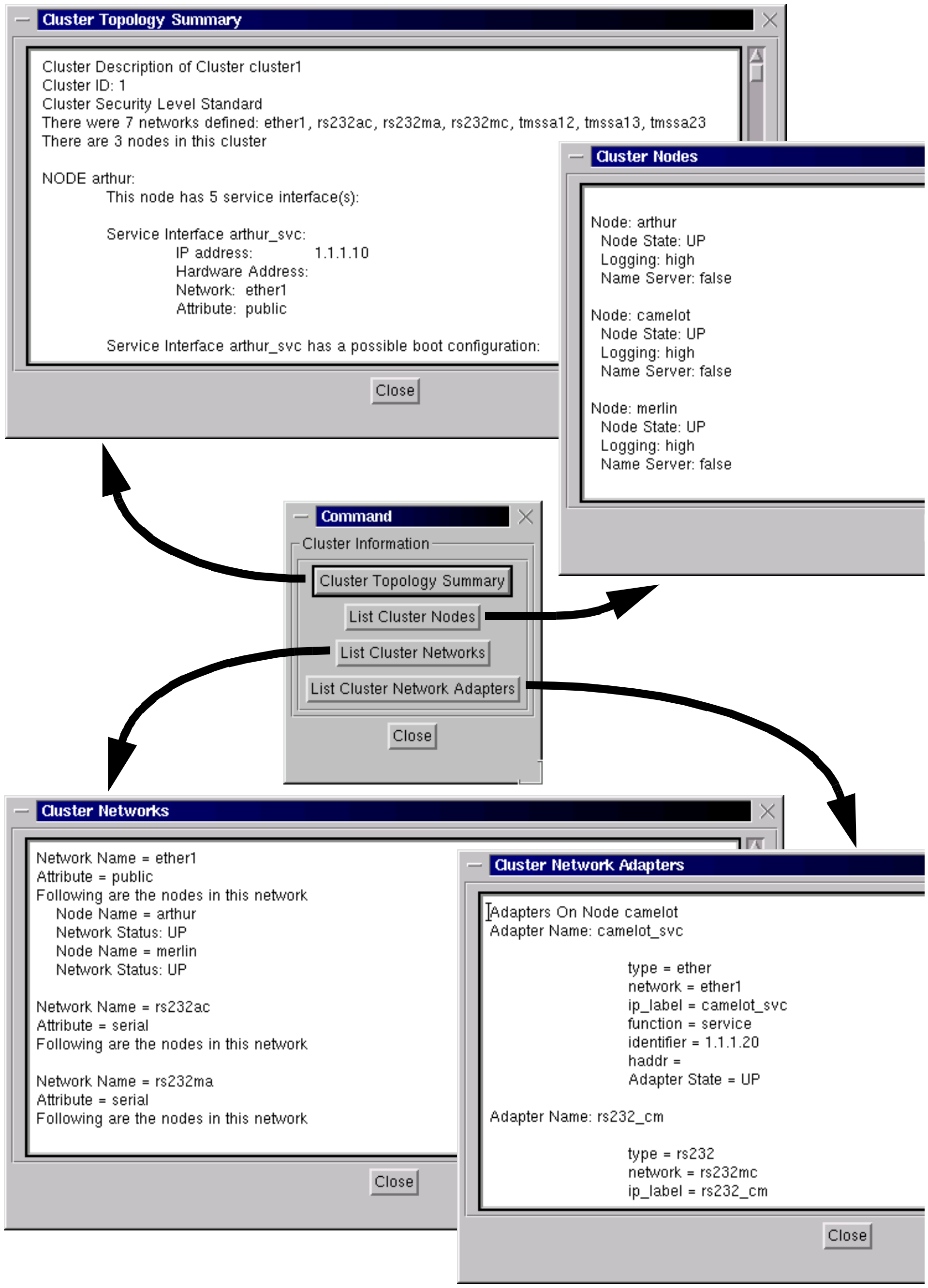

Figure 120. Cluster Information windows

For example, the first button, **Cluster Topology Summary**, opens a *Cluster Topology Summary* window and displays the output similar to the `/usr/sbin/cluster/utilities/cllscf` command.

Click on **List Cluster Nodes** button to get information about the nodes in the cluster and node parameters.

You can also list cluster network information and cluster network adapter information by clicking on **List Cluster Networks** and **List Cluster Network Adapters** button respectively.

Node-specific Information

To retrieve node-specific information, click on **Node-Specific Info** button shown in Figure 119 on page 117. You will see the *Node Specific Attributes Command* window shown in the center of Figure 121 on page 120.

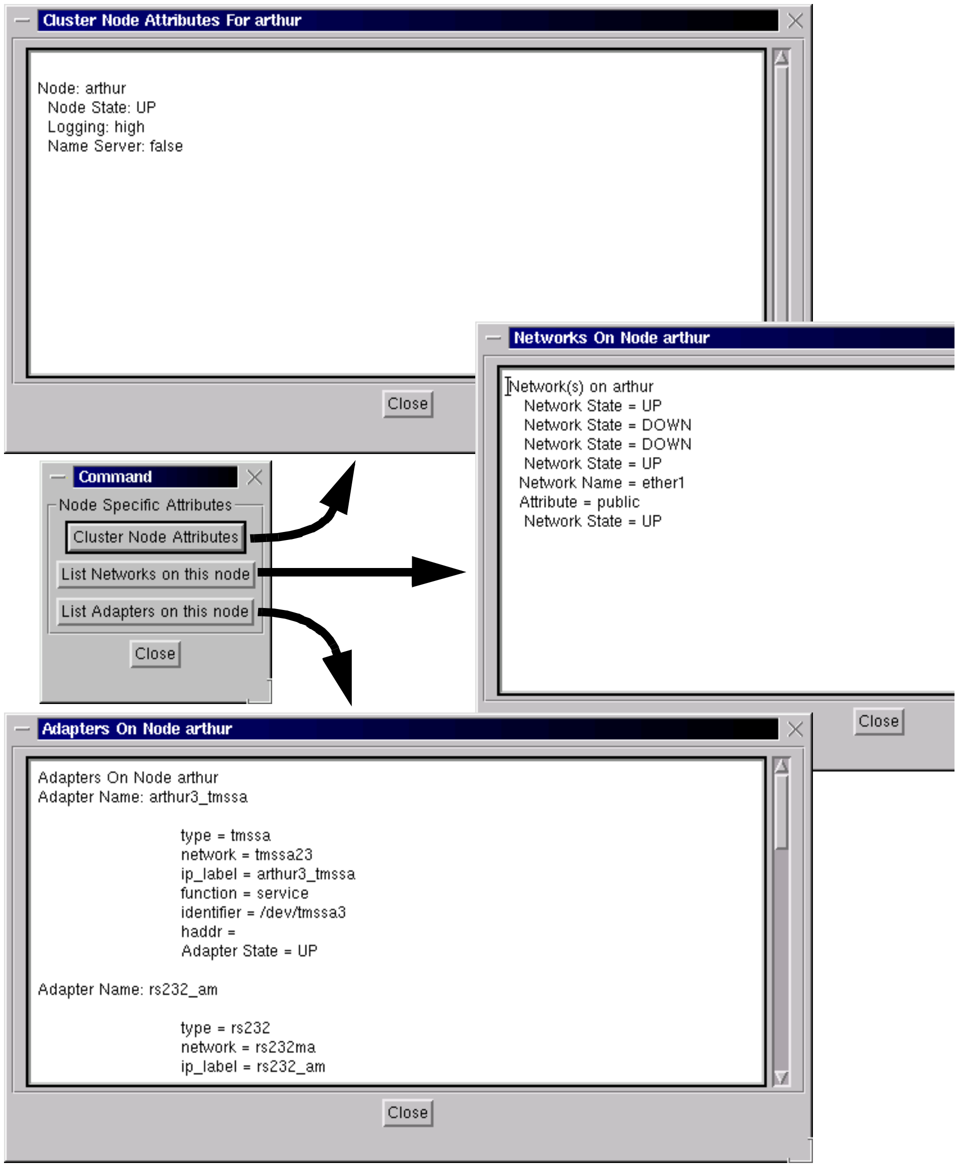

Figure 121. Node Specific Attributes windows

Resource Group information

To retrieve information about resource groups, their location, and status, click on the **Resource Group Info** button shown in Figure 119 on page 117. You will see the *Resource Group Information Command* window as shown in the center of Figure 122 on page 121.

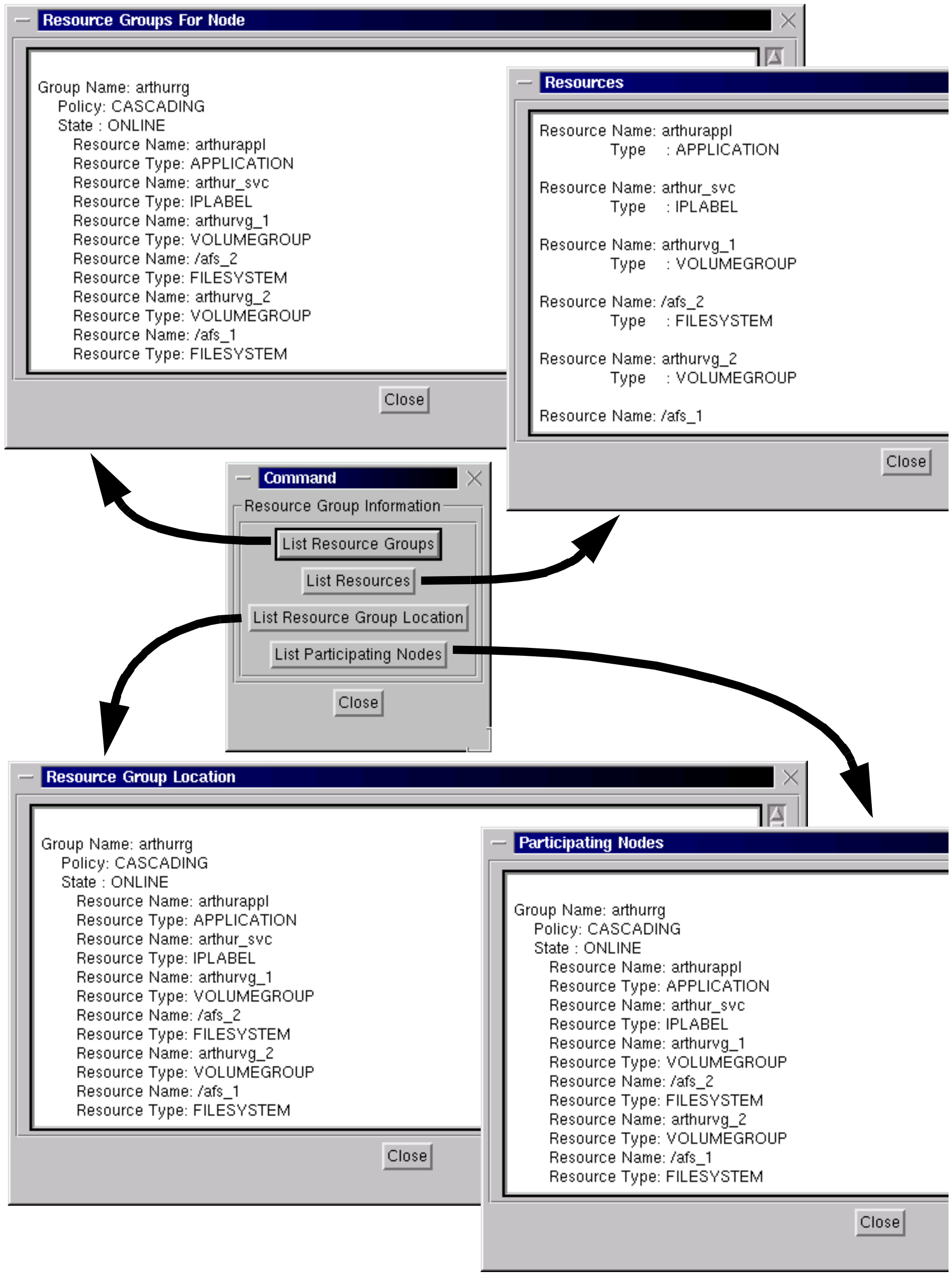

Figure 122. Resource Group Information windows

Attention

The information in the windows shown in Figure 122 on page 121 is available to HATivoli *only* for ES installations.

Cluster management

You can also manage the cluster. HATivoli provides you with the same functionality as the `smit hacmp` fastpath. Click on the **Cluster Mgmt** button shown in Figure 119 on page 117. You will see the *Cluster Management Command* window. Click on the **Open SMIT Window** button on the window. This operation provides you with the SMIT menu as shown in Figure 123.

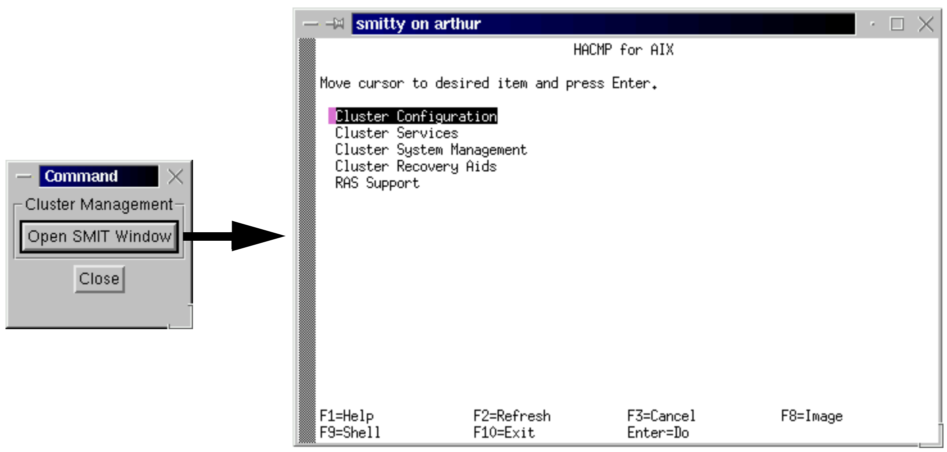

Figure 123. SMIT window on managed node

The ability to open a SMIT menu provides you with root access to the cluster node. Tivoli uses its own authentication services to access the remote system. The xterm opened for running SMIT is a Tivoli method that uses the oserv daemon for communication. Therefore you can access the SMIT menu even if the root password on the managed node is changed after the installation.

Network interface control

You can also perform network interface control; add, remove, edit, and reset.

For example, to edit the network interface properties, click on the **Edit Interface** button in Figure 119 on page 117. You will see the *Edit IP Interface* window:

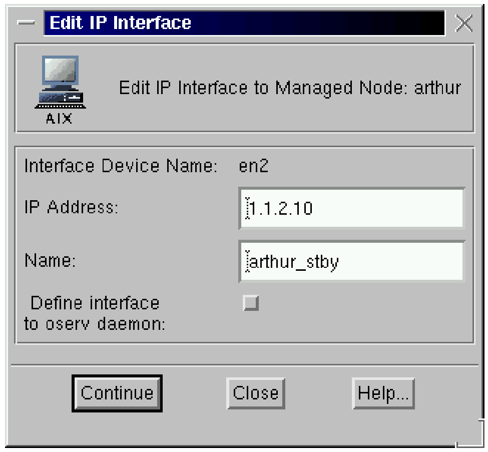

Figure 124. Edit IP interface window

Summary

Figure 125 on page 124 illustrates the relationship between the following windows:

- TME Desktop window
- Policy Region window
- Profile Manager window (see Figure 118 on page 116)
- Extended Node Properties window (see Figure 119 on page 117)
- Cluster Information window (see Figure 120 on page 118)
- Node Specific Attributes window (see Figure 121 on page 120)
- Resource Group Information window (see Figure 122 on page 121)
- Cluster Management window (see Figure 123 on page 122)
- Edit IP Interface window (see Figure 124)

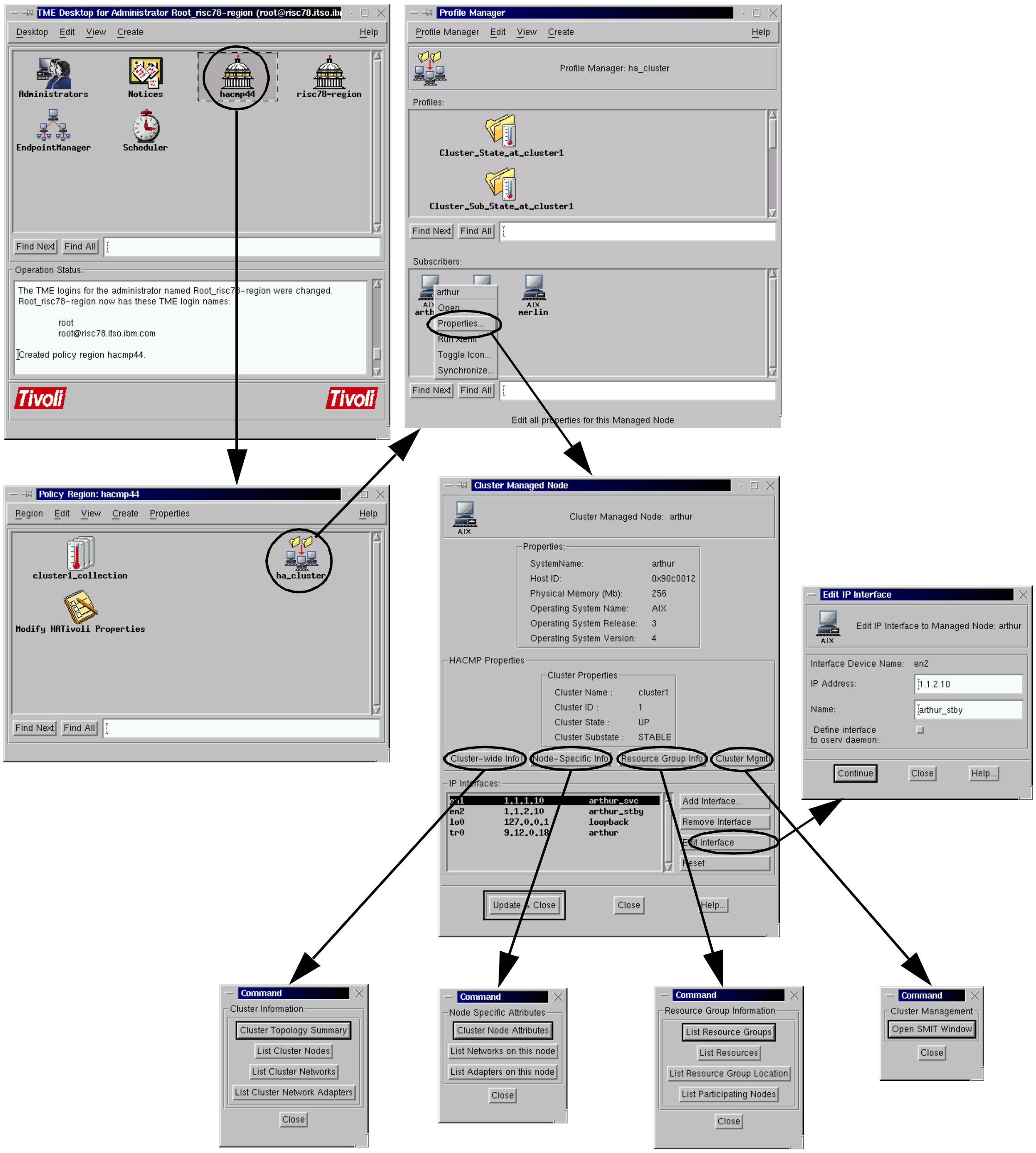

Figure 125. GUI provided by HATivoli

3.4.2 Monitoring cluster using Tivoli GUI

You can also use the GUI and applications provided by Tivoli to monitor the HACMP cluster. Because all the information about the HACMP cluster is collected and stored in the TMR database by TME 10 Distributed Monitoring, it is available for accessing.

Inside HATivoli

Monitoring is performed using the indicator collection. The indicator collection contains several indicators, each of which reflects a state that is collected by the HATivoli scripts (HACMP monitors) running on the managed nodes.

To monitor HACMP cluster (cluster1) using Tivoli GUI, open the *Policy Region: hacmp44* window shown in Figure 126.

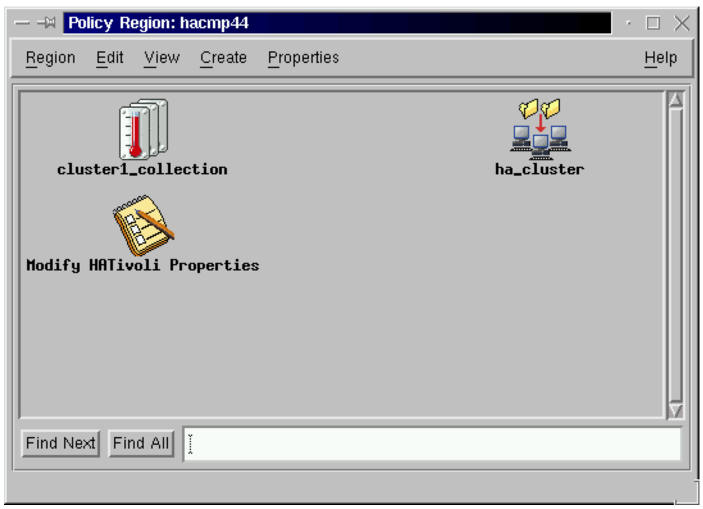

Figure 126. Policy Region:hacmp44 with temperature rise icon

If a problem occurs in the cluster1, the *cluster1_collection* icon in the window indicates "temperature rise" as shown in Figure 127.

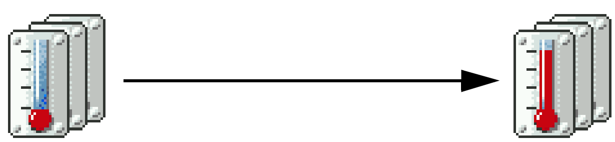

Figure 127. Temperature rise

To see which indicator is reporting the problem, double click on this icon. The *Distributed Monitoring Collection:cluster1_collection* window shown in Figure 128 appears.

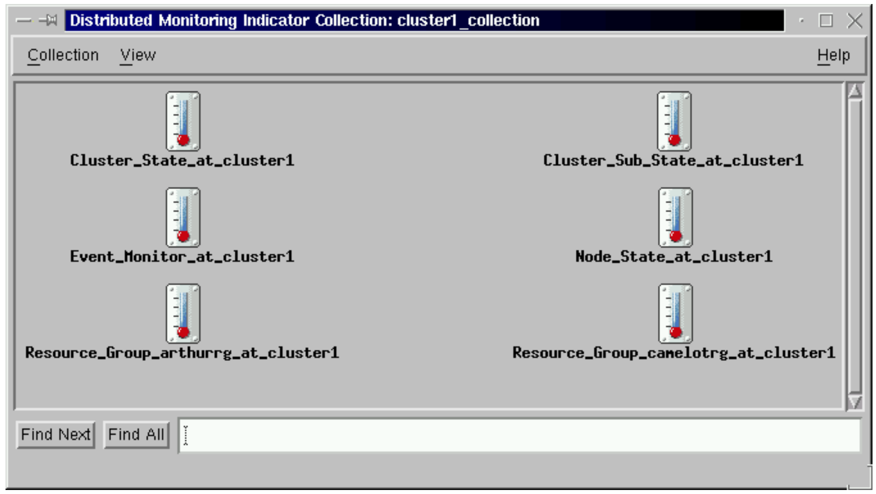

Figure 128. Distributed Monitoring Collection: cluster1_collection window

Inside HATivoli

The cluster1_collection indicator collection contains an indicator for each monitoring profile subscribed in the hacmp44 policy region by the HATivoli installation.

Look for the indicator icon that indicates “temperature rise." The indicator icons reflect varying degrees of severity of problems, depending on the height of the red color in the thermometer and the color-code marker alongside it. This is illustrated in Figure 129 on page 127.

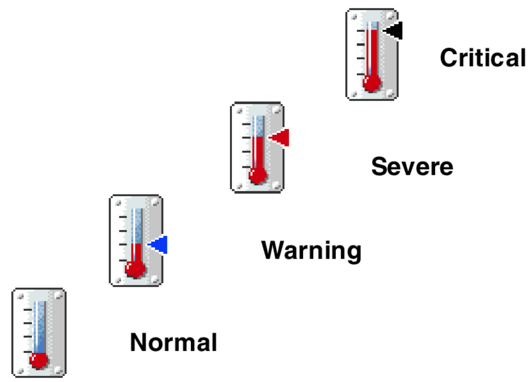

Figure 129. Severity and the height of the red color in the thermometer

The Figure 130 illustrates indicator displays for various cluster component states.

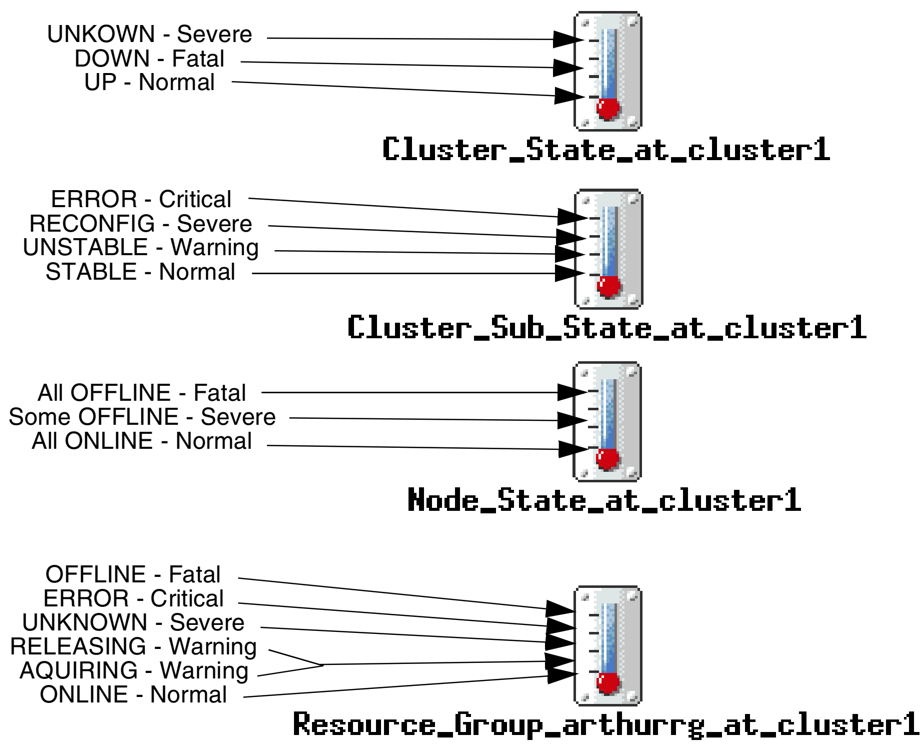

Figure 130. Description of cluster indicator icons

Note

Event Monitor icon is not implemented. For details, refer to Section 3.6.2, “How to monitor HACMP state information?” on page 144.

HATivoli provides also a Task Library that contains various tasks to modify monitoring profiles, distribute them, and synchronize configuration.

Double click on **Modify HATivoli Properties** icon in Figure 126 on page 125. The *Task Library:Modify HATivoli Properties* window appears, as shown in Figure 131.

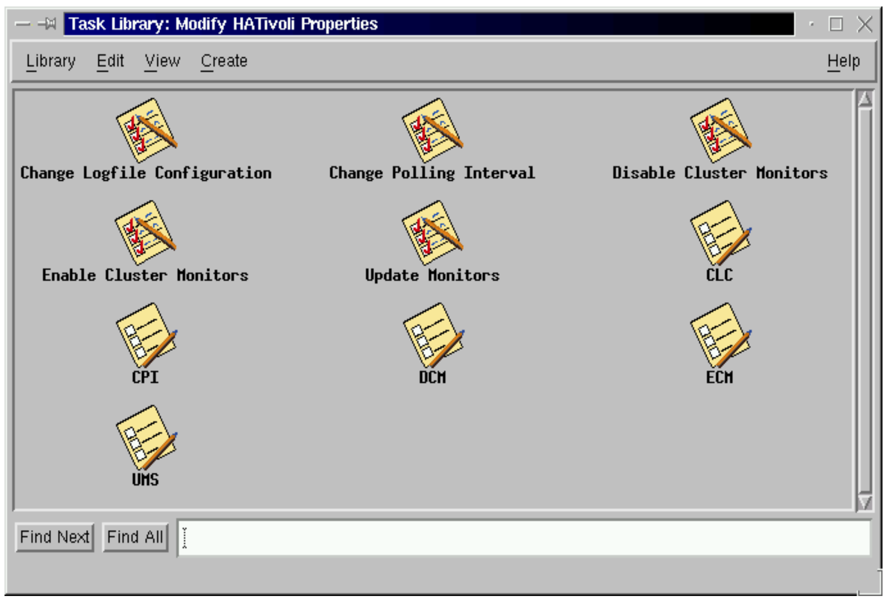

Figure 131. Modify HATivoli Properties window

You can perform the jobs and tasks corresponding to the icons in this window.

Summary

Figure 132 on page 129 illustrates the relationship between the following windows:

- TME Desktop window
- Policy Region window (see Figure 126 on page 125)
- Indicator Collection window (see Figure 128 on page 126)

- Task Library window (see Figure 131 on page 128)

These windows are discussed in this section.

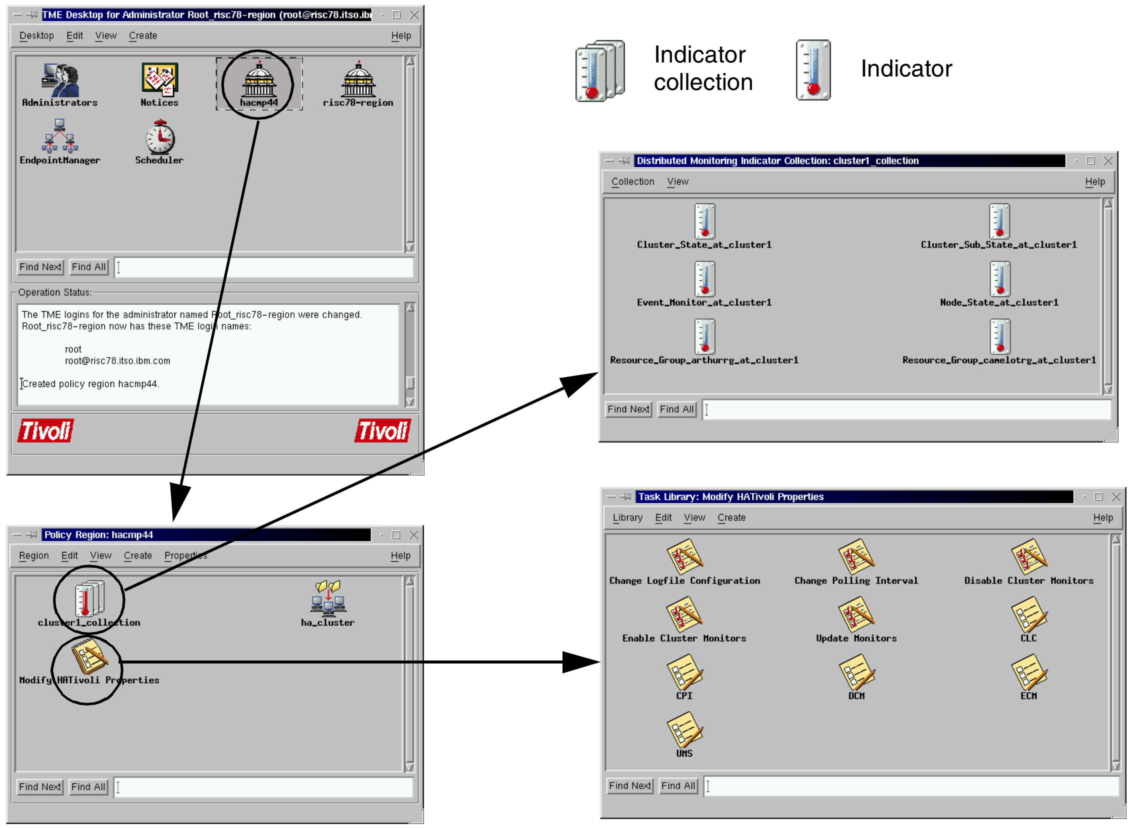

Figure 132. GUI provided by Tivoli

3.5 Adding a new cluster to an existing TMR

This section describes how to add another HACMP cluster to an exiting TMR environment. We are going to add an HACMP cluster (haes44) to the TMR created in Section 3.2, “Tivoli installation” on page 68. The cluster haes44 has two cluster nodes; trisc1 and trisc2. They are managed as managed nodes in the TMR. This scenario is illustrated in Figure 133 on page 130.

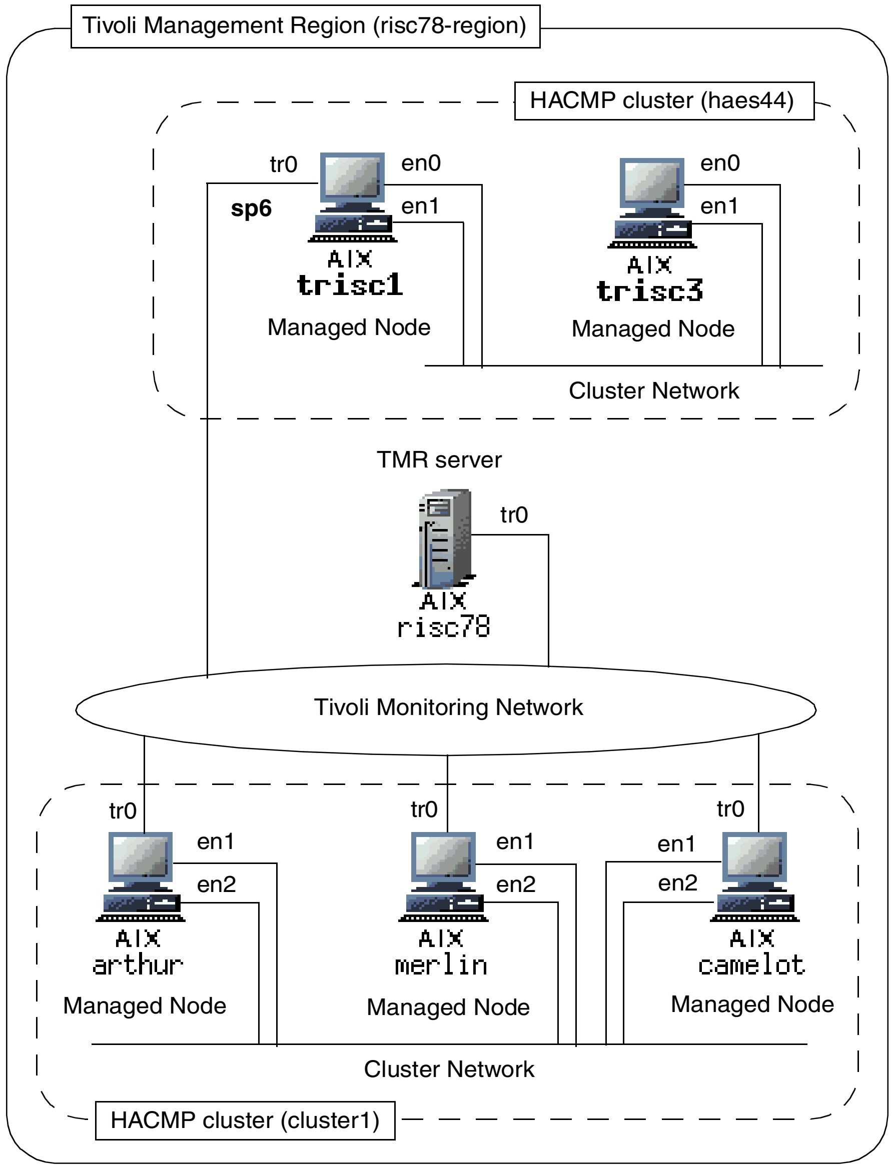

Figure 133. Adding another HACMP cluster to the TMR

Unlike the cluster1 cluster, the haes44 cluster does not have a separate network for Tivoli management. Therefore, we use IP aliasing on the standby network adapters on the cluster nodes.

The previous sections uses GUI for install operations. This section uses Tivoli Command Line Interface (CLI) commands and UNIX shell scripts instead.

Attention

All the operations in this chapter are performed as root user on the TMR server unless otherwise specified.

3.5.1 Preparations

Before starting installation, make the following preparations.

Adding IP aliases to standby adapters

Add the alias names and addresses for the standby adapters to /etc/hosts file on each node (trisc1 and trisc3):

```
10.50.50.1      trisc1
10.50.50.3      trisc3
```

Configure the IP alias on each node as follows:

```
# chdev -l en1 -a alias4=trisc1,255.255.255.0 //on node trisc1
```

and:

```
# chdev -l en1 -a alias4=trisc3,255.255.255.0 //on node trisc3
```

Note

Do not use `ifconfig en1 alias trisc1 netmask 255.255.255.0`, because this alias will be deleted when the machine is rebooted. The `chdev` command changes the ODM, so IP aliases will remain in the system configuration.

We use node trisc1 as a TCP/IP router between the Tivoli management network and the node trisc3, which is not connected to the Tivoli management network directly.

On trisc1, add two routes. One from the cluster internal (10.50.50) network to the Tivoli management network (9.12.0), and the other going the other way:

```
# chdev -l inet0 -a route =net,10.50.50,9.12.0.6,1,255.255.255.0
# chdev -l inet0 -a route =net,9.12.0,10.50.50.1,1,255.255.255.0
```

On trisc3, make trisc1 the default gateway:

```
# mktcpip -h risc3 -a 10.10.10.3 -m 255.255.255.0 -i en0 \
> -g 10.50.50.1 -t bnc
```

Check the communication between TMR server and each cluster nodes.

The following is the /etc/hosts file in our environment and network interface list on risc78, risc1, and risc3.

```
risc78 # cat /etc/hosts
9.12.0.78       risc78 risc78.itso.ibm.com  #this host
#---------------Rest of the world-------------------------------
# Tivoli region hosts
9.12.0.18       arthur
9.12.0.19       camelot
9.12.0.50       merlin
9.12.0.6        sp6
#--------------------------------------------------------------
# cluster1 network
1.1.1.10        arthur_svc
1.1.1.11        arthur_boot     arthur
1.1.1.20        camelot_svc
1.1.1.21        camelot-boot    camelot
1.1.2.20        camelot_stby
1.1.1.30        merlin_svc      merlin
1.1.2.30        merlin_stby
1.1.2.10        arthur_stby
#---------------------------------------------------------------
# haes44 network
10.10.10.1      risc1_svc       risc1
10.10.10.10     risc1_boot
10.20.20.1      risc1_stby
10.10.10.3      risc3_svc       risc3
10.10.10.30     risc3_boot
10.20.20.3      risc3_stby
#---------------------------------------------------------------
# Aliases for haes44 cluster
10.50.50.1      trisc1
10.50.50.3      trisc3
#---------------------------------------------------------------
risc78 #
```

```
risc78 # netstat -i
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
lo0   16896 link#1                          761396     0   761396     0     0
lo0   16896 127         loopback            761396     0   761396     0     0
lo0   16896 ::1                             761396     0   761396     0     0
tr0   1492  link#2      10.0.5a.b1.c4.2f   5845772     0  6370811     0     0
tr0   1492  9.12        risc78             5845772     0  6370811     0     0
risc78 #
```

```
risc1 # netstat -i
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
lo0   16896 link#1                          279743     0   281000     0     0
lo0   16896 127         loopback            279743     0   281000     0     0
lo0   16896 ::1                             279743     0   281000     0     0
en0   1500  link#2      2.60.8c.2d.20.f5    225207     0   223713     0     0
en0   1500  10.10       risc1_boot          225207     0   223713     0     0
en1   1500  link#3      2.60.8c.2c.d1.4a    120776     0   136714     0     0
en1   1500  10.20       risc1_stby          120776     0   136714     0     0
en1   1500  10.50       trisc1              120776     0   136714     0     0
tr0   1492  link#4      10.0.5a.b1.b5.2d    801606     0   413945     0     0
tr0   1492  9.12        sp6                 801606     0   413945     0     0
risc1 #
```

```
risc3 # netstat -i
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
lo0   16896 link#1                          122783     0   125336     0     0
lo0   16896 127         loopback            122783     0   125336     0     0
lo0   16896 ::1                             122783     0   125336     0     0
en0   1500  link#2      2.60.8c.2e.88.34    393943     0   283667     0     0
en0   1500  10.10       risc3_svc           393943     0   283667     0     0
en1   1500  link#3      2.60.8c.2d.7.ec     638076     0   402202     0     0
en1   1500  10.20       risc3_stby          638076     0   402202     0     0
en1   1500  10.50       trisc3              638076     0   402202     0     0
risc3 #
```

File system

We created the separate file system mounted under /tivoli as described in Section 3.2.1, “Preparations” on page 70.

Backup

We strongly recommend you back up the TMR database as described in Section 3.2.4.1, “Back up the TMR database” on page 87.

3.5.2 Adding managed nodes to the TMR

To add managed nodes, you need to create a file that contains the nodes to be added. We created the haes44.nodes file in the /tivoli directory. The file contains the node names, user names, and passwords for both cluster nodes as follows:

```
# cat /tivoli/haes44.nodes
trisc1,root,root_password_risc1
trisc3,root,root_password_risc3
#
```

Insert the TME 10 Framework 3.6 CD in the cdrom drive on TMR server, and mount the /cdrom file system. Execute the `wclient` command as follows:

```
# wclient -c /cdrom -f /tivoli/haes44.nodes \
> BIN=/tivoli/bin \
> LIB=/tivoli/lib \
> DB=/tivoli \
> MAN=/tivoli/man \
> APPD=/usr/lib/X11/app-defaults \
> CAT=/tivoli/cat \
> Autostart=1 \
> SetPort=1 \
> CreatePaths=1

Inspecting node trisc1...
Inspecting node trisc3...

Unless you cancel, the following operations will be executed:
  For the machines in the independent class:
    hosts(trisc1, trisc3)
>>>>>>>>> omitted lines <<<<<<<<<<
need to copy the architecture specific Framework files to:
         trisc3:/tivoli/trisc3.db
>>>>>>>>> omitted lines <<<<<<<<<<
Continue([y]/n)? y
>>>>>>>>> omitted lines <<<<<<<<<<
...Client install completed successfully.
 completed.

Finished client installation.
#
```

When the installation program prompts you to continue, type `y` and press enter to continue the installation.

When you finish the installation, the nodes are stored in the TMR database. To check this, execute the `wls` command as follows:

```
# wls risc78-region
risc78
arthur
merlin
camelot
trisc1
trisc3
#
```

Figure 134. Listing nodes in the TMR

You should find both trisc1 and trisc2 in the output.

Refer to Section 3.2.3, “Adding managed nodes to the TMR” on page 77 for GUI operation.

3.5.3 Verifying Tivoli installation

When the installation process is finished, the Framework software is installed on the new clients and the oserv daemon is started. You can verify the installation by issuing the `odadmin` command:

```
# odadmin odlist
Region      Disp  Flags  Port          IPaddr   Hostname(s)
1232341272     1    ct-    94       9.12.0.78   risc78.itso.ibm.com,risc78
              16    ct-    94       9.12.0.18   arthur
              17    ct-    94       9.12.0.50   merlin
              18    ct-    94       9.12.0.19   camelot
              19    ct-    94        9.12.0.6   sp6
              20    ct-    94      10.50.50.3   trisc3
#
```

For more information on the `odadmin` command, refer to Section 3.2.4, “Verifying Tivoli installation” on page 86.

Note that, although the node name for the first node is set to *trisc1* (IP alias on the standby adapter), the output of this command shows the name of the interface the TMR server is directly connected to; *sp6*. In the TMR database, the node is stored as trisc1, as you can see in Figure 134 on page 134.

3.5.4 Installing TME 10 Distributed Monitoring

In this step we install TME 10 Distributed Monitoring software on managed nodes; trisc1 and trisc3.

Insert the Distributed Monitoring CD in the cdrom drive on TMR server, and mount the /cdrom file system.

Then execute the `winstall` command as follows:

```
# winstall -c /cdrom -i SENT36.IND INCLUDE=/tivoli/include trisc1 trisc3
Checking product dependencies...
 product TMF_3.6 is already installed as needed.
Dependency check completed.
Inspecting node trisc1...
Inspecting node trisc3...
Installing product: TME 10 Distributed Monitoring 3.6

Unless you cancel, the following operations will be executed:
>>>>>>>>> omitted lines <<<<<<<<<<<<
Continue([y]/n)? y
Executing queued operation(s)
Distributing architecture specific Sentry engine bootstrap for trisc1
 ..Product install completed successfully.
 completed.
Distributing architecture specific Sentry engine bootstrap for trisc3
 ...Product install completed successfully.
 completed.
Registering product installation attributes...registered.
Finished product installation.
#
```

When the installation program prompts you to continue, type `y` and press enter to continue the installation.

Refer to Section 3.2.5, “Installing TME 10 Distributed Monitoring” on page 89 for GUI operation.

3.5.5 Installing patches to the Tivoli products

Install patches to TME 10 software using the procedure described in Section 3.2.7, “Installing patches on the Tivoli products” on page 93.

Attention

After each patch installation you *must* refresh the `oserv` daemon on TMR server and all managed nodes. You can do this by issuing the `odadmin` command on the TMR server:

```
# odadmin reexec all
Waiting for clients to finish reexec before doing TME server..
Clients have resumed operation. Reexec'ing TME server.
#
```

This operation will restart `oserv` daemon on all clients first, and than on the TME Server.

After the software update procedure is finished, back up the TMR database as described in Section 3.2.4.1, “Back up the TMR database” on page 87.

3.5.6 Configuring the TMR for HATivoli

There are two policy region already created. The risc78-region policy region was created when we installed TME 10 Framework on the TMR server. We created hacmp44 policy region to manage HACMP cluster in Section 3.3.1.1, “Creating policy region hacmp44” on page 97. You can use one of these policy regions, or create a new policy region. We decided to use the hacmp44 policy region because we have to manage the same type of resources.

There is one profile manager already created in hacmp44 policy region. This is ha_cluster profile manager. We created it in Section 3.3.1.3, “Creating the profile manager” on page 101. Because we want to manage each cluster separately, we decided to create a new profile manager for haes44 cluster. Its profile manager name is haes44_cluster.

Corresponding to this profile manager, we create a new indicator collection named haes44. This collection will be used to store the execution results of the monitoring profiles corresponding to haes44 cluster. We also subscribed nodes trisc1 and trisc3 to this profile manager.

The following screen lists the commands used to create the Tivoli objects discussed in this section:

```
# wcrtprfmgr @hacmp44 haes44_cluster
# wcrtsntcoll hacmp44 haes44
# wsub @haes44_cluster @ManagedNode:trisc1 @ManagedNode:trisc3
#
```

Refer to Section 3.3.1, “Configuring the TMR for HATivoli” on page 96 for GUI operation.

3.5.7 HATivoli installation on managed nodes

We installed HATivoli software on the cluster nodes as described in Section 3.3.3, “installing HATivoli on TMR server and managed nodes” on page 109.

After this installation we added two files needed for HATivoli IP aliases; /etc/wlocalhost and /usr/sbin/hativoli/ipaliases.conf. These files are required on each cluster node.

On node trisc1 we have:

```
risc1 # cat /etc/wlocalhost
trisc1
risc1 # cat /usr/sbin/hativoli/ipaliases.conf
ennetwork
risc1       trisc1
risc3       trisc3
risc1 #
```

and on node trisc3:

```
risc3 # cat /etc/wlocalhost
trisc3
risc3 # cat /usr/sbin/hativoli/ipaliases.conf
ennetwork
risc1       trisc1
risc3       trisc3
risc3 #
```

The /etc/wlocalhost file contains the name of the local IP address used for Tivoli communication. The /usr/sbin/hativoli/ipaliases.conf contains the cluster network name used for monitoring, and the relationship between the hostnames and IP aliases defined for Tivoli communication.

3.5.8 Configuring HATivoli on cluster nodes

Run the HATivoli installation scripts as described in Section 3.3.4, “Configuring HATivoli on cluster nodes” on page 110.

For the step described in Section 3.3.4.1, “Creating monitoring profiles” on page 110, it is important to synchronize cluster configuration after running the `/usr/sbin/hativoli/bin/install` script. This script creates some custom events needed for IP alias reconfiguration in case of a cluster event.

For the step described in Section 3.3.4.2, “Creating the Extended Node Properties application” on page 112, you only need to run the following script on each cluster node (trisc1 and trisc3):

```
# /usr/sbin/hativoli/AEF/install_aef_client
#
```

This operation is performed to make the clients (trisc1 and trisc3) aware of the Extended Node Properties application existing on the TMR server.

3.5.9 Configuration verification

To verify configuration, double click on the hacmp44 icon on the TME 10 Desktop window. This opens Policy Region:hacmp44 window as shown in Figure 135.

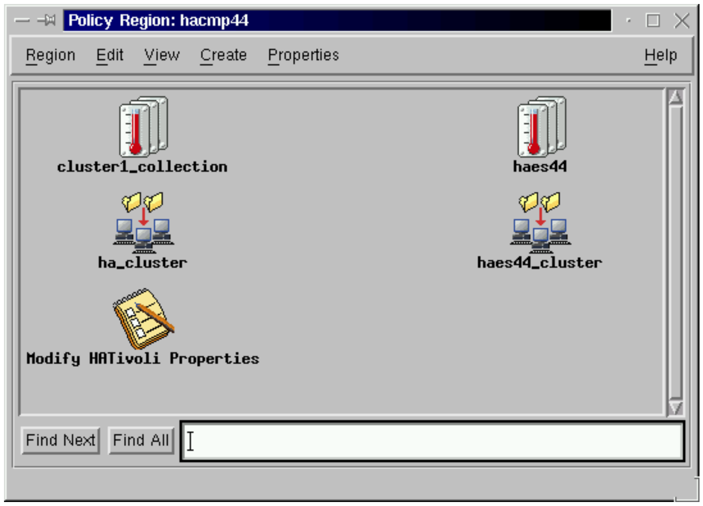

Figure 135. Policy Region window with haed44_cluster profile manager

In this window, you can find a new profile manager icon (haes44_cluster) and indicator collection icon (haes44).

Then double click on haes44_cluster profile manager icon and check if the monitoring profiles and subscribers exist as shown in Figure 136 on page 140.

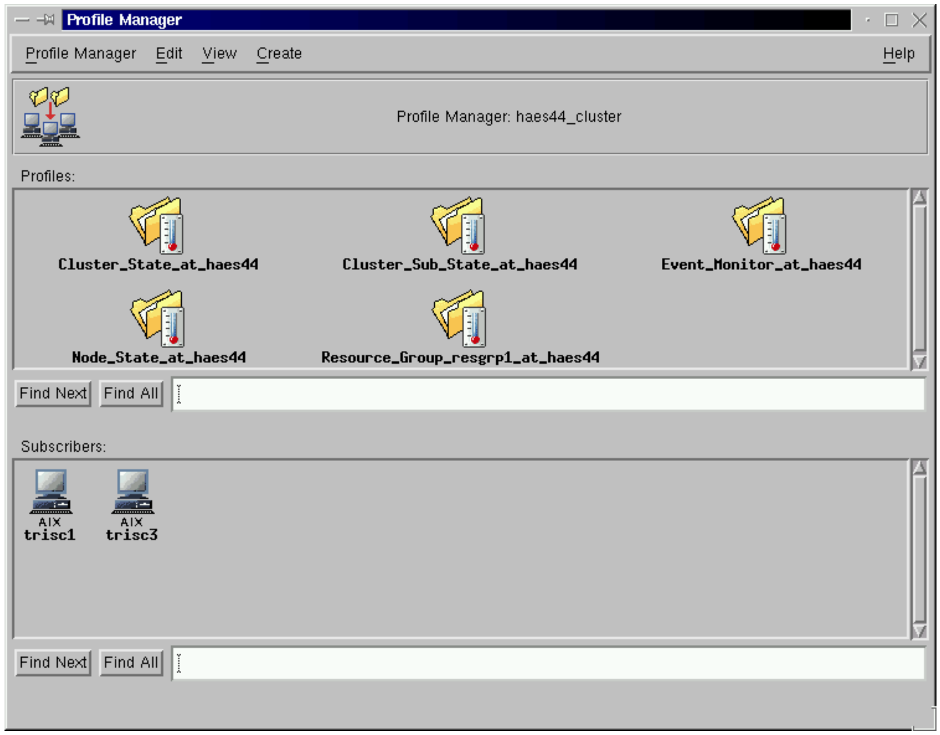

Figure 136. Profile Manager haes44_cluster window

3.6 Advanced topics about HATivoli

This section describes how HACMP and Tivoli work together using HATivoli. Topics include SNMP considerations, monitoring profiles, and monitoring profiles customization.

3.6.1 Collecting HACMP state information

An HACMP cluster is dynamic and can undergo various transitions in its state over time. To collect these HACMP state information, HATivoli uses its own Simple Network Management Protocol (SNMP) Management Information Base (MIB) class.

SNMP is an industry-standard specification for monitoring and managing TCP/IP-based networks. SNMP includes a protocol, a database specification, and a set of data objects. A set of data objects forms a MIB. SNMP provides a standard MIB that includes information such as IP addresses and the number of active TCP connections. The actual MIB definitions are encoded into the agents running on a system. The standard SNMP agent is the *snmpd* daemon.

SNMP can be extended through the use of the SNMP Multiplexing (SMUX) protocol to include enterprise-specific MIBs that contain information relating to a discrete environment or application. The SMUX peer daemon maintains information about the objects defined in its MIB.

The Cluster SMUX Peer daemon, *clsmuxpd*, maintains cluster status information in a special HACMP MIB. When clsmuxpd starts on a cluster node, it registers with the SNMP daemon, snmpd, and then continuously collects HACMP state information from the Cluster Manager daemon, *clstrmgr*. The clsmuxpd maintains an updated topology map of the cluster in the HACMP MIB as it tracks events and resulting states of the cluster. The information collected is made available over the network interface by the snmpd daemon. These relationship is illustrated in Figure 137.

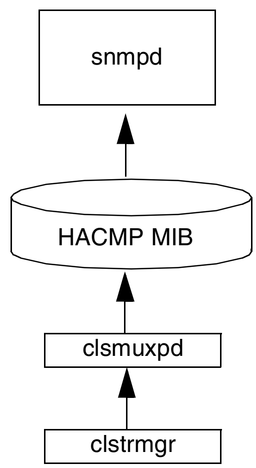

Figure 137. Collecting HACMP state information

HACMP MIB

The HACMP MIB definition consists of two files:

/usr/sbin/cluster/hacmp.defs
This file contains the compiled structure of the HACMP enterprise specific MIB.

/usr/sbin/cluster/hacmp.my
This file contains the definitions of the variables in the HACMP MIB.

The graph in Figure 138 illustrates the HACMP MIB structure.

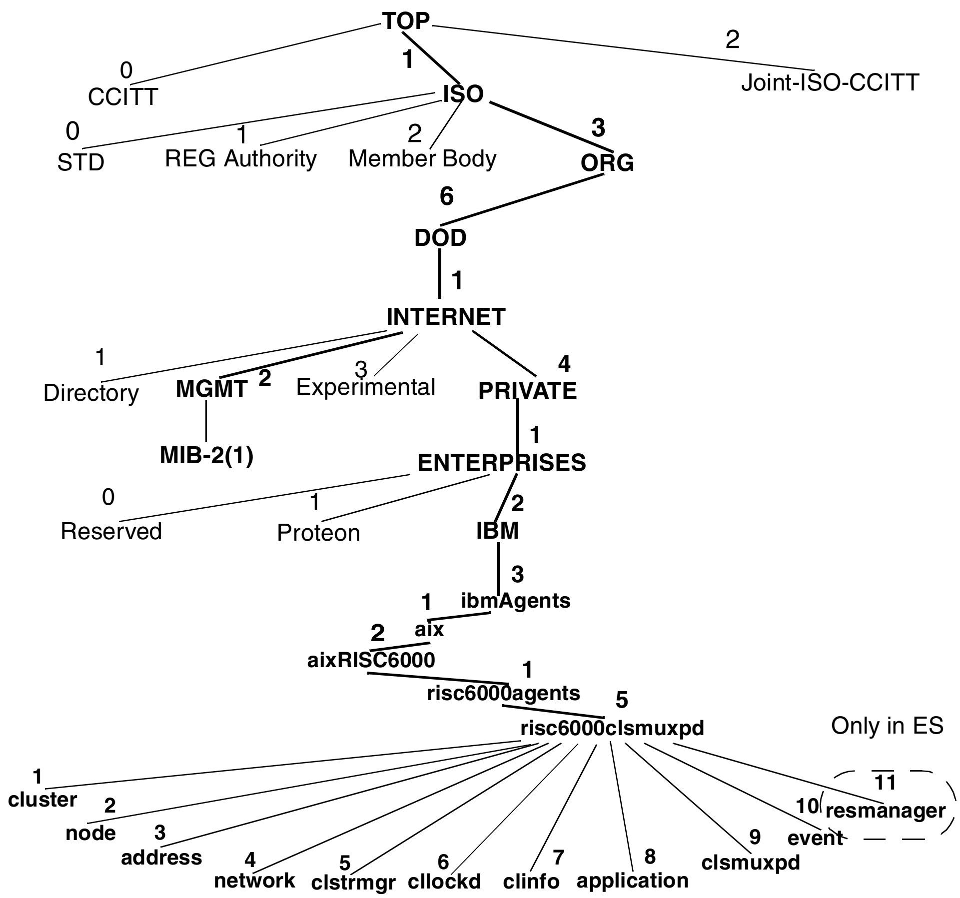

Figure 138. The HACMP MIB structure

When the cluster services are active, you can get information about the cluster using the `snmpinfo` command in addition to the cluster commands. The information obtained this way may be interpreted using the hacmp.my file.

The following is an example to retrieve cluster information using the `snmpinfo` command:

```
# snmpinfo -m dump -o /usr/sbin/cluster/hacmp.defs risc6000clsmuxpd.1
1.3.6.1.4.1.2.3.1.2.1.5.1.1.0 = 1
1.3.6.1.4.1.2.3.1.2.1.5.1.2.0 = "cluster1"
1.3.6.1.4.1.2.3.1.2.1.5.1.3.0 = ""
1.3.6.1.4.1.2.3.1.2.1.5.1.4.0 = 2
1.3.6.1.4.1.2.3.1.2.1.5.1.5.0 = 7
1.3.6.1.4.1.2.3.1.2.1.5.1.6.0 = 965313008
1.3.6.1.4.1.2.3.1.2.1.5.1.7.0 = 18000
1.3.6.1.4.1.2.3.1.2.1.5.1.8.0 = 32
1.3.6.1.4.1.2.3.1.2.1.5.1.9.0 = "arthur"
1.3.6.1.4.1.2.3.1.2.1.5.1.10.0 = "arthur"
1.3.6.1.4.1.2.3.1.2.1.5.1.11.0 = 3
1.3.6.1.4.1.2.3.1.2.1.5.1.12.0 = 7
#
```

Using hacmp.my file (you can find it on any HACMP node) you can translate the name of each MIB variable. For example, 1.3.6.1.4.1.2.3.1.2.1.5.1.8.0 (the clusterSubState) is equal to 32, which means the clusterSubState is STABLE.

For an ES cluster, there is an HACMP MIB class for the resource management, *resmanager*. The following is an example to retrieve cluster information from this class using the `snmpinfo` command. You can find a resource group name, a resource name, and a resource type, for example.

```
# snmpinfo -m dump -o /usr/sbin/cluster/hacmp.defs risc6000clsmuxpd.11
>>>>>>>>> omitted lines <<<<<<<<<<
1.3.6.1.4.1.2.3.1.2.1.5.11.1.1.2.1 = "arthurrg"         <-- resource group name
1.3.6.1.4.1.2.3.1.2.1.5.11.1.1.2.2 = "camelotrg"
>>>>>>>>> omitted lines <<<<<<<<<<
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.3.1.1 = "arthurappl"     <-- resource name
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.3.1.4 = "/afs_1"
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.3.1.11 = "arthur_svc"
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.3.1.13 = "arthurvg_1"
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.3.1.28 = "/afs_2"
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.3.1.29 = "arthurvg_2"
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.3.2.15 = "camelotappl"
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.3.2.18 = "/cfs_1"
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.3.2.19 = "/cfs_2"
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.3.2.24 = "camelot_svc"
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.3.2.26 = "camelotvg_1"
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.3.2.27 = "camelotvg_2"
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.4.1.1 = 1006            <-- resource type: application
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.4.1.4 = 1002
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.4.1.11 = 1000
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.4.1.13 = 1003
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.4.1.28 = 1002
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.4.1.29 = 1003
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.4.2.15 = 1006
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.4.2.18 = 1002
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.4.2.19 = 1002
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.4.2.24 = 1000
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.4.2.26 = 1003
1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.4.2.27 = 1003
>>>>>>>>> omitted lines <<<<<<<<<<

#
```

3.6.2 How to monitor HACMP state information?

After the HACMP state information has been collected, it can be monitored using HATivoli. HATivoli allows you to monitor cluster topology, node state, resource state, and configuration information about the cluster. In ES it is also possible to monitor actual resource group location for a given resource group.

This functionality is achieved by using the following Tivoli applications:

- Distributed Monitoring
- AEF for Extended Node Properties (see Figure 119 on page 117)
- Task Library for performing various Tivoli configurations

There are five types of *monitoring profiles* created by HATivoli in the TMR and distributed to the managed nodes (cluster nodes). These monitoring profiles cover the following information:

- Cluster state

- Cluster subState
- Node State
- Resource Group State
- Event Monitor

In the case described in Section 3.3, “HATivoli Installation and Configuration” on page 96, HATivoli has added the following monitoring profiles to the ha_cluster profile manager:

- Cluster_State_at_cluster1
- Cluster_Sub_State_at_cluster1
- Node_State_at_cluster1
- Resource_Group_arthurrg_at_cluster1
- Resource_Group_camelotrg_at_cluster1
- Event_Monitor_at_cluster1

These monitoring profiles can be seen in *Profiles* area of the window shown in Figure 117 on page 115.

Alternatively, you can use the `wls` command to list the monitoring profiles as follows:

```
# wls @ProfileManager:ha_cluster
Cluster_State_at_cluster1
Cluster_Sub_State_at_cluster1
Node_State_at_cluster1
Resource_Group_arthurrg_at_cluster1
Resource_Group_camelotrg_at_cluster1
Event_Monitor_at_cluster1
#
```

Each of these monitoring profiles contains one monitor (HATivoli script). These monitors are custom script types. They are automatically run on each subscriber node under the control of the Sentry Engine every three minutes. The Sentry Engine is a part of Distributed Monitoring application and provides a mechanism for running monitors and a timer. It reads its configuration from the node’s database and takes appropriate actions to run the monitors.

The monitors are actually querying the HACMP MIB to retrieve the information about the cluster. They return the result as a string that is transported by oserv daemon. The results are stored in the indicators on the TMR server.

This mechanism is illustrated in Figure 139 on page 147.

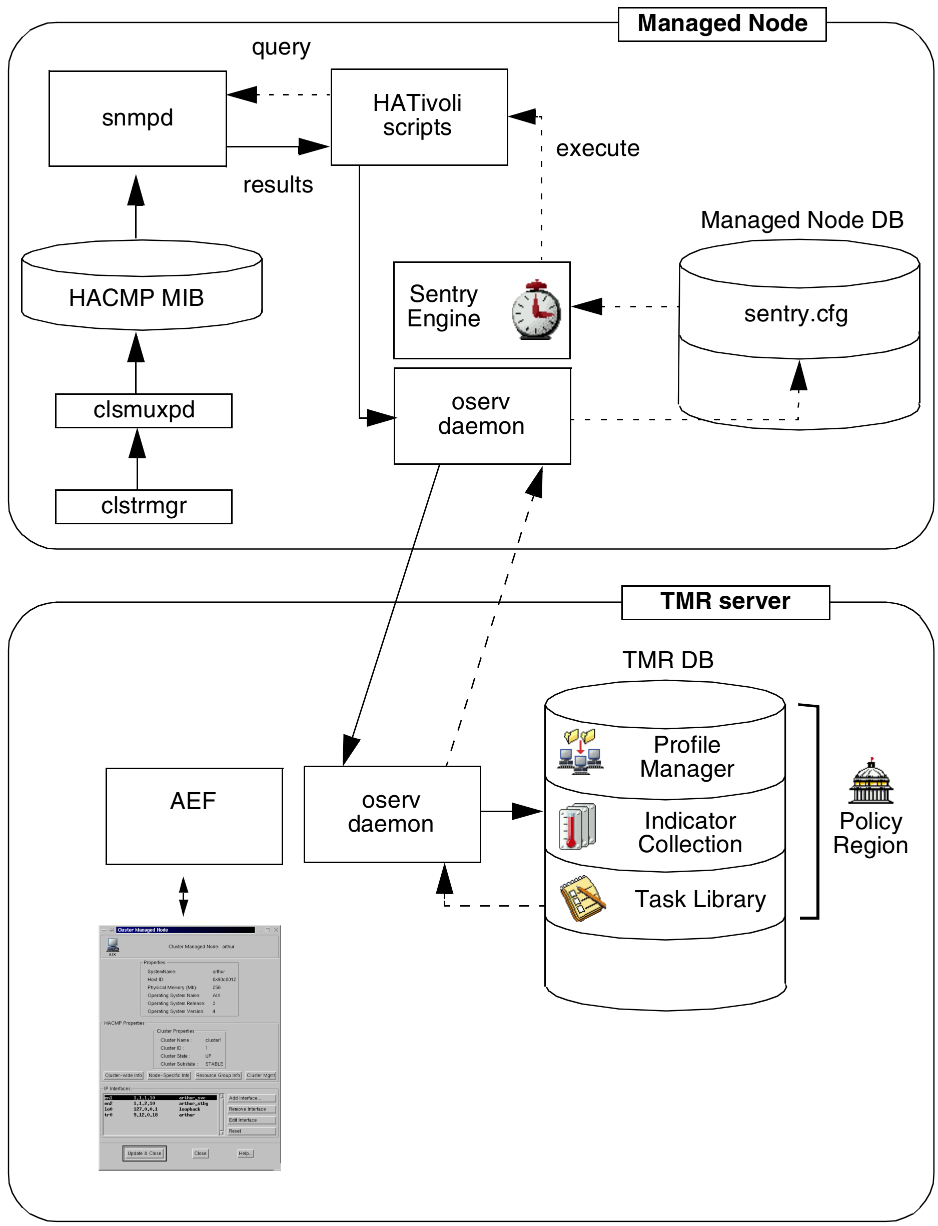

Figure 139. HATivoli monitoring process

Table 2 contains a list of the monitoring profiles and their associated values:

Table 2. Monitoring profiles and their associated values

Monitoring Profile	SNMP value	Monitor returned value
Cluster_State (1.3.6.1.4.1.2.3.1.2.1.5.1.4)	2	UP
	4	DOWN
	8	UNKNOWN
Cluster_Sub_State (1.3.6.1.4.1.2.3.1.2.1.5.1.8)	8	UNKNOWN
	16	UNSTABLE
	32	STABLE
	64	ERROR
	128	RECONFIG
Node_State (1.3.6.1.4.1.2.3.1.2.1.5.1.2.3)	2	UP
	4	DOWN
	32	JOINING
	64	LEAVING
Resource_Group_State (for each node in the cluster) (1.3.6.1.4.1.2.3.1.2.1.5.1.11.3.1.3)	2	ONLINE
	4	OFFLINE
	8	UNKNOWN
	16	AQUIRING
	32	RELEASING
	64	ERROR

The Event_Monitor is implemented slightly different from other monitoring profiles listed in Table 2.

The Event_Monitor monitoring profile is used to detect cluster configuration changes and update the Tivoli objects for cluster monitoring to reflect the new cluster configuration. Cluster configuration changes include:

- Resource group addition or deletion
- Node addition or deletion
- Resource group changes

- Cluster topology changes

This monitoring profile collects data about cluster configuration from the HACMP MIB.

Although it is associated with an indicator, it does not collect data in this container. The indicator is created only for the ease of the installation process.

Any cluster configuration changes are evaluated to determine if they affect any of the current monitors. If there is the need to delete, add, or change monitors, the Event Monitor runs a task to disable and delete the current monitors together with their corresponding indicators, then creates the new objects to reflect the actual cluster configuration and propagates the new monitors to the managed nodes.

3.6.3 How to modify monitors?

As described in previous section, each monitoring profile provided by HATivoli contains a monitor. The monitor has its own behavior defined by HATivoli, which can be modified. A monitor examines a cluster every three minutes by default. This section provides you with how to modify the interval of a monitor in the Cluster_State_at_cluster1 monitoring profile.

Double click on the Cluster_State_at_cluster1 monitoring profile icon on the ha_cluster profile manager:

This opens *Monitoring Profile Properties* window shown in Figure 140 on page 150.

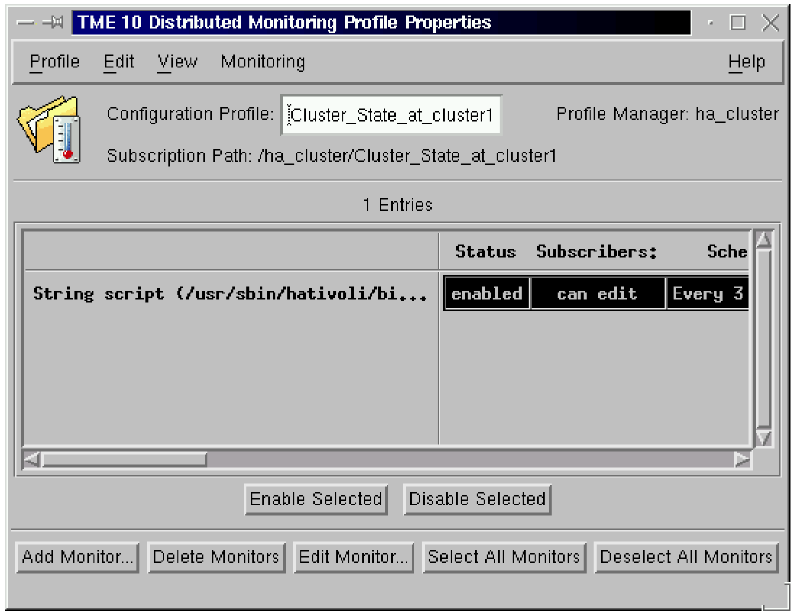

Figure 140. Monitoring profile properties window

This Monitoring Profile contains one monitor. To modify this monitor, select it then click on **Edit Monitor...** button at the bottom of this window. This opens the *Edit Monitor* window shown in Figure 141 on page 151.

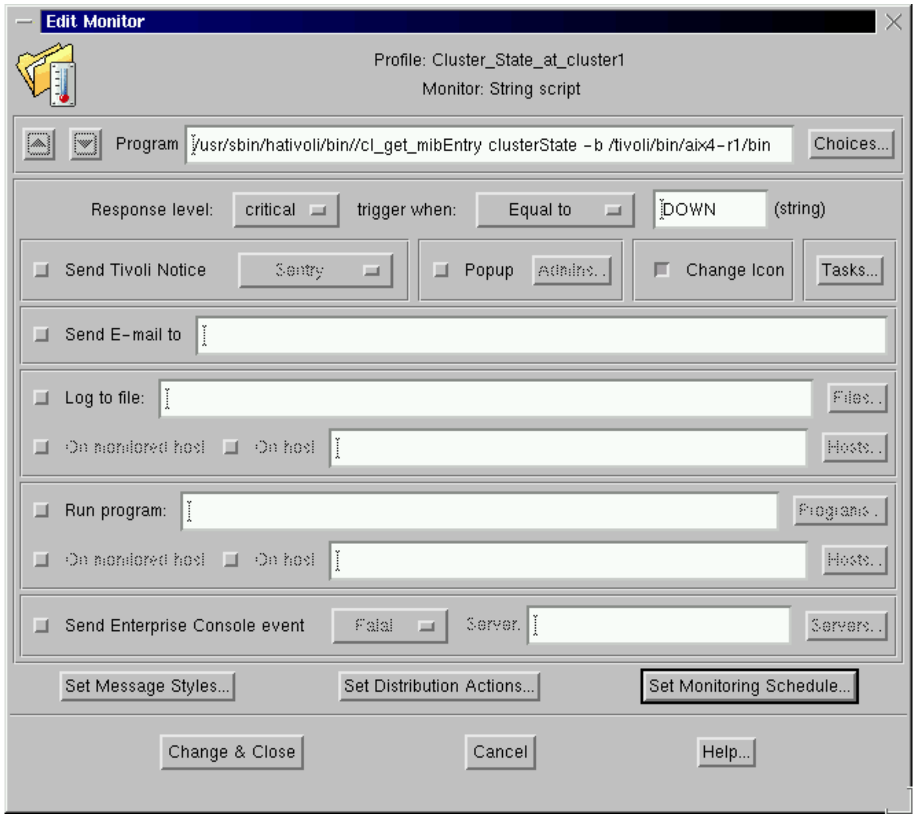

Figure 141. Edit Monitor window

The shell script which is used to setup this monitor is obtained by issuing the following CLI command:

```
riscs78 # wlsmon -sa @SentryProfile:"Cluster_State_at_cluster1"
#!/bin/sh
SentryName=${1:-'@SentryProfile:Cluster_State_at_cluster1'}
waddmon 'Universal' 'scustom' -a '/usr/sbin/hativoli/bin//cl_get_mibEntry cluste
rState -b /tivoli/bin/aix4-r1/bin' -t '3 minutes' -s 2000-08-12 7:05\
 -A D -b 8 -e 5 \
-A N -b 6 -e 7 \
-A W -b Mon -e Fri\
 -A weekend -b Sat -e Sun\
 +A Customhours -b 12 -e 11 +A Customdays -b Sun -e Sat \
-c 'critical' -R '==' 'DOWN' -i -c 'severe' -R '==' 'UNKNOWN' -i -c 'warning' -c
 'normal' -e '/usr/sbin/hativoli/bin/cl_reset' -c 'always' -i  "$SentryName"
risc78 #
```

To modify the monitoring interval, you need to modify the monitoring schedule. Click on the **Set Monitoring Schedule...** button in Edit Monitor window.

This opens the *Set Monitoring Schedule* window shown in Figure 142.

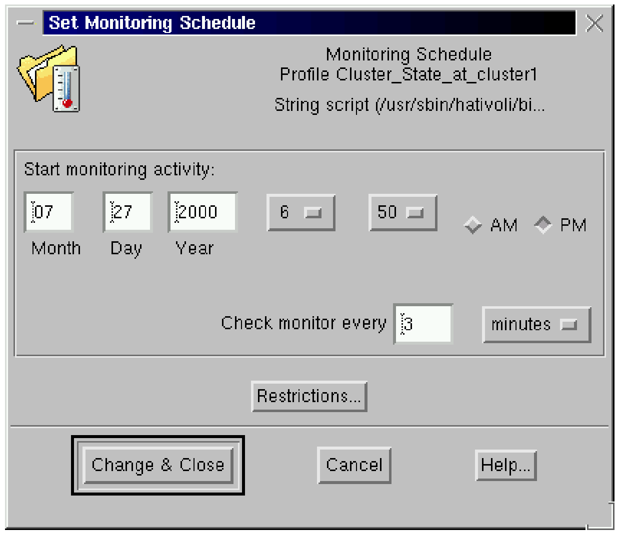

Figure 142. Set Monitoring Schedule window

As mentioned, monitor checks a cluster every three minutes by default. Change the value as you wish. Then click on the **Change & Close** button.

After you finished all the modifications for each monitor, you must save the modified monitor in the TMR database and distribute the profile to the subscribers (managed nodes). To save the modified monitor, select **Profile** > **Save** menu in the Monitoring Profile Properties window shown in Figure 143 on page 153.

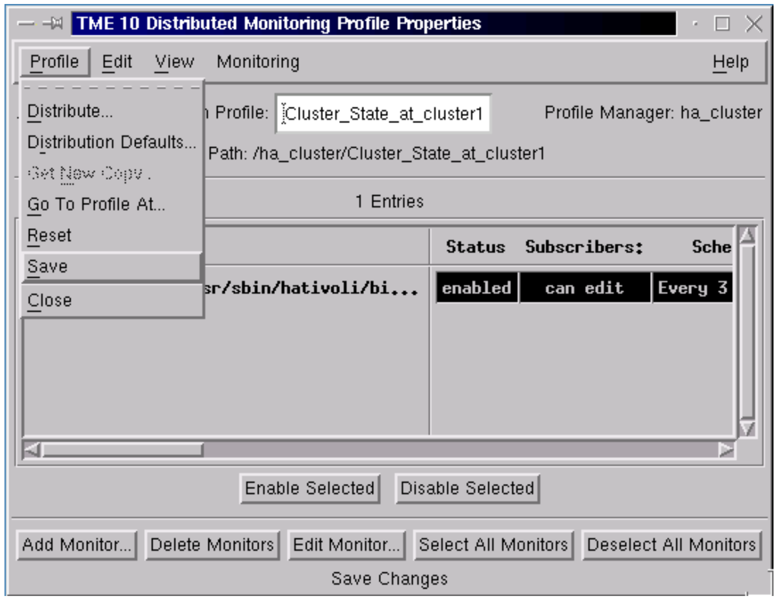

Figure 143. Saving the modified monitor

After saving the monitoring profile, select **Profile** > **Distribute** from menu as shown in Figure 144.

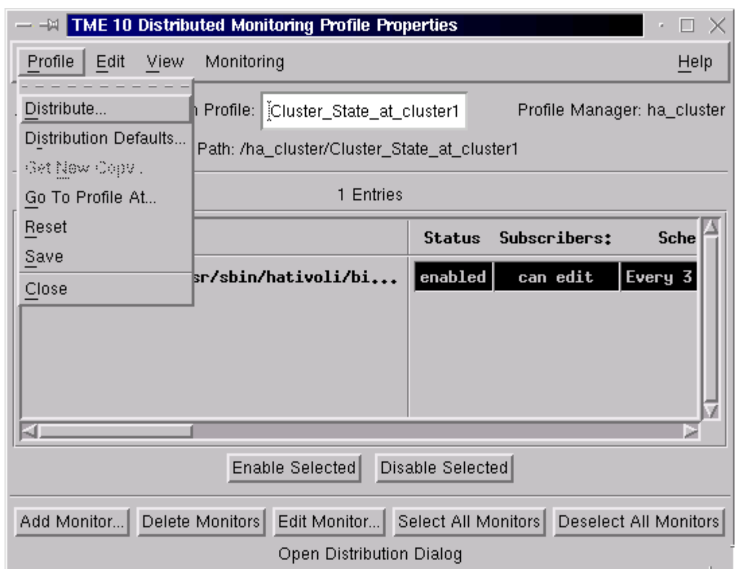

Figure 144. Distributing the modified monitor

The *Distribute Profile* window appears as shown in Figure 145 on page 154. Select the appropriate distribution actions, then click on **Distribute & Close** button. We selected “All levels of subscribers” and “Make subscribers’ profile EXACT COPY of this profile”.

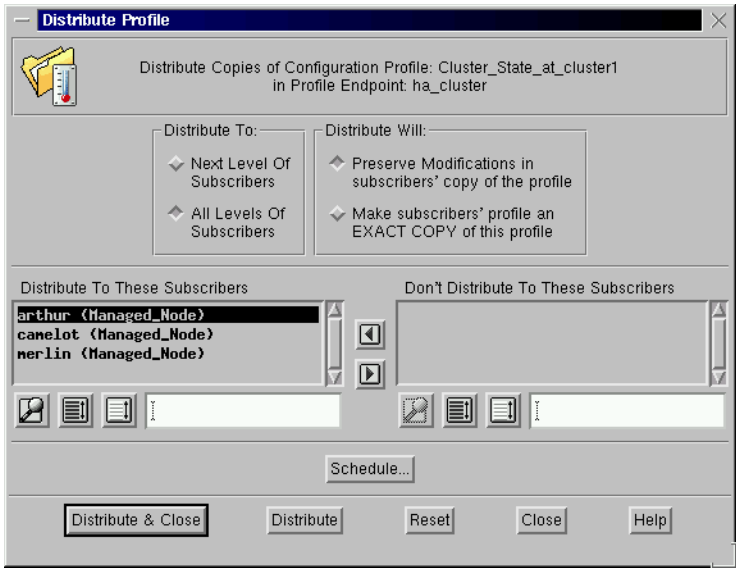

Figure 145. Distribute Profile window

Be careful when modifying individual profiles because executing the tasks in the task libraries defined for HATivoli will override the individual modifications (see Figure 131 on page 128).

Summary

Figure 146 on page 155 illustrates the relationship between the following windows:

- TME Desktop window
- Policy Region window
- Profile Manager window
- Distributed Monitoring Profile Properties window (see Figure 140 on page 150)
- Edit Monitor window (see Figure 141 on page 151)
- Set Monitoring Schedule window (see Figure 142 on page 152)

These windows are discussed in this section.

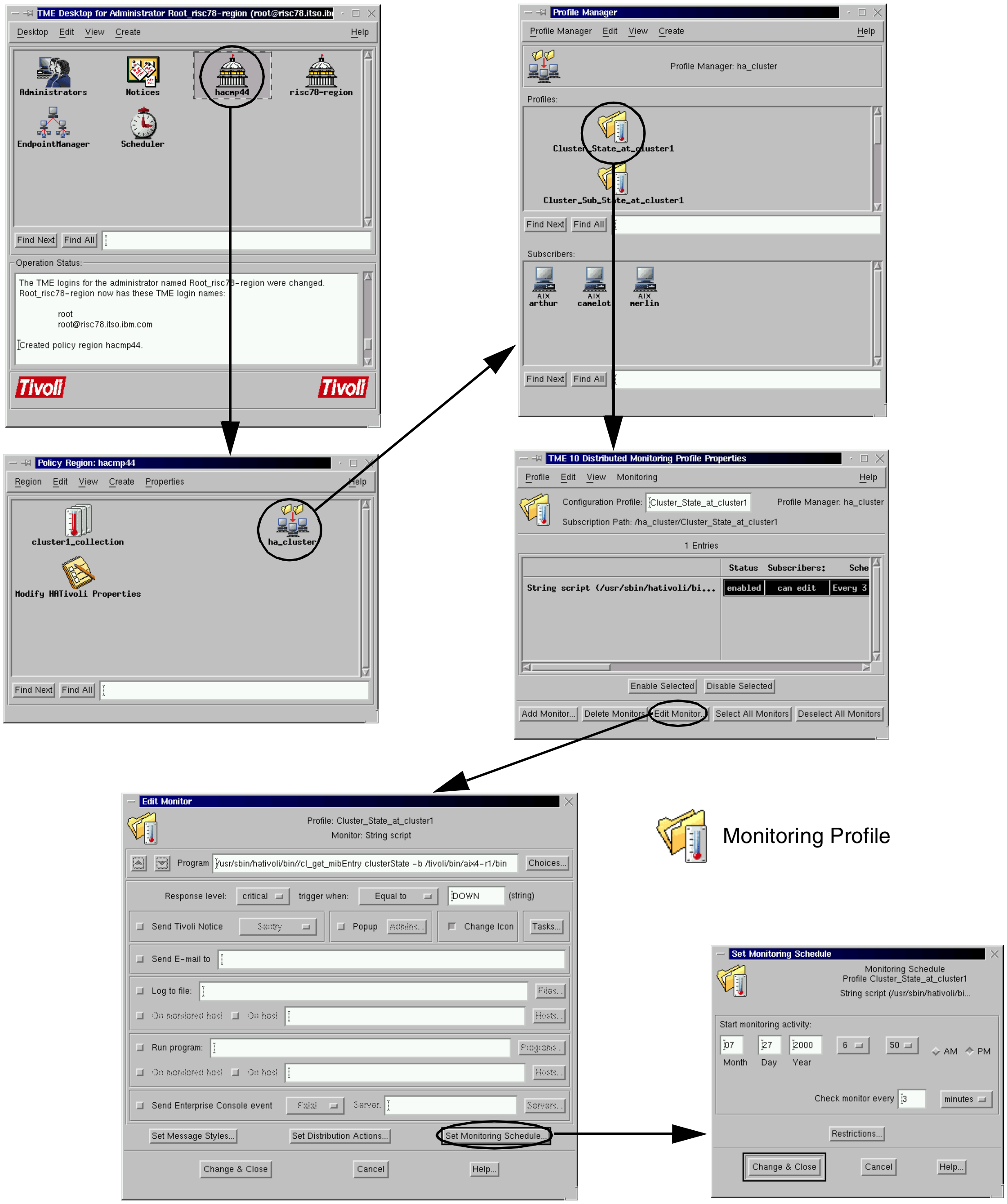

Figure 146. GUI for modifying a monitor

Chapter 4. Cascading without fallback

Cascading without fallback (CWOF) is a new resource group policy. The policy is similar to cascading, except that a CWOF resource group does not fallback to a higher priority node when it joins or reintegrates the cluster.

4.1 Defining cascading without fallback

It is important to keep in mind the difference between *fallover* and *fallback*. You will encounter these terms frequently in discussion of the various resource group policies.

Fallover refers to the movement of a resource group from the primary node on which it currently resides (owner node) to another active node (backup node) after its owner node (primary node) experiences a failure. The new owner is not a joining or reintegrating node.

Fallback refers to the movement of a resource group from its owner node (backup node) specifically to a node that is joining or reintegrating (primary node) into the cluster. A fallback occurs during a node_up event.

The figures in this section help us to understand a CWOF resource group. In our environment there are three nodes in the HACMP cluster. arthur is a primary node (priority=1) for cascading resource group arthurrg. camelot is a primary node (priority=1) for CWOF resource group camelotrg. merlin is a backup node for arthur and camelot. Its priority is two for both arthurrg and camelotrg resource groups.

If all nodes are up and the HACMP is active on all nodes, we have the situation shown in Figure 147 on page 158.

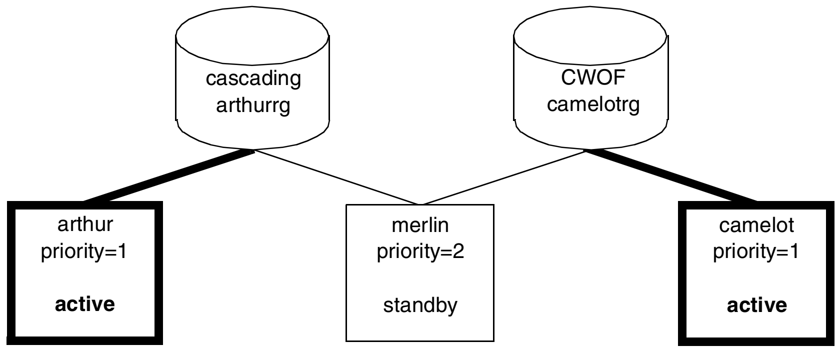

Figure 147. Initial configuration

4.1.1 Cascading resource group

When a fallover occurs, the available node with the highest priority acquires the cascading resource group. If that node is unavailable, the node with the next-highest priority acquires the resource group, and so on.

If arthur goes down, then arthurrg resource group falls over to the backup node merlin because merlin is the next-highest priority node for this resource group. The priorities on node arthur and merlin do not change. See Figure 148.

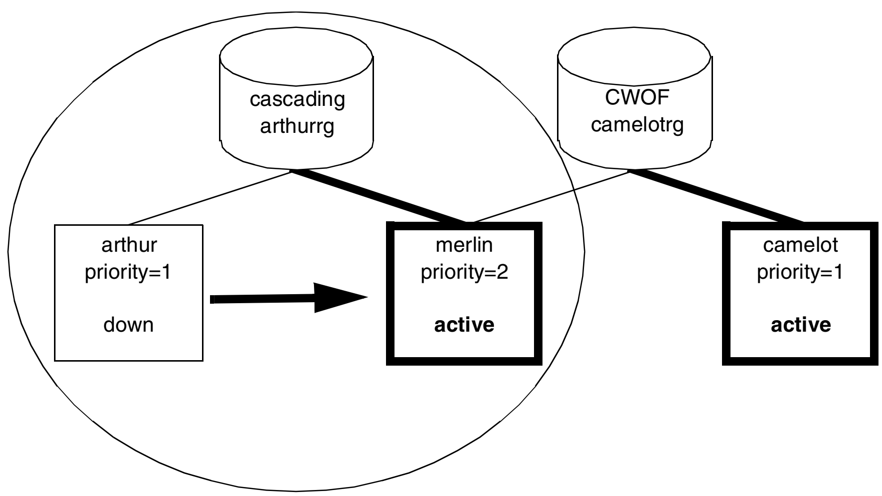

Figure 148. Fallover of a cascading resource group

When a node with a higher priority for that resource group joins or reintegrates into the cluster, it takes control of the resource group. That is, the resource group falls back from nodes with lower priorities to the higher priority node. When arthur reintegrates into the cluster, cascading resource group arthurrg falls back to primary node arthur. See Figure 149.

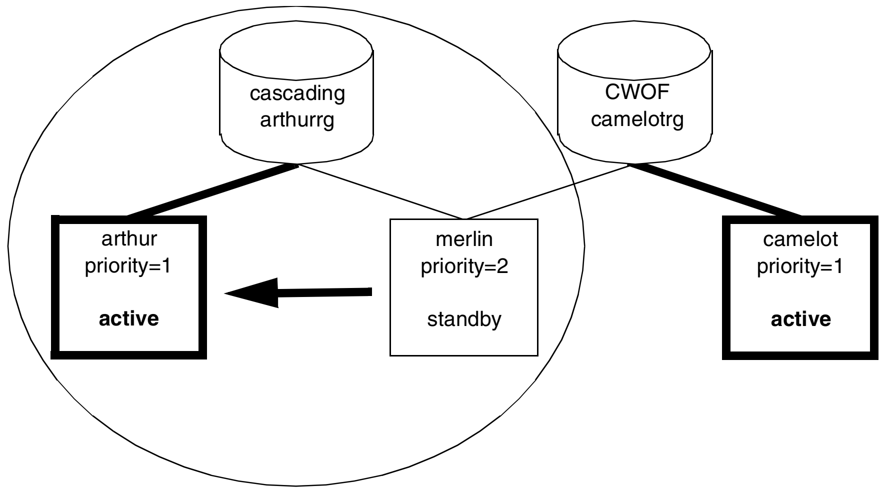

Figure 149. A cascading resource group falls back to a primary node

4.1.2 CWOF resource group

Now take a similar scenario when the node camelot with a CWOF resource group goes down. First step is similar to a cascading resource group. camelotrg CWOF resource group falls over to a backup node merlin. See Figure 150 on page 160.

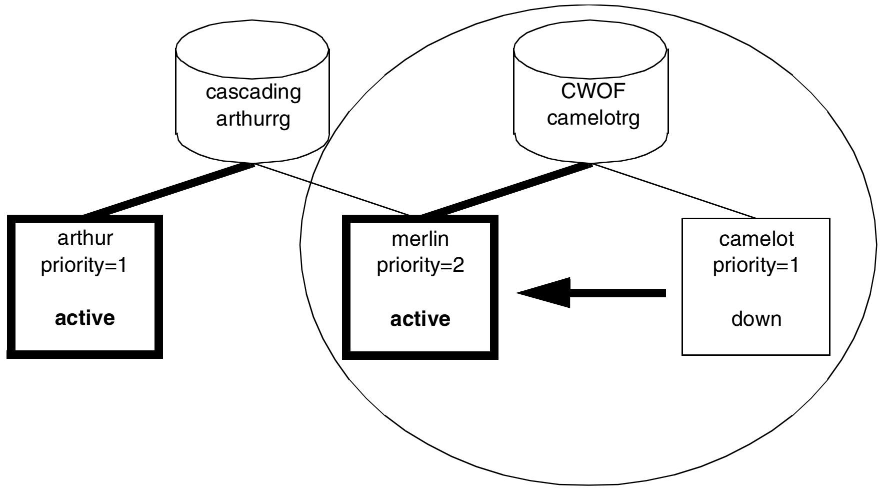

Figure 150. Fallover of CWOF resource group

When node camelot reintegrates into the cluster, CWOF resource group camelotrg does not fall back to the primary node camelot. It is still on backup node merlin and merlin has still priority two for this resource group. Note that when a node with a higher priority for a CWOF resource group joins or reintegrates into the cluster, a CWOF resource group will not fallback to the primary node. See Figure 151.

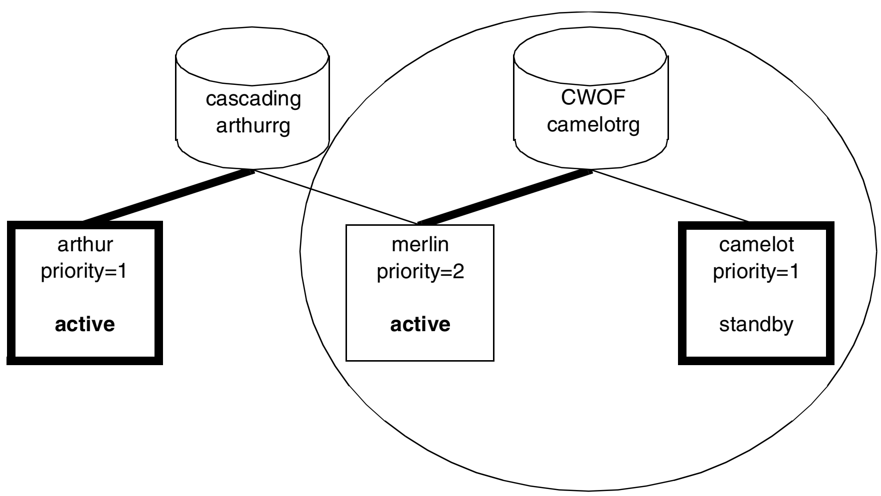

Figure 151. A CWOF resource group does not fallback

If you want to migrate a CWOF resource group back to the primary node, you must initiate this operation manually by using cldare. See Section 4.6.5, “Resource group is down while primary node is up” on page 176 for more information.

4.2 Reasons to use a CWOF resource group

The fallback causes a service outage, but CWOF allows the system manager to schedule the outage rather than have it take place at reintegration time.

CWOF provides the flexibility of a cascading group along with the higher availability of a rotating resource group. See Table 3.

Table 3. Comparison of CWOF, cascading, and rotating resource groups

	CWOF	Cascading	Rotating
Can be configured without a service address	Yes	Yes	No
Can use a standby adapter on another node	Yes	Yes	No
Has a primary node	Yes	Yes	No
Never interrupts service when a node joins	Yes	No	Yes

We recommend that you use a cascading resource group (not CWOF) when you have a strong preference for which cluster node you want to control a resource group from. For example, you may want the cluster node with the highest processing capabilities to control the resource group. So after fallover of a resource group, you want to fallback it as soon as possible.

4.3 Limitations of CWOF in HAS

Whenever an event occurs that requires fallover of a resource group, each node needs to compute whether it should take the resource group. Hence, each node needs information about resource locations.

In ES, the requisite information is provided by RSCT. In HAS, the information is provided by the global ODM. Since the global ODM does not provide as much information as, and is slower to access than RSCT, there are restrictions on a CWOF resource group in HAS:

- CWOF resource group is limited to a maximum of two nodes in HAS. Therefore you can have only a primary and one backup node.
- Either a /.rhosts file must be present on both nodes, or Kerberos enabled, for CWOF to function in HAS.

In ES, on the other hand:

- There is no special limit on the number of nodes.
- There is also no need for kerberos or /.rhosts file.

Keep it in mind that the CWOF resource group does not fallback automatically. In environments with many cluster nodes and resource groups, you might have a situation in which one node hosts too many resource groups after fallover. This may cause performance problems after node failure and its reintegration until you manually redistribute resource groups.

4.4 Differences between CWOF and other resource group policies

In this section we explain how a CWOF resource group differs from a rotating resource group or a DARE sticky move of a resource group.

4.4.1 CWOF differs from a rotating resource group

Rotating resource groups share some similarities with CWOF resource groups. However, there are important differences. Unlike cascading groups, rotating groups interact with each other. Because rotating resource groups require the use of IP address takeover, all nodes in the resource chain must share the same network connection to the resource group. If several rotating groups share a network, only one of these resource groups can be up on a given node at any time. Thus, rotating resource groups distribute themselves. Note that a rotating resource group must have a service address configured, and cannot use a standby adapter on a takeover node.

CWOF resource groups, however, may “clump” together with multiple CWOF groups on the same node. CWOF does not require an IP address to be associated with the group.

4.4.2 CWOF differs from a DARE sticky move

A DARE sticky move makes the node to which the resource group is moved the highest priority node for that group. Note that CWOF does not change the node priority.

DARE migration is enhanced in a CWOF resource group. A cascading resource group supports a DARE migration with the sticky option only. This means that the node to which this resource group falls over becomes the highest priority node until another DARE migration changes this (until a DARE to another node, DARE to stop, or a DARE to default). A CWOF resource group supports both sticky and non-sticky DARE migrations.

Note

You may find it is helpful to perform a non-sticky DARE migration when doing maintenance on a CWOF owner node. Should the default node fail, the CWOF resource group will return to the owner node if it is available.

You can find out more about DARE sticky and non-sticky move in Chapter 7, “Changing Resources and Resource Groups” in *HACMP V4.3 AIX: Administration Guide*.

4.5 Configuring a CWOF resource group

We recommend that you configure a CWOF resource group using the SMIT. To use SMIT, type the `smit cm_cfg_res.select` fastpath, then select cascading resource group. In the SMIT menu shown in Figure 152, you see the new *Cascading Without Fallback Enabled* field. Set this field to determine the fallback behavior of a cascading resource group.

```
         Change/Show Resources/Attributes for a Resource Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                     [Entry Fields]
  Resource Group Name                                 My_Resource_Group
  Node Relationship                                   cascading
  Participating Node Names                            node_1 node_2

  Service IP label                                   [node_1_svc]         +
  HTY Service Label                                  []
  Filesystems                                        [/fs_1 /fs_2]        +
>>>>>>>> omitted lines <<<<<<<<

  Inactive Takeover Activated                         false               +
  Cascading Without Fallback Enabled                  true                +
  9333 Disk Fencing Activated                         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 152. Configuring a CWOF resource group

If the CWOF field is set to *false*, a cascading resource group falls back to any higher priority node when such a node joins or reintegrates into the cluster, causing an interruption in service. In this case you define a cascading resource group.

When the CWOF field is set to *true*, a cascading resource group will not fallback to any node that joins or reintegrates into the cluster. It migrates from its owner node only if the owner node fails. It will not fall back to the owner node when it reintegrates into the cluster. If you set the CWOF field to *true*, you define a CWOF resource group. The default value for the CWOF field is false.

Note

Do not forget that after you set or change the CWOF field, you need to synchronize the cluster resources.

4.6 Examples

This section describes the steps involved in the scenario described in Section 4.1, "Defining cascading without fallback" on page 157.

4.6.1 Preparations

Before we show the scenario, we need to configure the resource groups arthurrg and camelotrg.

Configuring a cascading resource group, arthurrg

Participating nodes in arthurrg resource group are nodes arthur and merlin. We add a cascading resource group arthurrg by using the `smit cm_add_grp` fastpath. Then we type the resource group name and participating node names as shown in Figure 153 on page 165.

```
                          Add a Resource Group

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

                                                       [Entry Fields]
* Resource Group Name                                 [arthurrg]
* Node Relationship                                    cascading         +
* Participating Node Names                            [arthur merlin]    +




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

Figure 153. Add a resource group

When you hit enter, you will see the SMIT menu shown in Figure 154. There is a new field, Cascading Without Fallback Enabled. We leave this field with default value *false*. This means that we set arthurrg as a cascading resource group.

```
           Change/Show Resources/Attributes for a Resource Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                      [Entry Fields]
  Resource Group Name                                  arthurrg
  Node Relationship                                    cascading
  Participating Node Names                             arthur merlin

  Service IP label                                    [arthur_svc]           +
  HTY Service Label                                   []
  Filesystems                                         [/afs_1 /afs_2]         +
>>>>>>>> omitted lines <<<<<<<<

  Inactive Takeover Activated                          false                  +
  Cascading Without Fallback Enabled                   false                  +
  9333 Disk Fencing Activated                          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 154. Change Attributes for a Resource Group

Configuring a CWOF resource group, camelotrg

Using the same procedure we add a CWOF resource group camelotrg with participating nodes camelot and merlin. See Figure 155.

```
                           Add a Resource Group

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

                                                        [Entry Fields]
* Resource Group Name                                [camelotrg]
* Node Relationship                                   cascading        +
* Participating Node Names                           [camelot merlin]   +







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

Figure 155. Add a CWOF resource group

The difference is that we set the CWOF field to *true, as seen in* Figure 156.

```
         Change/Show Resources/Attributes for a Resource Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                     [Entry Fields]
  Resource Group Name                                 camelotrg
  Node Relationship                                   cascading
  Participating Node Names                            camelot merlin

  Service IP label                                   [camelot_svc]          +
  HTY Service Label                                  []
  Filesystems                                        [/cfs_1 /cfs_2]        +
>>>>>>>> omitted lines <<<<<<<<

  Inactive Takeover Activated                         false                 +
  Cascading Without Fallback Enabled                  true                  +
  9333 Disk Fencing Activated                         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 156. Set the CWOF flag to true

Do not forget to synchronize the cluster resources. See Figure 157 on page 167.

```
                 Synchronize Cluster Resources

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

[TOP]                                                   [Entry Fields]
  Ignore Cluster Verification Errors?                [No]                +
  Un/Configure Cluster Resources?                    [Yes]               +
* Emulate or Actual?                                 [Actual]            +
* Skip Cluster Verification                          [No]                +

  Note:
  Only the local node's default configuration files keep the changes you
  make for resource DARE emulation. Once you run your emulation, to restore
  the original configuration rather than running an actual DARE, run the
  SMIT command, "Restore System Default Configuration from Active
  Configuration." We recommend that you make a snapshot before.

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

Figure 157. Synchronize Cluster Resources

When a synchronization process is finished we have two cascading resource groups. arthurrg resource group is a cascading resource group with primary node arthur and backup node merlin, while the camelotrg resource group is a CWOF resource group with primary node camelot and backup node merlin.

4.6.2 Utilities

This section uses two utilities for monitoring cluster status (`clstat`) and cluster resources (`clfindres`).

clstat

When all nodes are up and the HACMP is running on all nodes, the `/usr/sbin/cluster/clstat` command shows the cluster status as shown in Figure 158 on page 168. It shows that the cluster is up and stable, and all nodes are up.

```
                 clstat - HACMP Cluster Status Monitor
              ---------------------------------------------

Cluster: cluster1       (1)             Sat Jul 22 10:39:33 EDT 2000
              State: UP                 Nodes: 3
              SubState: STABLE
        Node: arthur            State: UP
           Interface: arthur_svc (0)           Address: 1.1.1.10
                                               State:   UP
           Interface: arthur1_tmssa (1)        Address: 0.0.0.0
                                               State:   UP
           Interface: arthur3_tmssa (3)        Address: 0.0.0.0
                                               State:   UP

        Node: camelot           State: UP
           Interface: camelot_svc (0)          Address: 1.1.1.20
                                               State:   UP
           Interface: camelot2_tmssa (1)       Address: 0.0.0.0
                                               State:   UP
           Interface: camelot3_tmssa (2)       Address: 0.0.0.0
                                               State:   UP

        Node: merlin            State: UP
           Interface: merlin_svc (0)           Address: 1.1.1.30
                                               State:   UP
           Interface: merlin1_tmssa (2)        Address: 0.0.0.0
                                               State:   UP
           Interface: merlin2_tmssa (3)        Address: 0.0.0.0
                                               State:   UP

******************* f/forward, b/back, r/refresh, q/quit *****************
```

Figure 158. Output from the clstat command

clfindres

For the status of resource groups we use the `/usr/sbin/cluster/utilities/clfindres` command as shown in Figure 159.

```
merlin# /usr/sbin/cluster/utilities/clfindres
GroupName     Type       State    Location   Sticky Loc
---------   ----------   ------   --------   ----------
arthurrg    cascading      UP     arthur
camelotrg   cascading      UP     camelot

merlin#
```

Figure 159. Output from the clfindres command

arthurrg resource group is up on its primary node arthur. camelotrg resource group is up on its primary node camelot. This output indicates our initial configuration shown in Figure 147 on page 158.

4.6.3 Fallover and fallback of a cascading resource group

This example simulates node arthur going down and then reintegrating into the cluster. This is illustrated in Figure 148 on page 158 and Figure 149 on page 159.

First, we simulate failure on a node arthur with the takeover shutdown mode as follows.

```
                       Stop Cluster Services

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

                                                     [Entry Fields]
* Stop now, on system restart or both              now                    +

  BROADCAST cluster shutdown?                      true                   +
* Shutdown mode                                    takeover               +
          (graceful or graceful with takeover)




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

Check the cluster status using the `clstat` command as follows:

```
                  clstat - HACMP Cluster Status Monitor
               ---------------------------------------------

Cluster: cluster1       (1)             Sat Jul 22 10:39:33 EDT 2000
                State: UP               Nodes: 3
                SubState: STABLE
        Node: arthur            State: DOWN
           Interface: arthur_boot (0)          Address: 1.1.1.11
                                               State:   DOWN
           Interface: arthur1_tmssa (1)        Address: 0.0.0.0
                                               State:   DOWN
           Interface: arthur3_tmssa (3)        Address: 0.0.0.0
                                               State:   DOWN

        Node: camelot           State: UP
           Interface: camelot_svc (0)          Address: 1.1.1.20
                                               State:   UP
           Interface: camelot2_tmssa (1)       Address: 0.0.0.0
                                               State:   DOWN
           Interface: camelot3_tmssa (2)       Address: 0.0.0.0
                                               State:   UP

        Node: merlin            State: UP
           Interface: merlin_svc (0)           Address: 1.1.1.30
                                               State:   UP
           Interface: merlin1_tmssa (2)        Address: 0.0.0.0
                                               State:   UP
           Interface: merlin2_tmssa (3)        Address: 0.0.0.0
                                               State:   DOWN

******************* f/forward, b/back, r/refresh, q/quit *****************
```

We can see that node arthur is down. Check the status of resource groups using the `clfindres` command as follows.

```
merlin# /usr/sbin/cluster/utilities/clfindres
GroupName     Type       State    Location   Sticky Loc
---------   ----------   ------   --------   ----------
arthurrg    cascading      UP     merlin
camelotrg   cascading      UP     camelot

merlin#
```

We can see that the resource group arthurrg is running on node merlin

We start the HACMP on node arthur with the `smit clstart` fastpath. Cascading resource group arthurrg falls back to its primary node arthur as follows.

```
arthur# /usr/sbin/cluster/utilities/clfindres
GroupName     Type       State    Location   Sticky Loc
---------   ----------   ------   --------   ----------
arthurrg    cascading      UP     arthur
camelotrg   cascading      UP     camelot

arthur#
```

4.6.4 Fallover and fallback of a CWOF resource group

This section simulates node camelot going down and then reintegrating into the cluster. This is illustrated in Figure 150 on page 160 and Figure 151 on page 160.

First we simulate failure on a node camelot with the takeover shutdown mode. Use the `clstat` command to check that node camelot is down:

```
                  clstat - HACMP Cluster Status Monitor
               ---------------------------------------------

Cluster: cluster1       (1)             Sat Jul 22 10:39:33 EDT 2000
                State: UP               Nodes: 3
                SubState: STABLE
        Node: arthur            State: UP
           Interface: arthur_svc (0)            Address: 1.1.1.10
                                                State:   UP
           Interface: arthur1_tmssa (1)         Address: 0.0.0.0
                                                State:   DOWN
           Interface: arthur3_tmssa (3)         Address: 0.0.0.0
                                                State:   UP

        Node: camelot           State: DOWN
           Interface: camelot_boot (0)          Address: 1.1.1.21
                                                State:   DOWN
           Interface: camelot2_tmssa (1)        Address: 0.0.0.0
                                                State:   DOWN
           Interface: camelot3_tmssa (2)        Address: 0.0.0.0
                                                State:   DOWN

        Node: merlin            State: UP
           Interface: merlin_svc (0)            Address: 1.1.1.30
                                                State:   UP
           Interface: merlin1_tmssa (2)         Address: 0.0.0.0
                                                State:   DOWN
           Interface: merlin2_tmssa (3)         Address: 0.0.0.0
                                                State:   UP

******************* f/forward, b/back, r/refresh, q/quit *****************
```

After takeover of node camelot, the CWOF resource group camelotrg falls over to backup node merlin. We check this with the `clfindres` command as follows:

```
merlin# /usr/sbin/cluster/utilities/clfindres
GroupName     Type       State    Location   Sticky Loc
---------   ----------   ------   --------   ----------
arthurrg    cascading      UP     arthur
camelotrg   cascading      UP     merlin

merlin#
```

We start the HACMP on node camelot. After some seconds node camelot is up, but still on boot address. The output of the `clstat` command follows:

```
                    clstat - HACMP Cluster Status Monitor
                 ---------------------------------------------

Cluster: cluster1       (1)             Sat Jul 22 10:39:33 EDT 2000
                State: UP               Nodes: 3
                SubState: STABLE
        Node: arthur            State: UP
           Interface: arthur_svc (0)          Address: 1.1.1.10
                                               State:   UP
           Interface: arthur1_tmssa (1)        Address: 0.0.0.0
                                               State:   UP
           Interface: arthur3_tmssa (3)        Address: 0.0.0.0
                                               State:   UP

        Node: camelot           State: UP
           Interface: camelot_boot (0)         Address: 1.1.1.21
                                               State:   UP
           Interface: camelot2_tmssa (1)       Address: 0.0.0.0
                                               State:   UP
           Interface: camelot3_tmssa (2)       Address: 0.0.0.0
                                               State:   UP

        Node: merlin            State: UP
           Interface: merlin_svc (0)           Address: 1.1.1.30
                                               State:   UP
           Interface: merlin1_tmssa (2)        Address: 0.0.0.0
                                               State:   UP
           Interface: merlin2_tmssa (3)        Address: 0.0.0.0
                                               State:   UP

******************* f/forward, b/back, r/refresh, q/quit *****************
```

In this case a CWOF camelotrg resource group does not fall back to its primary node camelot. It is still running on backup node merlin.

```
camelot# /usr/sbin/cluster/utilities/clfindres
GroupName     Type       State    Location   Sticky Loc
---------   ----------   ------   --------   ----------
arthurrg    cascading      UP     arthur
camelotrg   cascading      UP     merlin

camelot#
```

If you want a CWOF resource group to fall back to its primary node, initiate the operation manually on any cluster node with an active cluster manager process. On node camelot we enter the `smit cl_resgrp_start.select` fastpath, then select the camelotrg resource group:

```
              Cluster Resource Group Management

Move cursor to desired item and press Enter.

  Bring a Resource Group Online
  Bring a Resource Group Offline
  Move a Resource Group





    _________________________________________________________________
   |                        Select a Resource Group                  |
   |                                                                 |
   | Move cursor to desired item and press Enter.                    |
   |                                                                 |
   |   arthurrg                                                      |
   |   camelotrg                                                     |
   |                                                                 |
   | F1=Help                 F2=Refresh              F3=Cancel       |
   | F8=Image                F10=Exit                Enter=Do        |
F1 | /=Find                  n=Find Next                             |
F9 |_________________________________________________________________|
```

When we press enter, we see the SMIT menu:

```
                  Bring a Resource Group Online

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

                                                        [Entry Fields]
  Resource Group to Bring Online                      camelotrg
  Emulate or Actual?                                  Actual              +
  Perform Cluster Verification First?                 No                  +
  Ignore Cluster Verification Errors?                 No                  +





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

We set the value in the field "Perform Cluster Verification First?" to No because we did not change anything in the cluster configuration from the last cluster synchronization. Verification takes time, and in this case we want to skip it.

When we press enter, the SMIT menu shows the following during the migration process:

```
                              COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

Executing cldare command: cldare -M camelotrg:default  -v

Performing preliminary check of migration request...

Migration request passed preliminary check for compatibility with
current cluster configuration and state.

Verifying additional pre-requisites for Dynamic Reconfiguration...
...completed.

clsnapshot: Creating file /usr/es/sbin/cluster/snapshots/active.0.odm...

clsnapshot: Succeeded creating Cluster Snapshot: active.0.

cldare: Requesting a refresh of the Cluster Manager...
0513-095 The request for subsystem refresh was completed successfully.
...completed.

Waiting for migrations to occur...........completed.

Performing final check of resource group locations:

GroupName    Type       State    Location   Sticky Loc
---------   ----------   ------   --------   ----------
camelotrg   cascading     UP     camelot
-------------------------------------------------------

Requested migrations succeeded.


F1=Help            F2=Refresh          F3=Cancel          F6=Command
F8=Image           F9=Shell            F10=Exit           /=Find
n=Find Next
```

Note that SMIT uses the `cldare` command with the option `default`. The `-v` flag skips the verification. You can also enter this command from the command line on any cluster node for particular resource group and with an active cluster manager process as follows:

```
# /usr/sbin/cluster/utilities/cldare -M camelotrg:default -v
```

The `clfindres` command shows that camelotrg resource group is back up on primary node camelot:

```
camelot# /usr/sbin/cluster/utilities/clfindres
GroupName     Type       State    Location   Sticky Loc
---------   ----------   ------   --------   ----------
arthurrg    cascading      UP     arthur
camelotrg   cascading      UP     camelot

camelot#
```

4.6.5 Resource group is down while primary node is up

You may encounter a situation in which the primary node is up but a resource group remains down. Because no subsequent node that comes up will acquire the resource group, the resource group will remain in an inactive state. This situation happens in the following cases:

- If a CWOF resource group is placed on a non-primary node, and that node is brought down by either a graceful shutdown or the `cldare stop` command.
- In the fallover option of Application Monitoring, if a rg_move event moves a resource group from its primary node to a lower priority node, and you bring the lower priority node down by a graceful shutdown or the `cldare stop` command.
- If you use the `cldare stop` command to bring down a cascading resource group that is assigned an inactive takeover value of false, and which resides on the primary node. After DARE stopping resource group, you should not assume that a joining or reintegrating node will bring that resource group online.

To demonstrate the first case, we simulated failure on node camelot with the takeover shutdown mode. After a few seconds the camelotrg resource group falls over to a node merlin:

```
merlin# /usr/sbin/cluster/utilities/clfindres
GroupName     Type       State    Location   Sticky Loc
---------   ----------   ------   --------   ----------
arthurrg    cascading      UP     arthur
camelotrg   cascading      UP     merlin

merlin#
```

Then we start the HACMP on node camelot, and when it reintegrates into the cluster, we simulate failure of node merlin with a graceful shutdown. Output from the `clstat` command shows that node camelot is up on its boot address after reintegration, and node merlin is down after graceful shutdown:

```
                  clstat - HACMP Cluster Status Monitor
               ---------------------------------------------

Cluster: cluster1       (1)             Sat Jul 22 10:39:33 EDT 2000
                State: UP               Nodes: 3
                SubState: STABLE
        Node: arthur            State: UP
           Interface: arthur_svc (0)          Address: 1.1.1.10
                                               State:   UP
           Interface: arthur1_tmssa (1)        Address: 0.0.0.0
                                               State:   UP
           Interface: arthur3_tmssa (3)        Address: 0.0.0.0
                                               State:   DOWN

        Node: camelot           State: UP
           Interface: camelot_boot (0)         Address: 1.1.1.21
                                               State:   UP
           Interface: camelot2_tmssa (1)       Address: 0.0.0.0
                                               State:   UP
           Interface: camelot3_tmssa (2)       Address: 0.0.0.0
                                               State:   DOWN

        Node: merlin            State: DOWN
           Interface: merlin_svc (0)           Address: 1.1.1.30
                                               State:   DOWN
           Interface: merlin1_tmssa (2)        Address: 0.0.0.0
                                               State:   DOWN
           Interface: merlin2_tmssa (3)        Address: 0.0.0.0
                                               State:   DOWN

******************* f/forward, b/back, r/refresh, q/quit *****************
```

Note that camelotrg resource group is not available on any node in spite of the fact that primary node camelot is up.

```
camelot# clfindres
GroupName     Type       State    Location   Sticky Loc
---------   ----------   ------   --------   ----------
arthurrg    cascading      UP     arthur
camelotrg   cascading     DOWN    N/A

camelot#
```

If you want to bring up camelotrg resource group, you have to bring up a CWOF resource group manually.

You can start a resource group from the `smit cl_resgrp_staart.select` fastpath. Alternatively, you can use the `cldare` command with the `default` flag as follows:

```
# cldare -M <resource_group_name>:default [:sticky]
```

You need not specify a target node. The resource group is activated on the node that has been designated as the primary node.

Note

The `cldare` command must be run from a node with an active cluster manager process in order for the dynamic reconfiguration to proceed.

Note that you can use the `default` flag when you want to restore a resource group to its original state, for example to remove an earlier sticky designation.

You can also use the `sticky` flag to the `cldare` command if you want the resource group to stay on that node after a failure or reintegration of another cluster node.

More information about the `cldare` command and the `sticky` flag, refer to Chapter 7, “Changing Resources and Resource Groups“ and Appendix B in *HACMP V4.3 AIX: Administration Guide*.

Note

When you use SMIT to bring a resource group online, `sticky` is not an option. The resource group will fall over and fall back to other nodes just as the resource policy dictates.

In our example we bring up camelotrg resource group using the `cldare` command without verification on camelot node:

```
camelot# cd /usr/es/sbin/cluster/utilities
camelot# ./cldare -M camelotrg:default  -v
Performing preliminary check of migration request...

Migration request passed preliminary check for compatibility with
current cluster configuration and state.

Verifying additional pre-requisites for Dynamic Reconfiguration...
...completed.


clsnapshot: Creating file /usr/es/sbin/cluster/snapshots/active.0.odm...

clsnapshot: Succeeded creating Cluster Snapshot: active.0.

cldare: Requesting a refresh of the Cluster Manager...
0513-095 The request for subsystem refresh was completed successfully.
...completed.


Waiting for migrations to occur.......... completed.

Committing location information to ODM on all nodes...... completed.

Performing final check of resource group locations:

GroupName     Type       State    Location   Sticky Loc
---------   ----------   ------   --------   ----------
camelotrg   cascading      UP     camelot
-------------------------------------------------------

Requested migrations succeeded.

camelot#
```

As you can see in the end of the output, resource group camelotrg is back up on the primary node camelot.

Chapter 5. Cluster verification enhancements

This chapter illustrates and describes the following items that are improved or added in the verification process:

- clverify utility
- Cluster names
- Serial network configurations
- Optional verification

All features are present in both HAS and ES.

5.1 clverify utility

After defining the cluster topology and configuration, we recommend you run the cluster verification utility `clverify` on one node to check that all cluster nodes agree on the cluster configuration and the assignment of the HACMP resources. Run this utility before starting the HACMP. You should also verify the cluster verification after making changes to your hardware or software. Note that this utility is not totally new. It was just supplemented with some new checks. The `clverify` utility is under /usr/sbin/cluster/diag directory.

The `clverify software` command verifies that HACMP specific modifications to AIX system files exist and are correct. It has one flag, `lpp`.

The `clverify cluster` command has the following flags:

- The `topology` flag verifies that all nodes agree on the cluster topology. It contains the following two flags:
  - The `topology check` flag checks for agreement on cluster, node, network, and adapter information. This flag is not available from the SMIT. For example, it checks for invalid characters in cluster names, node names, network names, adapter names, and resource group names.
  - The `topology sync` flag allows you to synchronize the cluster topology if necessary, forcing agreement with the local node's definition.
- The `config` flag verifies that networks are configured correctly and that all nodes agree on the ownership of resources. It contains the following three flags:
  - The `config networks` flag checks the valid configuration of adapters and serial lines, and the netmask consistency on all cluster nodes.

- In the HACMP 4.4 it also checks each cluster node to determine whether multiple RS232 serial networks exist on the same tty device.
- In ES it limits the number of non-IP networks to two of the same type per node.
- It detects adapters in the "defined" state as opposed to "available."

- The `config resources` flag:
  - This checks agreement among all nodes on the ownership of defined resources (filesystems, volume groups, disks, application servers, and events) and on the distribution of resources in case of a takeover.
  - This checks, more specifically, the existence and defined ownership of filesystems to be taken over.
  - This checks the volume group and disks where the filesystems reside to verify that the takeover information matches the owned resources information.
  - This checks the major device numbers for NFS-exported directories.
  - This ensures that application servers are configured correctly.
  - This prints out diagnostic information about custom snapshot methods, custom verification methods, custom pre/post events, and redirection of cluster log files.
- The `config all` flag checks the network topology and resources, and runs all custom defined verification methods.

If you have configured Kerberos on your system, the `clverify` utility also verifies that:

- All IP labels listed in the configuration have the appropriate service principals in the .klogin file on each node in the cluster.
- All nodes have the proper service principals.
- Kerberos is installed on all nodes in the cluster.
- All nodes have the same security mode setting.

You can run the cluster verification through the SMIT or you can run `clverify` utility interactively or directly. Using the SMIT, enter the `smit clverify.dialog` fastpath as follows:

```
                              Verify Cluster

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

                                                          [Entry Fields]
  Base HACMP Verification Methods                        both              +
          (Cluster topology, resources, both, none)
  Custom Defined Verification Methods                    [All]             +
  Error Count                                            []                #
  Log File to store output                               []



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

For more information about using the `clverify` command interactively or about adding, changing, or removing custom defined verification methods that perform specific checks, refer to Chapter 8 “Verifying a Cluster Configuration” in *HACMP V4.3 AIX: Administration Guide*. Also see the clverify man pages for details about using this utility.

Note

The cluster synchronization and verification functions use the `rcmd` and `rsh` commands, and thus require /.rhosts file.

The `clverify` utility does not report filesystem and fast recovery inconsistencies.

5.2 Defined list of valid characters in the HACMP configurations

Valid characters in the HACMP 4.4 configurations are:

- Letters a - z and A - Z.
- Digits 0 - 9.
- The underscore <_> and the hyphenation <->[1].

Note

The first character *must* be a letter.

[1] The hyphenation “-” may appear only in adapter names.

Keep it in mind that cluster name can be an ASCII text string no longer than 31 characters. We recommend that:

- Names are unique inside the HACMP cluster.
- You use a consistent naming convention.
- Names are logical.
- Names are less than eight characters long.
- You do not use hostname for node names or cluster name.

> **Note**
>
> Users with invalid characters in their existing configuration may encounter errors when they upgrade to the HACMP 4.4

We demonstrate verification examples in the following sections.

5.2.1 Incorrect resource group name

In this example we change a resource group name to an incorrect name and then synchronize cluster resources.

We enter the `smit cm_chg_grp.select` fastpath, then select camelotrg and change the name *camelotrg* to *camelot-rg* as follows:

```
                        Change/Show a Resource Group

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

                                                        [Entry Fields]
  Resource Group Name                                  camelotrg
  New Resource Group Name                             [camelot-rg]
  Node Relationship                                    cascading         +
  Participating Node Names                            [camelot merlin]   +




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

Press enter to synchronize the cluster resources. During the synchronize process we get an error about invalid resource group name:

```
                          COMMAND STATUS

Command: failed        stdout: yes          stderr: no

Before command completion, additional instructions may appear below.

>>>>>>>> omitted lines <<<<<<<<

Retrieving Cluster Topology...
Verifying Cluster Topology...

ERROR: Invalid resource group name: 'camelot-rg'

Verifying Configured Resources...

>>>>>>>> omitted lines <<<<<<<<

F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next
```

5.2.2 Adapter name with the hyphenation

In this example we change an adapter name to the name with hyphenation <->, and then verify the cluster topology with the `clverify` command.

We enter the `smit cm_config_adapters.chg.select` fastpath, select camelot_boot, and change *camelot_boot* to *camelot-boot* as follows.

```
                        Change/Show an Adapter

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                      [Entry Fields]
* Adapter IP Label                                     camelot_boot
  New Adapter Label                                   [camelot-boot]
  Network Type                                        [ether]          +
  Network Name                                        [ether1]         +
  Network Attribute                                    public          +
  Adapter Function                                     boot            +
  Adapter Identifier                                  [1.1.1.21]
  Adapter Hardware Address                            []
  Node Name                                           [camelot]        +

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

Then we verify cluster topology with the `clverify` command. Verification completed normally without errors:

```
camelot# /usr/sbin/cluster/diag/clverify cluster topology check
Contacting node arthur...
HACMPnetwork ODM on node arthur verified

>>>>>>>> omitted lines <<<<<<<<

Verification to be performed on the following:
        Cluster Topology

Retrieving Cluster Topology...


Verifying Cluster Topology...


Remember to redo automatic error notification if configuration has changed.

Verification has completed normally.
Command completed.

      --------- Hit Return To Continue ---------
```

5.3 Valid configurations for clusters with serial networks

When you plan serial networks in your cluster, keep the following in mind:

- HACMP 4.4 detects the existence of more than one serial network on the same port and reports an error. So HACMP rejects multiple serial networks on the same TTY. For more information, see the next section.
- ES supports only two non-IP networks of the same type per node. This means:
  - You can have more than two non-IP networks per node, but only two of the same type per node. For instance, you can have two tmssa networks and one rs232 network on the same node.
  - You can have two rs232, two tmscsi, and two tmssa networks per node at most.

  For more information, see Section 5.3.2, “More than two non-IP networks per node” on page 188.

Note

If you have used more than two non-IP networks of the same type per node in HAS and wish to perform a node-by-node migration to ES 4.4, then you must first reconfigure the cluster to satisfy ES requirements.

- ES can use any serial port for heartbeat, provided these two conditions are met:
  - The hardware supports use of that serial port for modem attachment.
  - The serial port is free for the HACMPs exclusive use.

  Some native serial ports on IBM processors do *not* meet these conditions. For example:
  - All ports on the S70, S7A, and S80.
  - Serial ports one and two in the F50, H50, and H70.

    Only serial port S3 should be used for the HACMP heart beat for these models, if a service processor is attached. If a service processor is *not* attached, you can use also serial ports one and two.
  - Ports one, two, and three on the F80, H80, and M80.

    These processors come with four serial ports. Only the fourth serial port can be used for HACMP heart beat.

> **Note**
>
> Refer to the hardware product documentation for information when determining if your serial ports meet the requirements.

- HAS supports mesh configurations with serial networks in clusters of more than three nodes while this is not supported in ES. Note that a mesh configuration means connection from each node to all other nodes. ES cluster with more than three nodes connected with non-IP networks (of the same type) must be connected point-to-point. No ES node can be linked to more than two other nodes.

The following section provides examples.

5.3.1 More than one serial network on the same TTY

The first example shows what happens if you define more than one serial network on the same TTY. In our scenario we have three nodes in the cluster; arthur, camelot, and merlin. We defined the rs232 serial networks as shown in Table 4 on page 188.

Table 4. rs232 serial networks

network name	arthur	camelot	merlin
rs232_am	/dev/tty0	X	/dev/tty0
rs232_cm	X	/dev/tty0	/dev/tty1
rs232_ac	/dev/tty0	/dev/tty0	X

Note that we have defined two serial networks (rs232_am and rs232_ac) on the same tty0 on node arthur, and two serial networks (rs232_cm and rs232_ac) on the same tty0 on node camelot.

When we verify network configuration with the `clverify` command or we try to synchronize the cluster topology through the SMIT, we get errors. In our example we enter the `clverify` command as follows:

```
# clverify cluster config networks
```

The output from the command is as follows:

```
camelot# /usr/sbin/cluster/diag/clverify cluster config networks
Contacting node arthur...
HACMPnetwork ODM on node arthur verified

>>>>>>>> omitted lines <<<<<<<<

Verifying Cluster Topology...

ERROR: Node [arthur] has [2] rs232s configured on tty [/dev/tty0]
ERROR: Node [arthur] has [2] rs232s configured on tty [/dev/tty0]
ERROR: Node [camelot] has [2] rs232s configured on tty [/dev/tty0]
ERROR: Node [camelot] has [2] rs232s configured on tty [/dev/tty0]

>>>>>>>> omitted lines <<<<<<<<

Verification has completed normally.
clconfig: Error(s) have been detected.
Exit Code: 1
Command completed.

      --------- Hit Return To Continue ---------
```

5.3.2 More than two non-IP networks per node

Second example shows what happens if you have more than two non-IP networks *of the same type* per node. In our scenario we have three nodes in the cluster; arthur, camelot, and merlin. We defined the tmssa and rs232 serial networks as shown in Table 5.

Table 5. Serial networks

network name	arthur	camelot	merlin
tmssa_am	/dev/tmssa_m	X	/dev/tmssa_a
tmssa_cm	X	/dev/tmssa_m	/dev/tmssa_c
tmssa_ac	/dev/tmssa_c	/dev/tmssa_a	X
rs232_am	/dev/tty0	X	/dev/tty0
rs232_cm	X	/dev/tty0	/dev/tty1
rs232_ac	/dev/tty1	/dev/tty1	X
rs232_aacc	/dev/tty2	/dev/tty2	X

Note that nodes arthur and camelot have two tmssa serial networks and three rs232 networks. Node merlin has two tmssa and two rs232 networks.

During the network configuration check with the `clverify` command, or during the cluster topology synchronization through SMIT, we get errors. In our example we enter the `clverify` command as follows:

```
# clverify cluster config networks
```

Output from the `clverify` command is as follows:

```
camelot# /usr/sbin/cluster/diag/clverify cluster config networks
Contacting node arthur...
HACMPnetwork ODM on node arthur verified

>>>>>>>> omitted lines <<<<<<<<

Verifying Cluster Topology...

ERROR: The node arthur has 3 rs232 networks.
No more than two networks of type rs232 may be configured on a given node.
ERROR: More than 2 non-IP networks of type tty/rs232/tty on node arthur
ERROR: The node camelot has 3 rs232 networks.
No more than two networks of type rs232 may be configured on a given node.
ERROR: More than 2 non-IP networks of type tty/rs232/tty on node camelot

>>>>>>>> omitted lines <<<<<<<<

Verification has completed normally.
clconfig: Error(s) have been detected.
Exit Code: 1
Command completed.

      --------- Hit Return To Continue ---------
```

Note that there are no error messages for the node merlin. It has defined four serial networks, two tmssa and two rs232.

5.4 Optional verification

To save time during the synchronization of the cluster resources or topology, in HACMP 4.4 you can skip cluster verification. This is used when configuration is well known and simply needs to be synchronized to all nodes.

A new field Skip Cluster Verification is added in the Synchronize Cluster Resources and Synchronize Cluster Topology SMIT menu. Default value is No.

```
                      Synchronize Cluster Topology

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

[TOP]                                                     [Entry Fields]
  Ignore Cluster Verification Errors?                  [No]                    +
* Emulate or Actual?                                   [Actual]                +
* Skip Cluster Verification                            [No]                    +

  Note:
  Only the local node's default configuration files
  keep the changes you make for topology DARE
  emulation. Once you run your emulation, to
  restore the original configuration rather than
  running an actual DARE, run the SMIT command,
  "Restore System Default Configuration from Active
  Configuration."
  We recommend that you make a snapshot before
[MORE...9]

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

Skipping cluster verification during synchronization is available only when a cluster is inactive. Even if one node is active (if at least one node has an active cluster manager process), the cluster verification will be run automatically.

You can skip cluster verification also from the command line using the `cldare` command with the `-x` flag.

5.4.1 Skipping verification

To skip cveirification, use the following procedure:

- To skip verification from the command line, enter the following command:

  `# cldare -r '-x'` (resources synchronization without verification)

  `# cldare -t '-x'` (topology synchronization without verification)

- To skip the cluster resources or topology verification using the SMIT, choose *Yes* at the Skip Cluster Verification field:

```
                    Synchronize Cluster Resources

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

[TOP]                                                   [Entry Fields]
  Ignore Cluster Verification Errors?                [No]                 +
  Un/Configure Cluster Resources?                    [Yes]                +
* Emulate or Actual?                                 [Actual]             +
* Skip Cluster Verification                          [Yes]                 +

  Note:
  Only the local node's default configuration files
  keep the changes you make for resource DARE
  emulation. Once you run your emulation, to
  restore the original configuration rather than
  running an actual DARE, run the SMIT command,
  "Restore System Default Configuration from Active
  Configuration."
[MORE...3]

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

Chapter 6. Tuning parameters

HACMP 4.4 provides new performance tuning parameters for easier and more granular control.

6.1 System tuning and the dead man switch

The term “dead man switch” (DMS) represents the situation of a cluster node that has been intentionally crashed by HACMP. DMS is essentially a timer that HACMP periodically resets. There are rare conditions, however, where HACMP is unable to reset the timer and the node is halted. These conditions typically occur in case of very serious performance degradation. Examples include:

- Large I/O transfers
- Running out of memory, which causes high paging activity
- Starving for CPU time
- Excessive error logging

While it may not seem obvious, the reason for implementing a DMS is to protect the data on the external disks. The DMS halts a node when this node enters a hung state that extends beyond the predefined timer. By crashing the node, the standby node is able to acquire the resources in an orderly fashion, avoiding possible contention problems, in particular for the external, shared disks.

Since the original root cause of the DMS is a performance problem, system tuning is vital in order to reduce the possibilities of experiencing crashes. There are three areas where tuning must be performed:

- Configure I/O pacing
- Increase the syncd frequency
- Adjust the heartbeat rate

HACMP 4.4 introduces new options to perform tuning in each of these areas. The new choice “Advanced Performance Tuning Parameters” is available in the SMIT menu shown in Figure 160 on page 194. You can reach it with the `smit cm_configure_menu` fastpath.

```
              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
  Advanced Performance Tuning Parameters




F1=Help             F2=Refresh          F3=Cancel           Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

Figure 160. The new tuning SMIT menu

After selecting the option “Advanced Performance Tuning Parameters”, we reach the SMIT menu shown in Figure 161.

```
             Advanced Performance Tuning Parameters

Move cursor to desired item and press Enter.

  Change/Show I/O pacing
  Change/Show syncd frequency
  Change/Show a Network Module




F1=Help             F2=Refresh          F3=Cancel           Esc+8=Image
Esc+9=Shell         Esc+0=Exit          Enter=Do
```

Figure 161. The Advanced Performance Tuning Parameters SMIT menu

The three options of this menu are explained in the next section, Section 6.3, “Increase the syncd frequency” on page 196, and Section 6.4.1, “Tuning the network interface modules” on page 197.

6.2 Configure I/O pacing

I/O pacing is an AIX configuration parameter that permits sharing system resources more equitably between all running processes when large write operations are occurring. Very large write I/O operations can cause serious performance problems and generate crashes of the DMS.

In addition to the standard AIX SMIT menu reachable with the `smit chgsys` fastpath, now HACMP 4.4 has its own menu to configure I/O pacing:

```
               Change/Show I/O pacing

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

                                                         [Entry Fields]
  HIGH water mark for pending write I/Os per file    [33]               +#
  LOW water mark for pending write I/Os per file     [24]               +#




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
```

Figure 162. Configuring I/O pacing

The recommended values are 33 and 24 for High and Low water marks respectively. If you want to change the recommended values, use the following formula:

High water mark = m x 4 + 1
Low water mark = n x 4

Where m and n are non-negative integers.

Note

The synchronize of the cluster configuration does *not* propagate the I/O pacing values to the other cluster nodes because this parameter is not stored in the HACMP ODM object classes. I/O pacing must be configured manually on each node.

See the *AIX Version 3.2 and 4 Performance Tuning Guide*, SC23-2365 for more information about I/O pacing.

6.3 Increase the syncd frequency

The syncd daemon is responsible for flushing all unwritten system buffers to disk. It is started automatically at IPL from the /sbin/rc.boot file, and is invoked by AIX every 60 seconds.

Adjusting the interval of the syncd daemon to 10 seconds forces more frequent I/O flushes to disk and reduces the risk of triggering the DMS due to heavy I/O traffic.

Before HACMP 4.4, you had to manually edit the /sbin/rc.boot file to change the syncd frequency. HACMP 4.4 introduces a new SMIT menu to do this. It is shown in Figure 163.

```
            Change/Show syncd frequency

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

                                                 [Entry Fields]
  syncd frequency (in seconds)                [10]                  #




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
```

Figure 163. Changing the syncd frequency

Note

The synchronize of the cluster configuration does *not* propagate the new syncd value to the other cluster nodes because it is not stored in the HACMP ODM object classes. It must be configured manually on each node.

6.4 Tuning the heartbeat rate in ES 4.4

All nodes in a cluster must be connected via one or more networks. Every supported network has a corresponding network module. Each network module maintains a connection with the network modules of the other nodes in the cluster. ES uses the network modules to exchange the keepalive packets among all cluster nodes. Each network module has some configuration parameters that determine the rate at which keepalive packets are exchanged. This rate is usually referred to as the heartbeat rate or the failure detection rate.

In ES 4.4, the heartbeat rate is made up of two fields:

Frequency The time interval between keepalive packets.

Sensitivity The number of consecutive keepalive packets that must be missed before the interface is considered to have failed.

The time needed to detect a failure can be calculated using this formula:

```
Frequency * Sensitivity * 2 seconds
```

6.4.1 Tuning the network interface modules

Figure 164 shows the SMIT menu that allows tuning of the heartbeat rate. It can be reached with the `smit cm_config_networks.chg.select` fastpath.

```
                 Change/Show a Cluster Network Module

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

                                                        [Entry Fields]
* Network Module Name                                  rs232
  Description                                         [RS232 Serial Protocol]
  Grace Period                                        [30]                #
  Failure Detection Rate                               Normal             +
  Failure Cycle                                       [4]                 #
  Heartbeat Rate (in tenths of a second)              [20]                #


  Note:  A changed value in the Heartbeat Rate
  field will be ignored if the Failure Detection
  Rate is not set to "Custom".

  Note:  Changes made in this panel must be
  propogated to the other nodes by synching
  topology.

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
```

Figure 164. The RS232 network interface module

Grace Period

The Grace Period is the time limit within which a network fallover must be taken care of.

Failure Detection Rate
This field can be set to one of these four values: Normal, Slow, Fast, or Custom. Normal, Slow, and Fast give predefined heartbeat rates. Custom must be set in order to change the default value of the “Failure Cycle” field.

Failure Cycle
The Failure Cycle is actually sensitivity, and determines the number of consecutive keepalive packets that must be missed before this interface is considered to have failed.

Heartbeat Rate
The Heartbeat Rate is actually frequency, and determines the time interval, in tenths of a second, between keepalive packets.

Note

In ES 4.4, frequency can not be lower than 1 second.

The values of the SMIT menu are saved in the ODM object class called HACMPnim, shown in Figure 165.

```
risc1# odmget -q name=rs232 HACMPnim

HACMPnim:
        name = "rs232"
        desc = "RS232 Serial Protocol"
        addrtype = 1
        path = ""
        para = ""
        grace = 30
        hbrate = 2000000
        cycle = 4
```

Figure 165. The HACMPnim object class

When you tune the network interface module, we recommend the following steps:

1. Change the Failure Detection Rate to a slower value, for example moving from fast to normal or from normal to slow.
2. Set the Failure Detection Rate to Custom and tune the Failure Cycle field by selecting a higher number.
3. Adjust the Heartbeat Rate by choosing a higher value.

Note

After any change to the network interface module parameters, the cluster topology must be synchronized.

6.5 New topology services AIX Error Log entries in ES 4.4

ES 4.4 relies upon RS/6000 Cluster Technology (RSCT) 1.2. A new feature of RSCT 1.2 is that topology services creates AIX Error Log entries when certain abnormal conditions occur. Because the RSCT software is used by both ES and PSSP, some error labels apply only to ES while others apply only to PSSP.

Prior to RSCT 1.2, topology services used to write information only to its log files, typically under the /var/ha directory. Since these log files contain very detailed information, it is often difficult to understand topology services activities. By having new entries created in AIX Error Log when abnormal situations occur, troubleshooting topology services problems becomes a much easier task.

Figure 166 on page 200 shows the complete list of all the topology services AIX Error Log labels.

```
risc1# errpt -t | grep TS_
00CCD298 TS_THCREATE_ER      PERM S  Cannot start. A thread can not be create
011080A2 TS_SECURITY_ST      INFO O  Message authentication failure in Topolo
0D00C3C4 TS_DMS_WARNING_ST   INFO S  DeadMan Switch (DMS) close to trigger
1D4610EC TS_SP_DIR_ER        PERM S  Cannot start. Cannot create working dire
1F7F2062 TS_IPADDR_ER        PERM S  Cannot start or refresh. Cannot convert
20D516B8 TS_SDR_ER           PERM S  Cannot start. Cannot access or update da
21C35623 TS_LIBERR_EM        PEND S  Topology Services client library encount
26ADF581 TS_SHMAT_ER         PERM S  Cannot start. Cannot attach to a shared
292374DE TS_LSOCK_ER         PERM S  Cannot start. Cannot open listening sock
2A188FDE TS_DUPNETNAME_ER    PERM S  Cannot start/refresh: duplicated network
2AB65A8D TS_SERVICE_ER       PERM S  Cannot start or refresh. No service entr
47E4956B TS_THREAD_STUCK_ER  PERM S  Main thread blocked: exiting
4D9226A5 TS_NODEDOWN_EM      PEND U  Remote nodes down
4D9226A5 TS_NODEDOWN_EM      PEND U  Remote nodes down
59F09C1D TS_LOGFILE_ER       PERM S  Cannot start or continue. Cannot open pr
5EBACC17 TS_SPIPDUP_ER       PERM S  Cannot start or refresh. IP address dupl
6453FE6E TS_SHMEMKEY_ER      PERM S  Cannot start. Cannot get interprocess co
645637FC TS_START_ST         INFO O  Topology Services daemon started
6EA7FC9E TS_MISCFG_EM        PEND U  Local adapter misconfiguration detected
81132988 TS_DCECRED_ER       PERM S  Cannot start. Cannot get DCE credentials
83F4EBF9 TS_LOC_DOWN_ST      INFO S  Possible malfunction on local adapter
854298A1 TS_INVALIDMESG_ST   INFO O  Received large number of invalid message
88522CA3 TS_CWSADDR_ER       PERM S  Cannot start. Control workstation addres
8D7CF8D9 TS_THATTR_ER        PERM S  Cannot start. Cannot create or destroy a
8E47EBFF TS_SPLOCAL_ER       PERM U  Local adapters missing in configuration
91A34651 TS_SHMGET_ER        PERM S  Cannot start. Cannot get a shared memory
95A9DAD0 TS_NODEUP_ST        INFO O  Remote down nodes came back up
9949BD20 TS_NODENUM_ER       PERM S  Cannot start. Cannot get local node numb
A204A4EE TS_STOP_ST          INFO O  Topology Services daemon stopped
A3E85343 TS_HAIPDUP_ER       PERM S  Cannot start or refresh. IP address dupl
A45AC96A TS_KEYS_ER          PERM S  Command hats_keys failed to get keyfile
A49627E6 TS_SPNODEDUP_ER     PERM S  Cannot start or refresh. Node number dup
B0107BA4 TS_ASSERT_EM        PEND S  Topology Services daemon exit abnormally
BEC6A0E0 TS_REFRESH_ER       PERM U  Error encountered during refresh operati
BEE2FB4A TS_DEATH_TR         UNKN U  Contact with a neighboring adapter lost
BFD0ADC5 TS_HALOCAL_ER       PERM U  Local adapters missing in configuration
C1FDC4E7 TS_AUTHMETH_ER      PERM S  Command lsauthpts failed to get authenti
C46498E2 TS_MIGRATE_ER       PERM S  Error encountered during migration-refre
C5D4E9F8 TS_IOCTL_ER         PERM S  Cannot retrieve network interface config
C95796E8 TS_CMDFLAG_ER       PERM S  Cannot start. Command-line flag incorrec
CB7E5EC7 TS_SECURITY2_ST     INFO O  More authentication failures during cert
D05D2F04 TS_HANODEDUP_ER     PERM S  Cannot start or refresh. Node number dup
D11173DE TS_MACHLIST_ER      PERM S  Cannot start or refresh. Cannot open con
D1BD179A TS_SECMODE_ER       PERM O  Local DCE security mode can not be verif
D730F82E TS_LATEHB_PE        PERF U  Late in sending heartbeat
DF534D29 TS_SEMGET_ER        PERM S  Cannot start. Cannot get or initialize a
EB38514E GS_TS_RETCODE_ER    PERM O  Connection failure between Group Service
EC7E7E0B TS_LONGLINE_ER      PERM S  Cannot start. Configuration line too lon
EEF083B9 TS_SYSPAR_ER        PERM S  Cannot start. Cannot get system partitio
FD20FB81 TS_CPU_USE_ER       PERM S  Using too much CPU: exiting
FD7A0E7E TS_UNS_SIN_TR       UNKN U  Local adapter disabled after unstable si
FFADC296 TS_RSOCK_ER         PERM S  Cannot start. Cannot open UDP socket for
```

Figure 166. The topology services AIX Error Log labels

6.5.1 Simulating an abnormal condition

We simulate here an abnormal condition in order to see the entries created in AIX Error Log by the topology services daemon.

Our two cluster nodes, risc1 and risc3, are connected via Ethernet and RS232 networks. Figure 167 and Figure 168 on page 202 show the definition of the heartbeat parameters for the two networks.

```
              Change/Show a Cluster Network Module

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

                                                        [Entry Fields]
* Network Module Name                                 ether
  Description                                         [Ethernet Protocol]
  Grace Period                                        [30]                #
  Failure Detection Rate                               Custom             +
  Failure Cycle                                       [4]                 #
  Heartbeat Rate (in tenths of a second)              [5]                 #


Note:  A changed value in the Heartbeat Rate
  field will be ignored if the Failure Detection
  Rate is not set to "Custom".

  Note:  Changes made in this panel must be
  propogated to the other nodes by synching
  topology.

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
```

Figure 167. The Ethernet heartbeat definition

```
              Change/Show a Cluster Network Module

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

                                                          [Entry Fields]
* Network Module Name                                   rs232
  Description                                          [RS232 Serial Protocol]
  Grace Period                                         [30]                #
  Failure Detection Rate                                Normal             +
  Failure Cycle                                        [4]                 #
  Heartbeat Rate (in tenths of a second)               [20]                #


Note:  A changed value in the Heartbeat Rate
  field will be ignored if the Failure Detection
  Rate is not set to "Custom".

  Note:  Changes made in this panel must be
  propogated to the other nodes by synching
  topology.

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
```

Figure 168. The RS232 heartbeat definition

After starting ES, we execute the `lssrc -ls topsvcs` command. This command, shown in Figure 169 on page 203, gives a perspective of the cluster status from the view point of topology services. For the purpose of our simulation, we have to keep in mind the line saying “trip interval = 16 seconds.” The trip interval determines the DMS timer.

```
risc1# lssrc -ls topsvcs
Subsystem         Group            PID     Status
 topsvcs          topsvcs          19288   active
Network Name   Indx Defd Mbrs St Adapter ID      Group ID
ennetwork_0    [ 0]    2    2  S 10.10.10.1      10.10.10.1
ennetwork_0    [ 0]             (10.10.10.10    )
ennetwork_0    [ 0] en0          0x81760b00      0x81760b14
HB Interval = 1 secs. Sensitivity = 4 missed beats
ennetwork_1    [ 1]    2    2  S 10.20.20.1      10.20.20.3
ennetwork_1    [ 1] en1          0x81760aac      0x81760abf
HB Interval = 1 secs. Sensitivity = 4 missed beats
rs232_0        [ 2]    2    2  S 255.255.0.0     255.255.0.1
rs232_0        [ 2] tty0         0x81760aad      0x81760ac3
HB Interval = 2 secs. Sensitivity = 4 missed beats
  2 locally connected Clients with PIDs:
haemd( 22222) hagsd( 10458)
  Dead Man Switch Enabled:
     reset interval = 1 seconds
     trip  interval = 16 seconds
  Configuration Instance = 2
  Default: HB Interval = 1 secs. Sensitivity = 4 missed beats
  Daemon employs no security
  Data segment size: 7637 KB. Number of outstanding malloc: 270
  User time 23 sec. System time 29 sec.
  Number of page faults: 53. Process swapped out 0 times.
  Number of nodes up: 2. Number of nodes down: 0.
```

Figure 169. The lssrc -ls topsvcs command

The topology services daemon is called topsvcs and runs at a very high priority of 31. In order to simulate an unusual condition, we have started a process with a priority of 30, which is more favored than topsvcs, and let this process run for 10 seconds. Basically, we have simulated a case where topology services has been starving for CPU time for 10 seconds.

Figure 170 shows the entries created in AIX Error Log by topology services when this abnormal condition occurred.

```
risc1# errpt
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
BEE2FB4A   0719165000 U U topsvcs        Contact with a neighboring adapter lost
BEE2FB4A   0719165000 U U topsvcs        Contact with a neighboring adapter lost
D730F82E   0719164900 P U topsvcs        Late in sending heartbeat
D730F82E   0719164900 P U topsvcs        Late in sending heartbeat
0D00C3C4   0719164900 I S hats           DeadMan Switch (DMS) close to trigger
```

Figure 170. The entries created in the error log

Figure 171 on page 204 shows the detailed information of the TS_DEATH_TR entry. The description of this entry, “Contact with a neighboring adapter lost,” is quite self-explanatory. Because the topology services daemon did not get control of the CPU for 10 seconds, this node was unable to receive keepalive packets from its cluster neighbor.

```
LABEL:          TS_DEATH_TR
IDENTIFIER:     BEE2FB4A

Date/Time:       Wed Jul 19 16:50:06
Sequence Number: 297
Machine Id:      000504936700
Node Id:         risc1
Class:           U
Type:            UNKN
Resource Name:   topsvcs
Resource Class:  NONE
Resource Type:   NONE
Location:        NONE
VPD:

Description
Contact with a neighboring adapter lost

Probable Causes
The neighboring adapter mal-functioned
Networking problem renders neighboring adapter unreachable

Failure Causes
The neighboring adapter mal-functioned
Problem with the network

        Recommended Actions
        Verify status of the faulty adapter
        Verify status of network

Detail Data
DETECTING MODULE
rsct,threephs.C,      1.135.1.5,3832
ERROR ID
.8hjsyyC8WRt.m0f.8cU08....................
REFERENCE CODE

The IP address of the faulty adapter
10.20.20.3
Node number where the adapter is located
           2
```

Figure 171. The TS_DEATH_TR label

Figure 172 on page 205 shows the detailed information of the TS_LATEHB_PE entry. Again, the description of this entry, “Late in sending heartbeat,” is self-explanatory. This entry is created when the amount of time that the topsvcs daemon was late in sending keepalive packets is equal to or greater than the amount of time needed to consider the network adapter down.

```
LABEL:          TS_LATEHB_PE
IDENTIFIER:     D730F82E

Date/Time:       Wed Jul 19 16:49:57
Sequence Number: 295
Machine Id:      000504936700
Node Id:         risc1
Class:           U
Type:            PERF
Resource Name:   topsvcs
Resource Class:  NONE
Resource Type:   NONE
Location:        NONE
VPD:

Description
Late in sending heartbeat

Probable Causes
Heavy CPU load
Severe physical memory shortage
Heavy I/O activities

Failure Causes
Daemon can not get required system resource

        Recommended Actions
        Reduce the system load

Detail Data
DETECTING MODULE
rsct,bootstrp.C,         1.135,1794
ERROR ID
.iUDALz38WRt.BPM18cU08...................
REFERENCE CODE

A heartbeat is late by the following number of seconds
           9
```

Figure 172. The TS_LATEHB_PE label

Figure 173 on page 206 shows the detailed information regarding the TS_DMS_WARNING_ST entry. This error indicates that the system is in a state where it may soon crash because of the DMS. This entry tells us that the DMS has been reset with a small time-to-trigger value on the timer. The time-to-trigger value is indicated towards the end of the entry, and it is equal to 5.934 seconds. The DMS trigger interval is 16 seconds, as indicated at the bottom of the entry.

```
LABEL:          TS_DMS_WARNING_ST
IDENTIFIER:     0D00C3C4

Date/Time:       Wed Jul 19 16:49:57
Sequence Number: 293
Machine Id:      000504936700
Node Id:         risc1
Class:           S
Type:            INFO
Resource Name:   hats

Description
DeadMan Switch (DMS) close to trigger

Probable Causes
Topology Services daemon cannot get timely access to CPU

User Causes
Excessive I/O load is causing high I/O interrupt traffic
Excessive memory consumption is causing high memory contention

        Recommended Actions
        Reduce application load on the system
        Change (relax) Topology Services tunable parameters
        Call IBM Service if problem persists

Failure Causes
Problem in Operating System prevents processes from running
Excessive I/O interrupt traffic prevents processes from running
Excessive virtual memory activity prevents Topology Services from making progres
s

        Recommended Actions
        Examine I/O and memory activity on the system
        Reduce load on the system
        Change (relax) Topology Services tunable parameters
        Call IBM Service if problem persists

Detail Data
DETECTING MODULE
rsct,haDMS_kex.c   1.3.1.1,532
ERROR ID

REFERENCE CODE

Time remaining until DMS triggers (in msec)
        5934
DMS trigger interval (in msec)
       16000
```

Figure 173. The TS_DMS_WARNING_ST label

To summarize, the DMS trigger interval is 16 seconds, we denied the topology services access to the CPU for 10 seconds, and for this reason topology services creates an entry in AIX Error Log to warn us that we were just 5.934 seconds away before the node would be crashed by the DMS.

> **Note**
>
> The DMS trigger interval can also be discovered in Figure 169 on page 203, in particular by looking at the line “trip interval = 16 seconds.”

Because the entry TS_DMS_WARNING_ST is a warning that the node may soon crash, some tuning needs to be performed as explained in Section 6.4, “Tuning the heartbeat rate in ES 4.4” on page 196.

6.5.2 The hatsdmsinfo command

The `hatsdmsinfo` command is part of RSCT 1.2 and can be very useful to track the statistics about the resets of the DMS timer that occur while an ES cluster is operating. Basically, each time the topsvcs daemon resets the timer, the remaining amount left on the timer is saved, as shown under the column “Time to Trigger” in Figure 174 on page 208. When looking at the output of the `hatsdmsinfo` command, the most useful lines are at the bottom, after the line saying “DMS Resets with small time-to-trigger”. The last line was stored when topology services created in AIX Error Log the TS_DMS_WARNING_ST entry. In fact the time stamp, 16:49:57, matches the amount of time left in the DMS timer, 5.934 seconds.

```
risc1# cd /usr/sbin/rsct/bin
risc1# ./hatsdmsinfo
Information for Topology Services -- HACMP/ES
DMS Trigger time: 16.000 seconds.
Last DMS Resets                         Time to Trigger (seconds)
07/19/00 16:55:36.730                   15.499
07/19/00 16:55:37.231                   15.500
07/19/00 16:55:37.731                   15.500
07/19/00 16:55:38.231                   15.500
07/19/00 16:55:38.732                   15.499
07/19/00 16:55:39.233                   15.500
07/19/00 16:55:39.733                   15.500

>>> omitted lines <<<

07/19/00 16:56:00.757                   15.499
07/19/00 16:56:01.258                   15.500
DMS Resets with small time-to-trigger   Time to Trigger (seconds)
Threshold value: 12.000 seconds.
07/19/00 16:49:57.840                    5.934
#
```

Figure 174. The hatsdmsinfo command

Note

The `hatsdmsinfo` command only works if ES is up and running. The values stored by `hatsdmsinfo` are reset to zero when the cluster is stopped. However, the entries in AIX Error Log remain.

Chapter 7. Administrative task enhancements

This chapter describes improvements for administrative tasks in HACMP 4.4. The new features are:

- Enhanced LVM TaskGuide
- Cluster Single Point of Control (C-SPOC) enhancements
- HACMP Logs on remote file systems

7.1 Enhanced LVM TaskGuide

The TaskGuide is a GUI that simplifies the task of creating a shared volume group within an HACMP cluster configuration. The TaskGuide presents a series of panels that guide you through the steps of specifying initial and sharing nodes, disks, concurrent or non-concurrent access, volume group name, physical partition size, and cluster settings. The TaskGuide can reduce errors, as it does not allow you to proceed with steps that conflict with the cluster's configuration. Online help panels give additional information to aid in each step.

The TaskGuide for creating shared volume groups was introduced in HACMP 4.3. In HACMP 4.4, the TaskGuide has two enhancements:

- TaskGuide adds a default JFS log at the time of creating a volume group.

  The TaskGuide automatically creates a JFS log after creating a non-concurrent shared volume group, as you would need to do manually when creating a shared volume group without the TaskGuide. You will not see any difference in the GUI.

Note

You may still need to rename and mirror the JFS log after creating the shared volume group.

- TaskGuide adds a physical location to the display while selecting disks.

  When using the TaskGuide for creating a cluster volume group in the HACMP 4.3, the Choose Physical Volume Group panel would not show which nodes the hdisks were members of. This caused confusion if different nodes had different hdisk numbers assigned to the same PVID. There was also no way of knowing which nodes were connected to the shared disks.

The LVM TaskGuide is enhanced to provide you with a physical location of the hdisk available for selection. You can view the nodes to which the available disks are physically connected as shown in Figure 175.

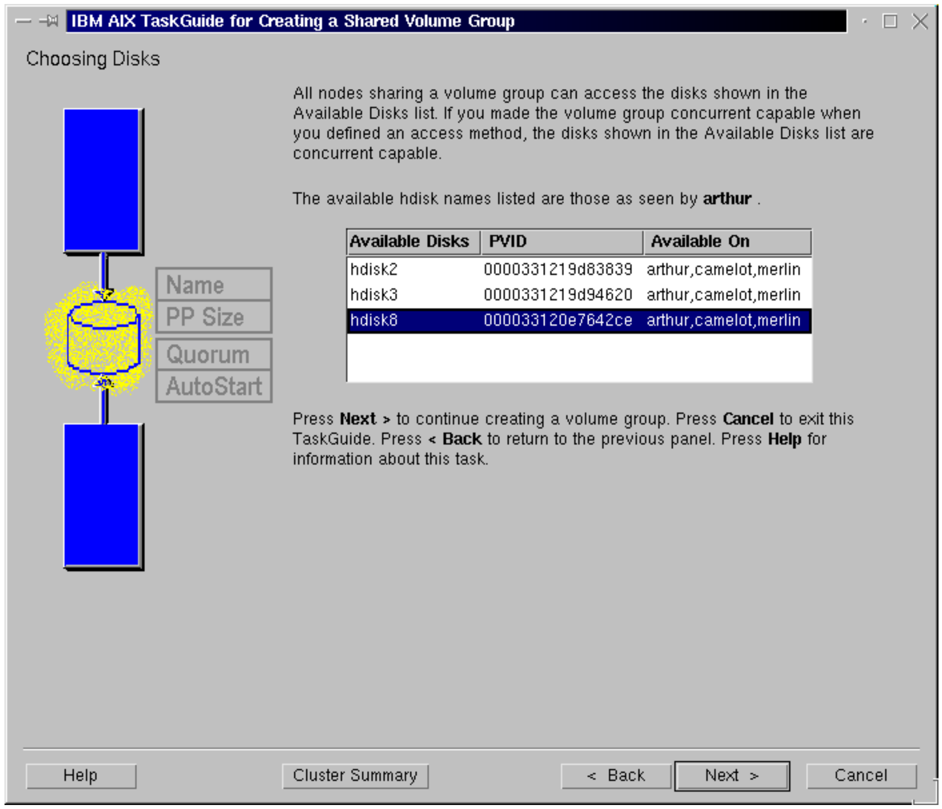

Figure 175. TaskGuide with more informations about disks

7.1.1 TaskGuide Requirements

Before starting the TaskGuide, make sure that:

- You have a configured HACMP cluster in place.
- You have the TaskGuide filesets installed.
- You are on a graphics capable terminal.
- You have set the display to your machine using your IP address or an alias. For example enter the `export` command:

```
# export DISPLAY=<your IP address>:0.0
```

- You have set the variable TERM. For example enter the `export` command:

```
# export TERM=xterm
```

7.1.2 Starting the TaskGuide

If you have all TaskGuide requirements, you can start the TaskGuide from the command line by typing the `cl_ccvg` command:

```
# /usr/sbin/cluster/tguides/bin/cl_ccvg
```

Alternatively, you can use SMIT as follows:

1. Enter the `smit cl_admin` fastpath and select **TaskGuide for Creating a Shared Volume Group**.

   After a pause, the TaskGuide "Welcome" panel appears.

2. Proceed through the panels to create or share a volume group.

   In the last panel, you have the option to cancel or to go back up and change what you have entered. If you are satisfied with your entries, click **Apply** to create the shared volume group as shown in Figure 176 on page 212.

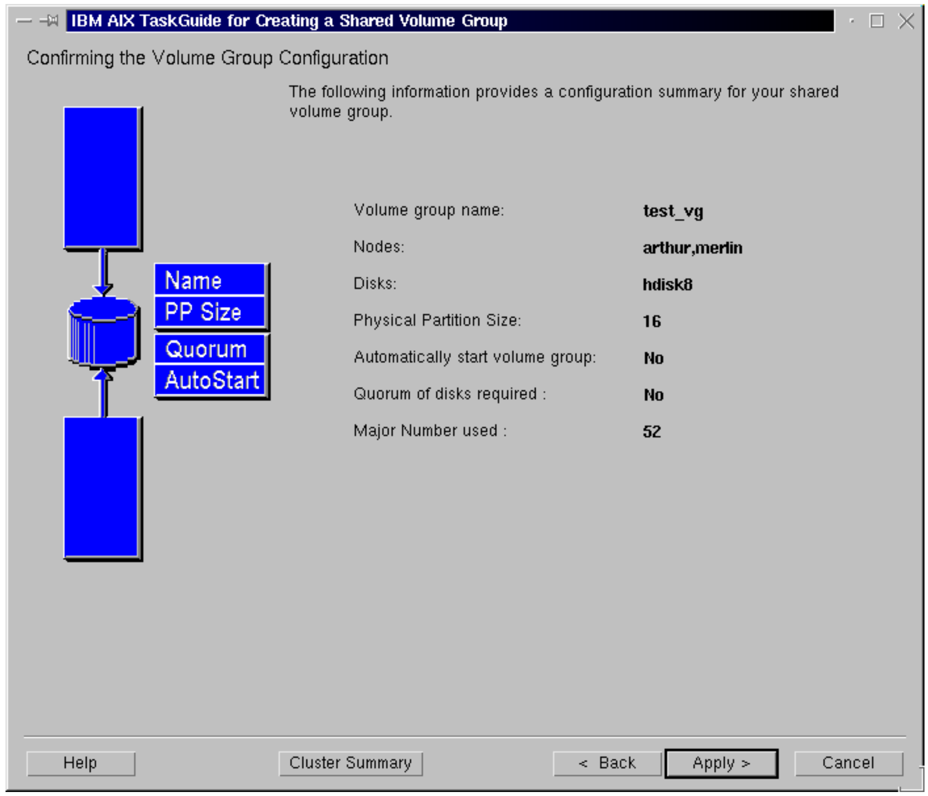

Figure 176. Last panel before applying the new shared volume group

7.2 C-SPOC file system enhancements

Before HACMP 4.4 you had to create a logical volume on a shared volume group before you could use C-SPOC to configure a file system.

With HACMP 4.4 you can either create a shared file system on an existing logical volume, or create a logical volume at the same time as the file system. Creating a logical volume at the same time as the file system (Add a Journaled File System) is a new feature in the HACMP 4.4. A logical volume is created automatically and is given a unique name in the cluster.

```
                         Shared File Systems

Move cursor to desired item and press Enter.

  Add a Journaled File System
  Add a Journaled File System on a Previously Defined Logical Volume
  List All Shared File Systems
  Change / Show Characteristics of a Shared File System
  Remove a Shared File System






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

Figure 177. Shared File System SMIT menu

In the following section we show how we add a shared JFS without previously defined cluster logical volume. We do not show how to add a shared JFS on a previously defined cluster logical volume (on a shared volume group), because there is nothing new. You can find out more about this operation in Chapter 4, “Maintaining Shared LVM Components” in *HACMP V4.3 AIX: Administration Guide*.

Before creating a shared JFS using C-SPOC, check that:

- All disk devices are properly attached to the cluster nodes.
- All disk devices are properly configured and available on all cluster nodes.
- The volume group that will contain the file system must be varied on at least one cluster node.
- From the SMIT all the nodes must be reachable.

> **Note**
>
> If you use the command, the last two restrictions can be overridden.

7.2.1 Creating a shared file system using SMIT

The following step creates a shared JFS without previously defined cluster logical volume:

1. Enter the `smit cl_lvsjfs` fastpath.
2. Select the volume group where the file system will be added. SMIT displays the menu for selecting file system attributes.

3. Enter a mount point and the file system attributes as follows:

```
                Add a Standard Journaled File System

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

[Entry Fields]
  Node Names                                         camelot,merlin
  Volume group name                                  camelotvg_1
* SIZE of file system (in 512-byte blocks)          [5000]
* MOUNT POINT                                       [/test_fs]
  PERMISSIONS                                        read/write      +
  Mount OPTIONS                                     []               +
  Start Disk Accounting?                             no              +
  Fragment Size (bytes)                              4096            +
  Number of bytes per inode                          4096            +
  Allocation Group Size (MBytes)                     16              +

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

SMIT checks the list of nodes that can own the resource group that contains the volume group, and creates the logical volume (on an existing log logical volume if present, otherwise it will create a new log logical volume). It adds the file system to the node where the volume group is varied on. All other nodes in the resource group will run the `importvg -L` command.

4. Configure the new file system under the HACMP (add it to the resource group, for instance) and synchronize resources.

7.2.2 Creating a shared file system using the command

We recommend that you use SMIT to add a shared file system. But you can also add a shared file system on specified nodes in the cluster by using the `cl_crlvfs` command. In this case you have more options than using SMIT. The syntax for this is as follows:

```
cl_crlvfs [-cspoc "[-f] [-g ResourceGroup | -n NodeList]"]
        -v VfsType -g VolumeGroup -m MountPoint -a size=Value
        [[-a Attribute=Value]...] [-u MountGroup] [-A {yes|no}]
        [-t {yes|no}] [-p {ro|rw}] [-l LogPartitions]
```

The `cl_crlvfs` command creates a logical volume and file system within a previously created volume group. If the volume group does not contain one, then the `cl_crlvfs` command will create and format a log logical volume as well. These actions are performed on a single node in the cluster. After completion on that node, the volume group description is imported to the remaining nodes that were specified.

Note

To use the cl_crlvfs command, an HACMP cluster must be defined and the topology configured. The underlying volume group must be varied on, except when the -cspoc "-f" flag is used (which is available from the command line only).

This command uses the AIX rsh facility to propagate commands to other nodes, and therefore requires the proper rsh access to all nodes. Thus each node must have a /.rhosts file that includes references to all boot and service interfaces for each cluster node.

New logical volumes created by the cl_crlvfs command will be named according to the scheme:

lv{0...9}[{0...9}...]   or   loglv{0...9}[{0...9}...]

If all nodes in the HACMP cluster are accessible at the time you execute the cl_crlvfs command, the names of any logical volumes created will be unique within the cluster. If one or more nodes is inaccessible and the -cspoc "-f" flag is not used, the command will fail. For more information, refer to Section 7.2.3, “Example” on page 216.

The -cspoc "-f" flag may be used to force execution of the cl_crlvfs command even when one or more nodes are inaccessible. However, in this case, unique logical volume names are not guaranteed. The -cspoc "-f" flag may also be used to force execution of the cl_crlvfs command when the underlying volume group is not varied on. In this case, the cl_crlvfs command will vary on the volume group on one node, create the file system, and then vary off the volume group again.

When this command fails, your intervention will be required to restore the state of the volume group, and/or remove unwanted logical volumes that may have been created.

You can find out more about the cl_crlvfs command and its flags in the manual, or in Appendix A “HACMP for AIX C-SPOC Commands” in *HACMP V4.3 AIX: Administration Guide*.

Note

C-SPOC commands can be executed from any cluster node, not necessarily from the node where you want to create a shared file system.

A shared file system can be also created through a DARE with the HACMP up and running on all nodes. Use the same procedure described in this chapter.

7.2.3 Example

This section provides an example that shows:

- How to add a shared file system where no logical volume is currently defined using the `cl_crlvfs` command and the SMIT.
- The name of an automatically created logical volume.
- If the name of a new logical volume is unique in the cluster.

Before we start, let us check logical volumes on our nodes. Node camelot has the following logical volume.

```
camelot# ls -l /dev/lv*
brw-rw----   1 root     system    10,  9 Jul 07 16:59 /dev/lv00
brw-rw----   1 root     system    48,  2 Jul 22 10:19 /dev/lv02
brw-rw----   1 root     system    47,  4 Jul 22 10:19 /dev/lv4
brw-rw----   1 root     system    47,  5 Jul 22 10:19 /dev/lv5
brw-rw----   1 root     system    47,  6 Jul 22 10:19 /dev/lv6
brw-rw----   1 root     system    48,  4 Jul 22 10:19 /dev/lv7

camelot#
```

Node merlin has the following logical volume.

```
merlin# ls -l /dev/lv*
brw-rw----   1 root     system    48,  2 Jul 22 10:38 /dev/lv02
brw-rw----   1 root     system    47,  4 Jul 25 10:35 /dev/lv4
brw-rw----   1 root     system    47,  5 Jul 25 10:35 /dev/lv5
brw-rw----   1 root     system    47,  6 Jul 25 10:35 /dev/lv6
brw-rw----   1 root     system    48,  4 Jul 22 10:38 /dev/lv7
brw-rw----   1 root     system    50,  2 Jul 22 10:38 /dev/lv8

merlin#
```

Node arthur has the following logical volume.

```
arthur# ls -l /dev/lv*
brw-rw----   1 root     system    50,  2 Jul 22 10:31 /dev/lv8

arthur#
```

1. Now we create the file system test_fs1 in the camelotvg_1 volume group. This volume group is part of the camelotrg resource group, and the participating nodes are camelot and merlin. We expect that the name of a new automatically created logical volume will be lv9 on both nodes camelot and merlin.

   We use the `cl_crlvfs` command on node camelot:

```
camelot# cd /usr/sbin/cluster/sbin
camelot# cl_crlvfs -cspoc -g'camelotrg' -v jfs -g'camelotvg_1' \
                         -a size='500' -m'/test_fs1' -a frag='4096' \
                         -a nbpi='4096' -a ag='8'
camelot: active
merlin: inactive
camelot: camelotlog_1
camelot: lv9
camelot: Based on the parameters chosen, the new /test_fs1 JFS file system
camelot: is limited to a maximum size of 134217728 (512 byte blocks)
camelot:
camelot: New File System size is 32768
merlin: camelotvg_1

camelot#
```

   Instead of the `-cspoc -g'camelotrg'` flag, you can also use the `-cspoc -n'camelot,merlin'` flag for the `cl_crlvfs` command.

   The `-cspoc -g` flag is used for a resource group name and the `-cspoc -n` flag is used for a list of nodes where we want to create a shared file system.

   As you can see from the output, the name of the automatically created local volume is lv9. We check this with the `ls -l` command on camelot:

```
camelot# ls -l /dev/lv*
brw-rw----   1 root     system    10,  9 Jul 07 16:59 /dev/lv00
brw-rw----   1 root     system    48,  2 Jul 22 10:19 /dev/lv02
brw-rw----   1 root     system    47,  4 Jul 22 10:19 /dev/lv4
brw-rw----   1 root     system    47,  5 Jul 22 10:19 /dev/lv5
brw-rw----   1 root     system    47,  6 Jul 22 10:19 /dev/lv6
brw-rw----   1 root     system    48,  4 Jul 22 10:19 /dev/lv7
brw-rw----   1 root     system    47,  7 Jul 25 10:34 /dev/lv9

camelot#
```

Also node merlin has a new logical volume, lv9:

```
merlin# ls -l /dev/lv*
brw-rw----   1 root     system    48,  2 Jul 22 10:38 /dev/lv02
brw-rw----   1 root     system    47,  4 Jul 25 10:35 /dev/lv4
brw-rw----   1 root     system    47,  5 Jul 25 10:35 /dev/lv5
brw-rw----   1 root     system    47,  6 Jul 25 10:35 /dev/lv6
brw-rw----   1 root     system    48,  4 Jul 22 10:38 /dev/lv7
brw-rw----   1 root     system    50,  2 Jul 22 10:38 /dev/lv8
brw-rw----   1 root     system    47,  7 Jul 25 10:35 /dev/lv9

merlin#
```

2. Then we create a logical volume lv10 on node arthur with the standard AIX `mklv` command as follows. This logical volume is only on node arthur.

```
arthur# ls -l /dev/lv*
brw-rw----   1 root     system    50,  2 Jul 22 10:31 /dev/lv8
brw-rw----   1 root     system    50,  7 Jul 25 10:37 /dev/lv10

arthur#
```

3. We create another file system test_fs2 in the camelotvg_2 volume group. This volume group is part of the camelotrg resource group, and the participating nodes are camelot and merlin. We expect that the name of a new automatically created logical volume will be lv11 on both nodes camelot and merlin. We use the `cl_crlvfs` command with the `-cspoc -n` flag as follows:

```
camelot# cd /usr/sbin/cluster/sbin
camelot# /cl_crlvfs -cspoc -n'camelot,merlin' -v jfs -g'camelotvg_2' \
                          -a size='500' -m'/test_fs2' -a frag='4096' \
                          -a nbpi='4096' -a ag='8'
camelot: active
merlin: inactive
camelot: camelotlog_2
camelot: lv11
camelot: Based on the parameters chosen, the new /test_fs1 JFS file system
camelot: is limited to a maximum size of 134217728 (512 byte blocks)
camelot:
camelot: New File System size is 32768
merlin: camelotvg_2

camelot#
```

The name of an automatically created local volume is lv11 as we expected. Node camelot has a new logical volume.

```
camelot# ls -l /dev/lv*
brw-rw----   1 root     system    10,  9 Jul 07 16:59 /dev/lv00
brw-rw----   1 root     system    48,  2 Jul 22 10:19 /dev/lv02
brw-rw----   1 root     system    48,  5 Jul 25 12:15 /dev/lv11
brw-rw----   1 root     system    47,  4 Jul 22 10:19 /dev/lv4
brw-rw----   1 root     system    47,  5 Jul 22 10:19 /dev/lv5
brw-rw----   1 root     system    47,  6 Jul 22 10:19 /dev/lv6
brw-rw----   1 root     system    48,  4 Jul 22 10:19 /dev/lv7
brw-rw----   1 root     system    47,  7 Jul 25 10:34 /dev/lv9

camelot#
```

Also, node merlin has a new logical volume:

```
merlin# ls -l /dev/lv*
brw-rw----   1 root     system    48,  2 Jul 25 12:16 /dev/lv02
brw-rw----   1 root     system    48,  5 Jul 25 12:16 /dev/lv11
brw-rw----   1 root     system    47,  4 Jul 25 10:35 /dev/lv4
brw-rw----   1 root     system    47,  5 Jul 25 10:35 /dev/lv5
brw-rw----   1 root     system    47,  6 Jul 25 10:35 /dev/lv6
brw-rw----   1 root     system    48,  4 Jul 25 12:16 /dev/lv7
brw-rw----   1 root     system    50,  2 Jul 22 10:38 /dev/lv8
brw-rw----   1 root     system    47,  7 Jul 25 10:35 /dev/lv9

merlin#
```

4. Finally we create the file system test_fs3 in the arthurvg_1 volume group. This volume group is part of the arthurrg resource group, and the participating nodes are arthur and merlin. arthur has two logical volumes named lv8 and lv10. We expect that the name of a new automatically created logical volume will be lv12 on both arthur and merlin.

We use SMIT in this case. After issuing the `smit cl_lvsjfs` fastpath, we select the arthurvg_1volume group, then we set up fields as follows:

```
                 Add a Standard Journaled File System

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

                                                       [Entry Fields]
  Node Names                                           arthur,merlin
  Volume group name                                    arthurvg_1
* SIZE of file system (in 512-byte blocks)            [50]
* MOUNT POINT                                         [/test_fs3]
  PERMISSIONS                                          read/write       +
  Mount OPTIONS                                       []                +
  Start Disk Accounting?                               no               +
  Fragment Size (bytes)                                4096             +
  Number of bytes per inode                            4096             +
  Allocation Group Size (MBytes)                       8                +

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 operation finished, a new file system is successfully created and the name of new logical volume is lv12:

```
                              COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

arthur: active
merlin: inactive
arthur: alg_1
arthur: lv12
arthur: Based on the parameters chosen, the new /test_fs3 JFS file system
arthur: is limited to a maximum size of 134217728 (512 byte blocks)
arthur:
arthur: New File System size is 32768
merlin: arthurvg_1

F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next
```

We check new logical volume lv12 on node arthur:

```
arthur# ls -l /dev/lv*
brw-rw----   1 root     system    50,  7 Jul 25 10:37 /dev/lv10
brw-rw----   1 root     system    49,  3 Jul 25 12:38 /dev/lv12
brw-rw----   1 root     system    50,  2 Jul 22 10:31 /dev/lv8

arthur#
```

We also check new logical volume lv12 on node merlin as follows.

```
merlin# ls -l /dev/lv*
brw-rw----   1 root     system    48,  2 Jul 25 12:16 /dev/lv02
brw-rw----   1 root     system    48,  5 Jul 25 12:16 /dev/lv11
brw-rw----   1 root     system    49,  3 Jul 25 12:42 /dev/lv12
brw-rw----   1 root     system    47,  4 Jul 25 10:35 /dev/lv4
brw-rw----   1 root     system    47,  5 Jul 25 10:35 /dev/lv5
brw-rw----   1 root     system    47,  6 Jul 25 10:35 /dev/lv6
brw-rw----   1 root     system    48,  4 Jul 25 12:16 /dev/lv7
brw-rw----   1 root     system    50,  2 Jul 22 10:38 /dev/lv8
brw-rw----   1 root     system    47,  7 Jul 25 10:35 /dev/lv9

merlin#
```

As we see through the example, HACMP 4.4 automatically creates logical volumes with unique names during the add a shared file system process. Names are in sequence like; lv8, lv9, lv10, lv11. If the name already exists, then the new name will be the next “free” name.

7.3 C-SPOC password configuration enhancements

Before HACMP 4.4, when you configure a user account for a cluster or nodes in a resource group, you need to set a password for the account on each node individually. From HACMP 4.4, you can set a password over the whole cluster or per resource group, like other account parameters.

Note, however, that this feature is only for setting up *initial* passwords. Users will be prompted to change their passwords when they log in, and users cannot use this feature to propagate their own passwords.

For security purposes, the clear password is never transmitted over the network, only the encrypted password from the file /etc/security/passwd. Hence, the user whose password is being configured must have an account on the local node.

You can configure user passwords using the SMIT or the command, if user is defined on local node. Otherwise the command fails. Note that all C-SPOC commands can be executed from any cluster node, not necessarily from the node where you want to set an initial passwords. You can also set passwords with the HACMP up and running on all nodes.

7.3.1 Changing password using SMIT

Using the `smit cl_usergroup` fastpath opens the following SMIT menu.

```
                       Cluster Users & Groups

Move cursor to desired item and press Enter.

  Users
  Groups
  Passwords






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

From the menu select **Passwords**. You will see the following SMIT menu:

```
                 Change a User's Password in the Cluster

Type or select a value for the entry field.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
  Select nodes by Resource Group                        []              +
*** No selection means all nodes! ***





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

In this menu, you can either choose a resource group or leave the field empty, which will select all nodes.

7.3.2 Changing password using the command

If you use the cl_chpasswd command, you have more options. For example, you can configure arbitrary sets of nodes. The command changes the user's password on all nodes of an HACMP cluster.

Note

Do not use the cl_chpasswd command if you have a Network Information Service (NIS) database installed on any cluster node. The command in this environment can cause database inconsistencies.

The syntax for the cl_chpasswd command is as follows:

```
cl_chpasswd [-cspoc "[-f] [-g ResourceGroup | -n NodeList]"] UserName
```

If you use the cl_chpasswd command only with a username and no other flags, the cl_chpasswd command changes the password on all cluster nodes where the user is defined.

The following is the requirements for successful execution of the cl_chpasswd command:

- The user name must exist on a local node (or the command will fail).
- The user name must exist on every cluster node in the node list.
- All cluster nodes must be running and accessible.

The flags have the following functions:

-cspoc
: Used to specify the following C-SPOC flags:

-f
: Forces the cl_chpasswd command to skip default verification. If this flag is set and a cluster node is not accessible, cl_chpasswd reports a warning and continues execution on the other cluster nodes. If the flag is set and the user does not exist on a cluster node, cl_chpasswd reports a warning and continues execution on the other cluster nodes.

-g ResourceGroup
: Generates a list of nodes participating in the resource group on which the cl_chpasswd command will be executed.

-n NodeList
: Specifies the node(s) on which the C-SPOC command will be

executed. You can specify more than one node by separating each node in the list with a comma.

If you do not specify either the `-g` or `-n` flags, the default action occurs on all cluster nodes.

Note

If any node in the cluster is a member of an SP with the `usermgmt_config` flag set to *true*, the `cl_chpasswd` command fails.

This command uses the AIX `rsh` facility to propagate commands to other nodes, and therefore requires the proper `rsh` access to all nodes (unless you are using Kerberos on your system). Thus, each node must have a /.rhosts file that includes references to all boot and service interfaces for each cluster node.

You can find out more about the `cl_chpasswd` command in the manual.

7.3.3 Examples

The examples in this section show you how to change the password for the user jane. You have the user jane defined on all nodes (arthur, camelot, and merlin) in the cluster. User jane is also a member of the arthurrg resource group. Participating nodes in this resource group are arthur and merlin. You want to set the initial password for user jane

7.3.3.1 Setting the initial password using the command

The following example uses the `cl_chpasswd` command:

- To change the password for the user jane on all cluster nodes, enter the command as follows:

  ```
  # /usr/sbin/cluster/sbin/cl_chpasswd jane
  ```

  The system will prompt for the new password twice. If both match it will change the jane's password on all cluster nodes.

- To specify one or more cluster nodes on which to execute the `cl_chpasswd` command, enter the commands as follows:

  ```
  # cd /usr/sbin/cluster/sbin/cl_chpasswd -cspoc \
  "-n arthur,camelot,merlin" jane
  ```

  The system changes the jane's password on nodes arthur, camelot, and merlin.

- To specify a resource group on which to execute the `cl_chpasswd` command, enter the command as follows:

```
# /usr/sbin/cluster/sbin/cl_chpasswd -cspoc "-g camelotrg" jane
```

  The system changes the jane's password on resource group camelotrg.

7.3.3.2 Setting the initial password using SMIT

You can set the initial password using SMIT. Enter the `smit cl_chpasswd` fastpath to get SMIT menu shown in Figure 178. Select the arthurrg resource group and enter jane as the user name.

```
                Change a User's Password in the Cluster

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

                                                          [Entry Fields]
  Selection nodes by resource group                     arthurrg
   *** No selection means all nodes! ***

* User NAME                                            [jane]             +




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

Figure 178. Change a User's Password in the Cluster

Note

When you use resource group name to set an initial password, you need to pay attention to the following:

- If you use the command or SMIT on node camelot, which is not a member of the arthurrg resource group, you will set the initial password for the user jane on nodes arthur and merlin as well as camelot.
- If you want to set the initial password for the user jane only on nodes of arthurrg which member is arthur and merlin, you need to use the command or SMIT on either node arthur or merlin.

7.4 HACMP logs on non-local file systems

In HACMP 4.3.1 (without PTF IY11251), you can store the cluster system logs on a remote file system without any warnings. Moving cluster system logs on a remote file system is sometimes useful. However, it could be dangerous. Logs should not be redirected to shared file systems or NFS file systems. If you have logs on those file systems you may have problems if the file system needs to unmount during a fall over event.

In HACMP 4.4, the SMIT menu has a new Allow Logs on Remote Filesystems field, the default of which is false. You must set this field to true if you want to configure a remote log. Otherwise you will get warning messages.

```
                  Change/Show a Cluster Log Directory

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

                                                    [Entry Fields]
  Cluster Log Name                                   hacmp.out
  Cluster Log Description                            Generated by event sc>
  Default Log Destination Directory                  /tmp
* Log Destination Directory                          []
  Allow Logs on Remote Filesystems                    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 179. Change/Show a Cluster Log Directory

Even if you set the Allow Logs on Remote Filesystems field to true, you still may get warning messages during the verification that is part of the resource synchronization process. For an example, refer to the next section.

The system prevents you from configuring logs on file systems controlled by the HACMP. This means it is not possible to set the cluster logs on a file system that is part of any volume group under the HACMP control. Otherwise you get an error message similar to Figure 180 on page 227.

```
                         COMMAND STATUS

Command: failed       stdout: yes          stderr: no

Before command completion, additional instructions may appear below.

cllog:  The directory /cfs_1,
specified for log file hacmp.out,
is part of the filesystem /cfs_1, which is managed by HACMP.
cllog:  ERROR:  Therefore, it cannot be used for this purpose.




F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next
```

Figure 180. SMIT menu with cllog warning

If you configure logs under the HACMP 4.3.1 and you apply the PTF, the default value of the Allow Logs on Remote Filesystems field is still true, while in HACMP 4.4 it is false.

7.4.1 Customizing Log Files, example

Should you redirect a log file to a directory of your choice, keep in mind that the requisite (upper limit) disk space for most cluster logs is 2 MB. 14 MB is recommended for hacmp.out log file.

If you want to redirect a cluster log from its default directory to another destination, take the following steps:

1. On every cluster node create a directory that is a mount point for the remote file system.

> **Note**
>
> You must create the log destination directory locally on all nodes for proper functionality. Don’t forget to mount the remote file system on all nodes. If you will not do this, the cluster logs will be in directory that you created under root.

2. Enter the `smit clusterlog_redir.select` fastpath.

   SMIT displays a list of cluster log files with short descriptions as shown in Figure 181 on page 228.

```
                 Select a Cluster Log Directory

 Move cursor to desired item and press Enter.

   cluster.log      - Generated by cluster scripts and daemons
   cluster.mmdd     - Cluster history files generated daily
   cm.log           - Generated by the clstrmgr daemon
   dms_loads.out    - Generated by deadman's switch activity
   hacmp.out        - Generated by event scripts and utilities
   emuhacmp.out     - Generated by the event emulator scripts
   cspoc.log        - Generated by CSPOC commands



 F1=Help                F2=Refresh             F3=Cancel
 F8=Image               F10=Exit               Enter=Do
 /=Find                 n=Find Next
```

Figure 181. Select a Cluster Log Directory

3. Select a log that you want to redirect.

   SMIT displays a menu with the selected log's name, description, destination directory, new destination directory, and the Allow Logs on Remote Filesystems field. See Figure 182 where we choose the hacmp.out log file.

```
                  Change/Show a Cluster Log Directory

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

                                                    [Entry Fields]
  Cluster Log Name                                   hacmp.out
  Cluster Log Description                            Generated by event sc>
  Default Log Destination Directory                  /tmp
* Log Destination Directory                          [/HA_logs]
  Allow Logs on Remote Filesystems                    true              +




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

Figure 182. Change a Cluster Log Directory

4. Edit `Log Destination Directory` field to change the default path name. In Figure 182 on page 228, we changed a log destination directory to /HA_logs and the Logs on Remote Filesystems field to true.

5. After you change a log destination directory, a prompt appears reminding you to synchronize cluster resources from this node (cluster log ODM must be identical across the cluster). The cluster log destination directories as stored on this node will be synchronized to all nodes in the cluster.

   During the synchronization you will see a message similar to Figure 183.

```
>>>>>>>> omitted lines <<<<<<<<

Verifying cluster log directories on Node: arthur

>>>>>>>> omitted lines <<<<<<<<

cm.log           9292 K disk space left on filesystem.
dms_loads.out    9292 K disk space left on filesystem.
hacmp.out:
cllog:  The directory /HA_logs,
specified for log file hacmp.out,
is part of the NFS-mounted filesystem camelot:/HA_logs.
cllog: WARNING: As a result, it could unexpectedly become unavailable.
hacmp.out        1110828 K disk space left on filesystem.
emuhacmp.out     9292 K disk space left on filesystem.

>>>>>>>> omitted lines <<<<<<<<

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

Figure 183. Part of the SMIT menu with cllog warnings

Note that log destination directory changes will take effect when you synchronize cluster resources. If the cluster is not up then, they will take effect the next time the cluster services are restarted.

Chapter 8. HACMP 4.4 and NFS

This chapter discusses the configuration of the Network File System (NFS) under HACMP 4.4.

Starting from 4.4, the HANFS product is no longer available, so HANFS 4.3.1 will be the last release. HANFS is no longer being shipped because both HAS 4.4 and ES 4.4 now include all the functionality previously present only in the HANFS software.

The following are the NGF functionalities in HACMP 4.4:

- New NFS cross mount syntax
- Capability to export a filesystem or a directory
- Capability to specify an alternate exports file
- Preservation of NFS locks upon takeover (2-node clusters *only*)
- Capability to perform the NFS mount over a specific network
- Improved cluster verification

All of this will be explained in more detail.

8.1 New NFS cross mount syntax

If we have 2 nodes in the cluster, NFS cross mount allows one node to behave as the NFS server while the other node is the NFS client.

Starting with HACMP 4.3.1, the configuration parameters of NFS cross mount have added to previous releases. Figure 184 on page 232 shows the NFS cross mount configuration parameters in SMIT. This SMIT menu can be reached with the `smit cm_cfg_res.select` fastpath.

```
          Change/Show Resources/Attributes for a Resource Group

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

[TOP]                                                    [Entry Fields]
  Resource Group Name                                  resgrp1
  Node Relationship                                    cascading
  Participating Node Names                             risc1 risc3

  Service IP label                                    [risc1_svc]           +
  Filesystems                                         [/fs1]                +
  Filesystems Consistency Check                        fsck                 +
  Filesystems Recovery Method                          sequential           +
  Filesystems/Directories to Export                   [/fs1]                +
  Filesystems/Directories to NFS mount                [/nfsfs1;/fs1]        +
  Network For NFS Mount                               []                    +
  Volume Groups                                       [extvg]               +
  Concurrent Volume groups                            []                    +
  Raw Disk PVIDs                                      []                    +
  Connections Services                                []                    +
  Fast Connect Services                               []                    +
  Application Servers                                 [imageappsrvr]        +
  Highly Available Communication Links                []                    +
  Miscellaneous Data                                  []

  Inactive Takeover Activated                          false                +
  Cascading Without Fallback Enabled                   false                +
  9333 Disk Fencing Activated                          false                +
  SSA Disk Fencing Activated                           false                +
  Filesystems mounted before IP configured             true                 +
[BOTTOM]


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
```

Figure 184. NFS cross mount configuration parameters

Cluster node risc1 will be the NFS server, while node risc3 will be the NFS client.

Note

NFS cross mount is *only* supported in cascading resource groups. Rotating and concurrent resource groups do not support it.

The following fields are relevant to the NFS cross mount definition:

Filesystems
Specify the list of filesystems that are mounted locally by the cluster node having control of this resource group. We only have one, /fs1.

Filesystems/Directories to Export
Specify the list of filesystems or directories to be NFS exported by the cluster node that has the filesystem mounted locally. We export /fs1.

Filesystems/Directories to NFS Mount
Specify the list of filesystems or directories to be mounted via NFS by all nodes in the cluster who do not have /fs1 mounted locally. The new syntax to use in this field is "/nfsmountpoint;/localmountpoint." In our configuration, /fs1 represents the /localmountpoint directory while /nfsfs1 is the /nfsmountpoint directory.

Note

The /nfsfs1 directory must be created manually on all cluster nodes. Otherwise, clverify will report an error during the synchronization of the cluster resources.

Filesystems mounted before IP configured
This field can be set to either true or false, with false being the default value. When configuring NFS cross mount, it is mandatory to select true. Refer to the next section for details.

8.1.1 Filesystems mounted before IP configured parameter

The easiest way to explain the meaning of this parameter is comparing the two examples. By setting the parameter to *false*, the order of execution of the events at cluster startup is:

1. node_up_local
2. acquire_service_addr risc1_svc
3. get_disk_vg_fs /fs1 extvg

See Figure 185 on page 234 for details.

```
risc1# tail -f /usr/sbin/cluster/history/cluster.07252000
Jul 25 16:21:29 EVENT START: node_up risc1
Jul 25 16:21:33 EVENT START: node_up_local
Jul 25 16:21:35 EVENT START: acquire_service_addr risc1_svc
Jul 25 16:21:47 EVENT START: acquire_aconn_service en0 ennetwork
Jul 25 16:21:48 EVENT START: swap_aconn_protocols en0 en1
Jul 25 16:21:49 EVENT COMPLETED: swap_aconn_protocols en0 en1
Jul 25 16:21:49 EVENT COMPLETED: acquire_aconn_service en0 ennetwork
Jul 25 16:21:50 EVENT COMPLETED: acquire_service_addr risc1_svc
Jul 25 16:21:50 EVENT START: get_disk_vg_fs /fs1 extvg
Jul 25 16:22:15 EVENT COMPLETED: get_disk_vg_fs /fs1 extvg
Jul 25 16:22:16 EVENT COMPLETED: node_up_local
Jul 25 16:22:17 EVENT COMPLETED: node_up risc1
Jul 25 16:22:18 EVENT START: node_up_complete risc1
Jul 25 16:22:30 EVENT START: node_up_local_complete
Jul 25 16:22:31 EVENT START: start_server imageappsrvr
Jul 25 16:22:32 EVENT COMPLETED: start_server imageappsrvr
Jul 25 16:22:33 EVENT COMPLETED: node_up_local_complete
Jul 25 16:22:35 EVENT COMPLETED: node_up_complete risc1
```

Figure 185. Filesystems mounted before IP configured set to false

With this order of execution, during a takeover HACMP would first acquire the failed node IP address (acquire_service_addr risc1_svc) and *then* the volume group and filesystem (get_disk_vg_fs /fs1 extvg). This sequence of events often results in “missing file or filesystems” error messages from NFS clients because they can communicate with the NFS server via TCP/IP, but cannot access the data on the external disks.

On the other hand, setting the parameter to *true* changes the sequence of execution to:

1. node_up_local
2. get_disk_vg_fs /fs1 extvg
3. acquire_service_addr risc1_svc

thus preventing error messages from NFS clients.

See Figure 186 on page 235 for details.

```
risc1# tail -f /usr/sbin/cluster/history/cluster.07252000
Jul 25 16:47:02 EVENT START: node_up risc1
Jul 25 16:47:05 EVENT START: node_up_local
Jul 25 16:47:06 EVENT START: get_disk_vg_fs /fs1 extvg
Jul 25 16:47:30 EVENT COMPLETED: get_disk_vg_fs /fs1 extvg
Jul 25 16:47:32 EVENT START: acquire_service_addr risc1_svc
Jul 25 16:47:46 EVENT START: acquire_aconn_service en0 ennetwork
Jul 25 16:47:47 EVENT START: swap_aconn_protocols en0 en1
Jul 25 16:47:47 EVENT COMPLETED: swap_aconn_protocols en0 en1
Jul 25 16:47:48 EVENT COMPLETED: acquire_aconn_service en0 ennetwork
Jul 25 16:47:48 EVENT COMPLETED: acquire_service_addr risc1_svc
Jul 25 16:47:49 EVENT COMPLETED: node_up_local
Jul 25 16:47:49 EVENT COMPLETED: node_up risc1
Jul 25 16:47:50 EVENT START: node_up_complete risc1
Jul 25 16:48:03 EVENT START: node_up_local_complete
Jul 25 16:48:04 EVENT START: start_server imageappsrvr
Jul 25 16:48:05 EVENT COMPLETED: start_server imageappsrvr
Jul 25 16:48:07 EVENT COMPLETED: node_up_local_complete
Jul 25 16:48:09 EVENT COMPLETED: node_up_complete risc1
```

Figure 186. Filesystems mounted before IP configured set to true

8.1.2 Starting the cluster

After starting HACMP, we look at the filesystems mounted on each node. Figure 187 shows the situation of node risc1 where /fs1 has been mounted locally and /nfsfs1 has been mounted via NFS.

```
risc1# df
Filesystem    512-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4           32768     18984   43%     1527    19% /
/dev/hd2          966656     63576   94%    17989    15% /usr
/dev/hd9var        40960     27496   33%      430     9% /var
/dev/hd3           49152     40376   18%       75     2% /tmp
/dev/hd1          155648    146016    7%      182     1% /home
/dev/lv00         802816     88032   90%       50     1% /SW
/dev/tivolilv     507904    128128   75%     3307     6% /tivoli
/dev/extlv1        81920     77648    6%       40     1% /fs1
risc1_svc:/fs1      81920     77648    6%        -     -  /nfsfs1
#
```

Figure 187. The mounts on node risc1

Note

All applications running on node risc1 must access the data *strictly* through the NFS mount point /nfsfs1, not through the /fs1 directory.

Figure 188 shows that on the standby node risc3 the filesystem has been mounted only via NFS on the /nfsfs1 mount point.

```
risc3# df
Filesystem    512-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4           32768     19232   42%     1509    19% /
/dev/hd2          933888     68872   93%    17914    16% /usr
/dev/hd9var        32768     21952   34%      343     9% /var
/dev/hd3           49152     44488   10%       68     2% /tmp
/dev/hd1           32768     31136    5%       75     2% /home
/dev/tivolilv     507904    250336   51%     3111     5% /tivoli
risc1_svc:/fs1      81920     77648    6%       40     1% /nfsfs1
#
```

Figure 188. The mounts on node risc3

8.1.3 Situation after takeover

Figure 189 shows what happens when the primary node of the resource group, risc1 in our case, fails. The standby node risc3 just locally mounts the /fs1 filesystem. When the takeover is occurring, applications accessing the data through the /nfsfs1 mount point briefly hang until node risc3 has acquired all the resources. But as soon as the risc1_svc IP address, the extvg volume group, and the /fs1filesystem are available, the applications on node risc3 are able to access the data transparently via the NFS mount point /nfsfs1. During takeover, the /nfsfs1 NFS mounted filesystem is not unmounted.

```
risc3# df
Filesystem    512-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4           32768     19216   42%     1512    19% /
/dev/hd2          933888     68824   93%    17916    16% /usr
/dev/hd9var        32768     21616   35%      347     9% /var
/dev/hd3           49152     43872   11%       75     2% /tmp
/dev/hd1           32768     31168    5%       75     2% /home
/dev/tivolilv     507904    250336   51%     3111     5% /tivoli
risc1_svc:/fs1      81920     77648    6%       40     1% /nfsfs1
/dev/extlv1        81920     77648    6%       40     1% /fs1
#
```

Figure 189. The mounts on node risc3 after takeover

8.2 Capability to export a filesystem or a directory

Before HACMP 4.4, NFS could only export filesystems. The new NFS implementation allows you to chose between exporting filesystems or directories. Figure 190 shows the definition in SMIT to export the /fs1/dir1 directory instead of the filesystem mount point /fs1.

```
           Change/Show Resources/Attributes for a Resource Group

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

[TOP]                                                   [Entry Fields]
  Resource Group Name                                 resgrp1
  Node Relationship                                   cascading
  Participating Node Names                            risc1 risc3

  Service IP label                                   [risc1_svc]           +
  Filesystems                                        [/fs1]                +
  Filesystems Consistency Check                       fsck                 +
  Filesystems Recovery Method                         sequential           +
  Filesystems/Directories to Export                  [/fs1/dir1]           +
  Filesystems/Directories to NFS mount               [/nfsfs1;/fs1/dir1]   +
  Network For NFS Mount                              [ennetwork]           +
  Volume Groups                                      [extvg]               +
  Concurrent Volume groups                           []                    +
  Raw Disk PVIDs                                     []                    +
  Connections Services                               []                    +
  Fast Connect Services                              []                    +
  Application Servers                                [imageappsrvr]        +
  Highly Available Communication Links               []                    +
  Miscellaneous Data                                 []

  Inactive Takeover Activated                         false                +
  Cascading Without Fallback Enabled                  false                +
  9333 Disk Fencing Activated                         false                +
  SSA Disk Fencing Activated                          false                +
  Filesystems mounted before IP configured            true                 +
[BOTTOM]

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
```

Figure 190. Exporting a directory

Figure 191 on page 238 and Figure 192 on page 238 show the situation with HACMP up and running.

```
risc1# df
Filesystem    512-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4           32768     18976   43%     1527    19% /
/dev/hd2          966656     63008   94%    17994    15% /usr
/dev/hd9var        40960     23768   42%      520    11% /var
/dev/hd3           49152     37376   24%      101     2% /tmp
/dev/hd1          155648    145864    7%      189     1% /home
/dev/lv00         802816     88032   90%       50     1% /SW
/dev/tivolilv     507904    128128   75%     3307     6% /tivoli
/dev/extlv1        81920     77640    6%       41     1% /fs1
risc1_svc:/fs1/dir1      81920     77640    6%        -     -  /nfsfs1
#
```

Figure 191. The /fs1/dir1 directory on risc1

```
risc3# df
Filesystem    512-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4           32768     19224   42%     1511    19% /
/dev/hd2          933888     68672   93%    17918    16% /usr
/dev/hd9var        32768     20080   39%      389    10% /var
/dev/hd3           49152     42488   14%       86     2% /tmp
/dev/hd1           32768     31040    6%       83     3% /home
/dev/tivolilv     507904    250336   51%     3111     5% /tivoli
risc1_svc:/fs1/dir1      81920     77640    6%       41     1% /nfsfs1
#
```

Figure 192. The /fs1/dir1 directory on risc3

8.3 Capability to specify an alternate exports file

The default NFS export options are read/write for everybody, root access to all cluster nodes, and client access to every client host. In some circumstances these default options may not be desirable; for example, we may want to give client access only to one specific host. The new NFS implementation allows an alternate exports file called /usr/sbin/cluster/etc/exports. When HACMP starts, it checks for the existence of this file, and, if it is found, the filesystems are exported using the options specified here. Therefore, NFS uses the default export options.

Note

The alternate exports file is also available in HACMP 4.3.1 by installing APAR IY05357.

An example of an alternate exports file is shown in Figure 193, which shows the SMIT menu reachable with the `smit mknfsexp` fastpath. This configures the /usr/sbin/cluster/etc/exports alternate exports file. In this example we give root access to both cluster nodes risc1 and risc3. However, we give client access only to a host called client.

```
              Add a Directory to Exports List

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

                                                        [Entry Fields]
* PATHNAME of directory to export                    [/fs1]                   /
* MODE to export directory                            read-write             +
  HOSTS & NETGROUPS allowed client access            [client]
  Anonymous UID                                      [-2]
  HOSTS allowed root access                          [risc1_svc risc3_svc]
  HOSTNAME list. If exported read-mostly             []
  Use SECURE option?                                  no                     +
  Public filesystem?                                  no                     +
* EXPORT directory now, system restart or both        restart                +
  PATHNAME of alternate Exports file                 [/usr/sbin/cluster/etc/>


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
```

Figure 193. Configuring the alternate exports file

Note

The /usr/sbin/cluster/etc/exports file should be created on only one cluster node. The synchronization of the cluster resources will propagate this file to all the other nodes in the cluster.

The NFS cross mount configuration parameters used for this example are the same ones shown in Figure 184 on page 232. After starting HACMP, we check the NFS export options used. As shown in Figure 194 on page 240, the NFS server node risc1 has used the options specified in the alternate exports file to give client access only to the host called client and give root access to both cluster nodes.

```
risc1# exportfs
/fs1 -root=risc1_svc:risc3_svc,access=client
#
```

Figure 194. The NFS export options

8.4 Preservation of NFS locks upon takeover

The new NFS implementation also includes the capability to preserve NFS locks upon takeover. This capability is, however, present in two-node clusters *only*.

Note

This capability is also available in HACMP 4.3.1 by installing the fixes of APARs IX88399, IX84550, and IX88459.

8.5 Capability to perform the NFS mount over a specific network

The new NFS implementation introduces the possibility to specify on which physical network all the NFS mounts are performed on. This is possible thanks to a new field called "Network for NFS Mount," which is highlighted in Figure 195 on page 241. SMIT allows you to specify in this field the network name as it is defined in the cluster topology. HACMP performs all the NFS mounts using the service IP label belonging to this network name.

```
         Change/Show Resources/Attributes for a Resource Group

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

[TOP]                                                    [Entry Fields]
  Resource Group Name                                  resgrp1
  Node Relationship                                    cascading
  Participating Node Names                             risc1 risc3

  Service IP label                                    [risc1_svc]           +
  Filesystems                                         [/fs1]                +
  Filesystems Consistency Check                        fsck                 +
  Filesystems Recovery Method                          sequential           +
  Filesystems/Directories to Export                   [/fs1]                +
  Filesystems/Directories to NFS mount                [/nfsfs1;/fs1]        +
  Network For NFS Mount                               [ennetwork]           +
  Volume Groups                                       [extvg]               +
  Concurrent Volume groups                            []                    +
  Raw Disk PVIDs                                      []                    +
  Raw Disk PVIDs                                      []                    +
  Connections Services                                []                    +
  Fast Connect Services                               []                    +
  Application Servers                                 [imageappsrvr]        +
  Highly Available Communication Links                []                    +
  Miscellaneous Data                                  []

  Inactive Takeover Activated                          false                +
  Cascading Without Fallback Enabled                   false                +
  9333 Disk Fencing Activated                          false                +
  SSA Disk Fencing Activated                           false                +
  Filesystems mounted before IP configured             true                 +
[BOTTOM]

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
```

Figure 195. Network for NFS mount

8.6 Improved cluster verification

In HACMP 4.4, the `clverify` command performs the following checks for the NFS configuration:

- IP address takeover must be configured.
- The network specified to perform the NFS mounts must be an IP network and must have a service IP label.
- Verify the existence of the alternate exports file on all cluster nodes.

- Verify that all filesystems and directories for NFS export are present in the alternate exports file.

Chapter 9. Upgrading/Migrating to HACMP 4.4

This chapter describes upgrade and migration path to HACMP 4.4 and operation of new conversion utilities. This helps you to design a plan for upgrading or migrating your current HACMP to HACMP 4.4.

The terms upgrade and migrate can, to some extent, be used interchangeably. For the purpose of this chapter, the term *upgrade* will be used when talking about a move from one release or version to a different release or version of the same feature (HAS, ES, or HANFS), for example, HAS 4.3.1 to HAS 4.4 is an upgrade. However, the move from one release or version to a different feature is a *migration*; HANFS 4.3.1 to HAS 4.4 is a migration.

9.1 Supported upgrade/migration paths

This section focuses on upgrade or migration paths from your current version to HACMP 4.4. There are two ways for upgrading or migration. One is node-by-node upgrade/migration. Node-by-node allows you to migrate to HACMP 4.4 without bringing the entire cluster offline at once. The other way uses a snapshot.

The conversion utilities supplied with HACMP 4.4 contain `cl_convert` and `clconvert_snapshot`. The `cl_convert` utility runs automatically during the *Install with Overwrite* procedure. You need not change ODM from previous version to HACMP 4.4 version by yourself. The node-by-node migration process handles a mixed versions cluster appropriately. Alternatively, you can use the `clconvert_snapshot` utility from the command line to convert a snapshot. `clconvert_snapshot` only rebuilds a snapshot file. In order to change the ODM, you must apply the snapshot to your HACMP cluster from the SMIT menu. Because the cluster synchronization process runs as a part of applying the snapshot, you should issue `clconvert_snapshot`, after the HACMP 4.4 software installation on all nodes in the cluster. Synchronization of a mixed versions cluster may have unexpected results.

You can also simply remove current HACMP, install HACMP 4.4, and reconfigure the cluster according to the cluster worksheets from SMIT menu. This is about the same as doing a new installation.

This section focuses on node-by-node upgrade/migration. For upgrade or migration using snapshot file, refer to Section 9.2.5, “Conversion using a snapshot” on page 260.

Table 6 shows conversion patterns supported by the utilities supplied in HACMP 4.4.

Table 6. Supported conversions

From / To	HAS 4.4	ES 4.4
HAS 4.2.2	Yes	No
HAS 4.3.1	Yes	No
HANFS 4.3.1	Yes	No
HAS 4.4	N/A	Yes[a]
ES 4.2.2	No	Yes
ES 4.3.1	No	Yes

a. See Section 9.1.1, “HAS 4.4 to ES 4.4” on page 246

Note

Node-by-node migration from HAS 4.4 to ES 4.4 requires APAR IY11438 to be installed.

If you wish to convert to HACMP 4.4 from versions earlier than those listed in Table 6 on page 244, you must first upgrade or migrate to one of the supported versions. For example, to upgrade from HACMP 4.2.1 to HACMP 4.4 you must first upgrade to HACMP 4.2.2. You will then be able to upgrade to HACMP 4.4.

Version compatibility was introduced with HACMP 4.2 to allow node-by-node upgrade or migration to a later release of HACMP. During the upgrade or migration, the cluster will consist of nodes at two different levels of HACMP. Version compatibility supports this cluster state. The rest of the cluster nodes are upgraded or migrated one at a time until all nodes in the cluster are running the same level of HACMP. The concept of version compatibility also applies in an ES environment. Table 7 on page 245 shows supported node-by-node upgrade or migration paths.

Table 7. Supported node-by-node upgrade or migration paths

From/To	HAS 4.2.2	HAS 4.3.1	ES 4.2.2	ES 4.3.1
HAS 4.1.0	Yes	Yes	Yes[a]	Yes
HAS 4.1.1	Yes	Yes	Yes[Table a]	Yes
HAS 4.2.0	Yes[b]	Yes	Yes[Table a]	Yes
HAS 4.2.1	Yes[Table b]	Yes	Yes[Table a]	Yes
HAS 4.3.0	No	Yes[Table b]	No	Yes
ES 4.2.1	No	No	Yes[c]	Yes[d]
ES 4.3.0	No	No	No	Yes[e]

a. Must first upgrade to a comparable level of HAS, at which point migration to ES is supported via the install medium. Only available for SP.
b. Upgrade to given level of HAS via PTF.
c. Upgrade to given level of ES via PTF. PSSP 2.3 or later environment only.
d. PSSP 3.1.1 or later environment only.
e. Need to order a media refresh. PSSP 3.1.1 or later environment only.

If your HACMP version is earlier than versions listed in Table 7, you need to remove your HACMP prior to upgrading to HACMP 4.4. An upgrade from HACMP/6000™ Version 1.2, 2.1, or 3.1 to HACMP 4.4 involves reinstalling HACMP on all nodes in the cluster. This means that at some point, the cluster must be brought down.

Note

When upgrading an HACMP cluster, you should not leave the cluster at mixed versions of HACMP for long time. New functionality supplied with HACMP 4.4 are available only when all nodes have been upgraded and the cluster has been synchronized. You cannot synchronize a mixed-version cluster.

Because HACMP 4.4 is supported on AIX 4.3.3 or later, you must upgrade the AIX operating system before upgrading HACMP software.

The *IBM Application Availability Guide*, which is maintained on the Web, includes compatibility details with AIX for AIX software products, support and marketing dates, and information about the latest version or release. It can be found at the following Web site:

http://www.ibm.com/servers/aix/products/ibmsw/list

This Web site is intended as a quick reference guide only. Thorough planning, including reference to the release notes and the software installation guide, should always be undertaken prior to any major software update.

Each Program Temporary Fix (PTF) is associated with a problem description called an Authorized Program Analysis Report (APAR). The APAR database is available and searchable online at the following Web site:

http://techsupport.services.ibm.com/rs6000/fixes

From this page, you can download specific fixes for AIX including HACMP.

9.1.1 HAS 4.4 to ES 4.4

You can migrate from a running HAS 4.4 cluster to a running ES 4.4 cluster without bringing the entire cluster offline at once, thereby keeping all cluster resources available during the migration process.

In order to perform node-by-node migration from HAS to ES;

- All nodes in the cluster must have HAS 4.4 installed and committed.
- You cannot have HAGEO for AIX installed on the cluster.
- HAS supports mesh configuration of non-IP networks, while ES does not. Before migrating from HAS to ES, you may need to redesign their non-IP topology.

For concrete migration steps from HAS 4.4 to ES 4.4, refer to Chapter 14, “Installing the HACMP/ES software“ in *HACMP V4.3 AIX: Enhanced Scalability & Administration Guide, Vol. 1*.

Note

HACMP 4.4 node-by-node migration from HAS to ES requires APAR IY11438 to be installed. Failure to apply this fix will result in a partitioned cluster. The fix must be applied after installation of ES 4.4, but prior to starting cluster services.

9.1.2 HANFS 4.3.1 to HAS 4.4

HAS 4.4 provides HANFS users a node-by-node migration path from HANFS 4.3.1 to HAS 4.4. HACMP 4.4 now supports the NFS export behavior of the HANFS cluster (See Chapter 8, “HACMP 4.4 and NFS” on page 231).

Limitations of migration are:

- In an HANFS cluster, only two nodes are supported.
- Changes to the cluster topology or configuration cannot be made once you have started the migration process. You can make sure any necessary changes after both nodes have finished the migration process.
- The following features are not supported during migration:
  - Event Emulation
  - C-SPOC commands
  - Lock Manager
- Direct migration from HANFS to ES is not supported.

For concrete migration steps from HANFS 4.3.1 to HAS 4.4, refer to Chapter 24, “Upgrading an HACMP Cluster“ in *HACMP V4.3 AIX: Install Guide*.

9.1.3 HAView

HAView has not been changed in this release, and is still at version 4.3.1.0. Migration will not change HAView.

9.2 Conversion utilities

Conversion utilities of HACMP 4.4 provide infrastructure for converting between versions of HACMP. Likewise, conversion between HACMP features (HANFS to HAS and HAS to ES) is streamlined.

This release replaces the previous `cl_convert` and `clconvert_snapshot` utilities with entirely new utilities. However, the end-user functionality is not changed. These changes to the utilities are internal. The internal changes allow utilities to be easily customized for each new release of HACMP for greater reliability of conversions.

This section describes operations of the `cl_convert` and `clconvert_snapshot`, including a description of the structure of these utilities.

9.2.1 cl_convert

The `cl_convert` utility is located in directory /usr/sbin/cluster/conversion. This utility runs as part of the *Install with Overwrite* procedure, and automatically updates the HACMP ODM object classes to the HACMP 4.4 version.

The conversion takes place in stages, to provide a recovery path in case of failure. Conversion procedure during installation of HACMP 4.4 is described as following:

1. Figure 196 shows the initial condition. The ODMDIR environment variable is set to /etc/objrepos.

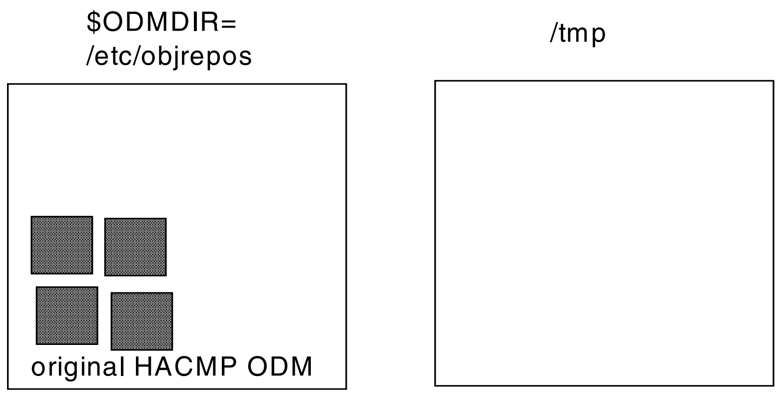

Figure 196. Before conversion

2. Utilities copy ODM classes to a staging area in /tmp, and also to HACMP*.old in /etc/objrepos as shown in Figure 197 on page 248. If conversion process fails, the HACMP*.old files will be used for backout.

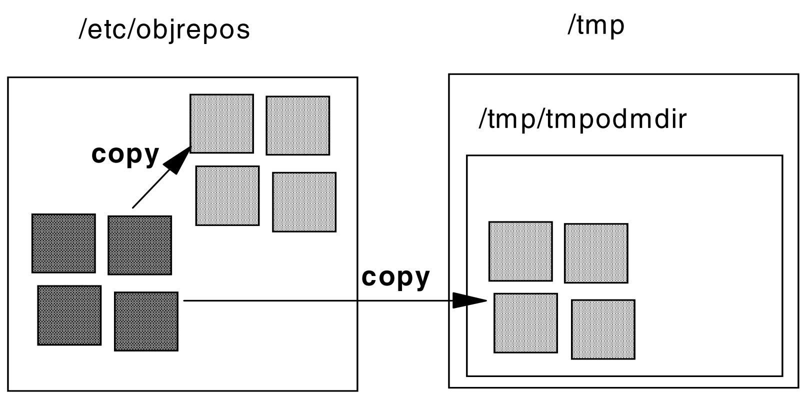

Figure 197. Copy ODM files to the directory /tmp/tmpodmdir

3. After checking that the HACMP*.old files exist, conversions are applied in the staging directory as shown in Figure 198 on page 249.

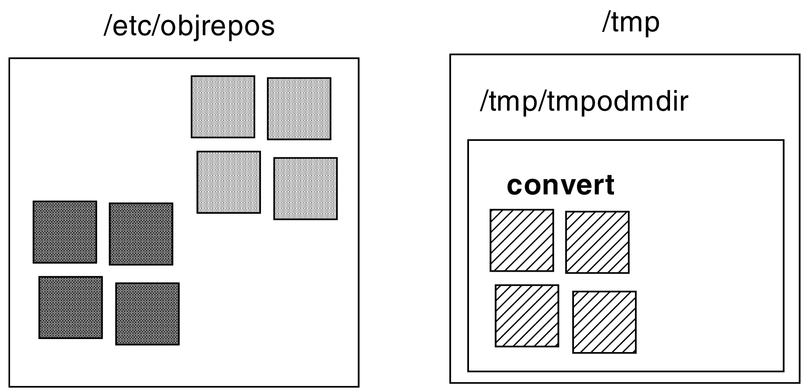

Figure 198. Conversions are applied in the /tmp/tmpodmdir directory

4. When the conversions are complete, the ODM files are copied from the staging area back into /etc/objrepos as shown in Figure 199.

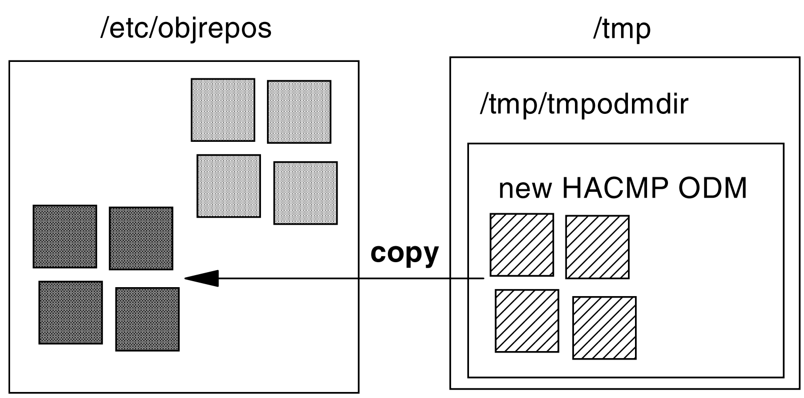

Figure 199. Copy back ODM files to /etc/objrepos

5. When all the ODM files are copied back to /etc/objrepos, the HACMP*.old files are removed as shown in Figure 200 on page 250.

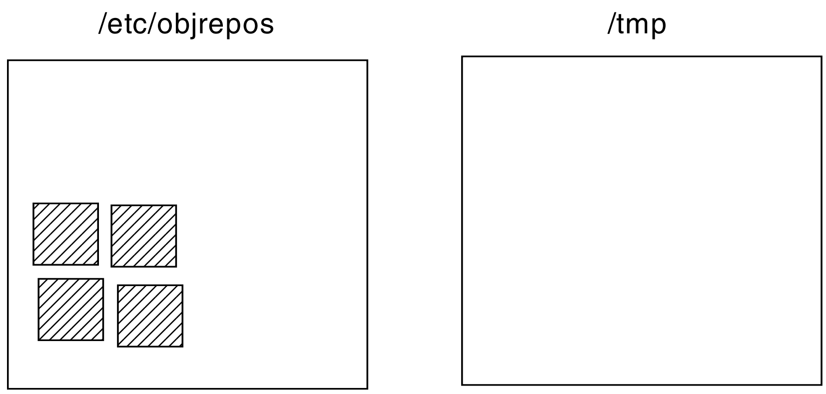

Figure 200. Delete temporary files and directory

6. /tmp/clconvert.log records the status of conversions.

When installation fails, you must run `cl_convert` from the command line. For example, enter:

```
# cl_convert -v 4.3.1 -F
```

You will be able to convert ODM from HACMP 4.3.1 to HACMP 4.4 (assuming that HACMP 4.4 is currently installed on the node). If you need to specify the previously installed product (HAS, ES or HANFS), the `-C`, `-N`, or `-E` flag is required. `-C` flag specifies conversion from HAS, `-N` flag is for HANFS, and `-E` flag is for ES. These flags are mutually exclusive.

Note

The AIX environmental variable $ODMDIR must be set to the directory you wish to convert. If you do not know your previous version, do not run this command.

9.2.2 clconvert_snapshot

The `clconvert_snapshot` utility is also located in directory /usr/sbin/cluster/conversion. You can run `clconvert_snapshot` to upgrade cluster snapshots from previous versions of HACMP to the most recent version of HAS or ES. The `clconvert_snapshot` utility is not run automatically during installation, and must always be run from the command line. See the following examples.

Example 1: Conversion from HAS 4.3.1 to HAS 4.4

The following is the command example:

```
# clconvert_snapshot -v 4.3.1 -s mysnapshot.odm
```

This example is applicable if HAS 4.4 is currently installed on the node where `clconvert_snapshot` will be run and *mysnapshot.odm* is an HAS 4.3.1 snapshot ODM data file. In this example, `clconvert_snapshot` will look for *mysnapshot.odm* in the directory specified by the $SNAPSHOTPATH environmental variable. If a $SNAPSHOTPATH environmental variable is not provided, `clconvert_snapshot` will look in /usr/sbin/cluster/snapshots.

Example 2: Conversion from HAS 4.4 to ES 4.4

The following is the command example:

```
clconvert_snapshot -C -v 4.4 -s /tmp/mysnapshot
```

This example is applicable if ES 4.4 is currently installed on the node where `clconvert_snapshot` will be run and *mysnapshot.odm* is an HAS 4.4 snapshot ODM data file. In this example, the .odm extension was not specified, therefore, `clconvert_snapshot` will look for a snapshot ODM data file called /tmp/mysnapshot.odm.

If you run `clconvert_snapshot` on one node in the cluster, `clconvert_snapshot` will call `cl_convert` with `-i` and `-F` flags. And when `clconvert_snapshot` completes, *mysnapshot.odm* will be upgraded to a target version snapshot ODM data file. Also, a new file *mysnapshot.odm.old*, will be created. This is a copy of the original snapshot ODM data file.

`clconvert_snapshot` executes only upgrade cluster snapshot ODM data file. In order to apply the snapshot ODM data file converted to newer version, you must run the `clsnapshot` utility using SMIT menu on the node where the snapshot ODM data file has been upgraded.

Conversion procedure of `clconvert_snapshot` is described as follows:

1. Figure 201 on page 252 shows an initial condition. The SNAPSHOTPATH environmental variable is set to the directory that contains original snapshot.

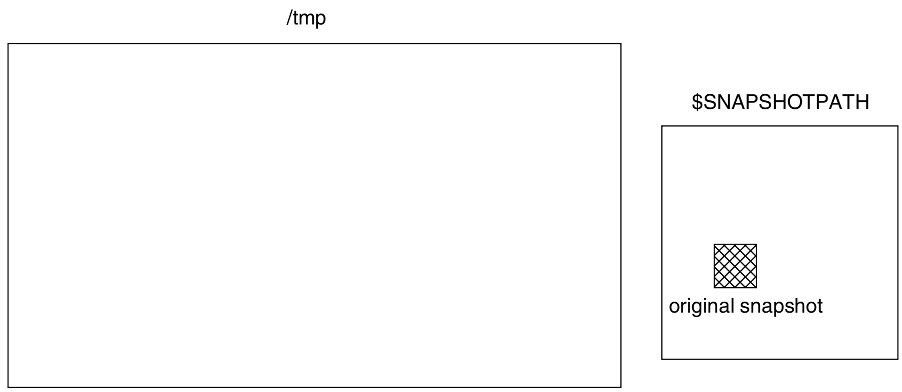

Figure 201. Before converting

2. `clconvert_snapshot` creates the directory /tmp/tmpsnapshotdir as temporary ODM directory, and extracts snapshot file in the directory specified by the $SNAPSHOTPATH environmental variable. `clconvert_snapshot` sets $ODMDIR to /tmp/tmpsnapshotdir (See Figure 202).

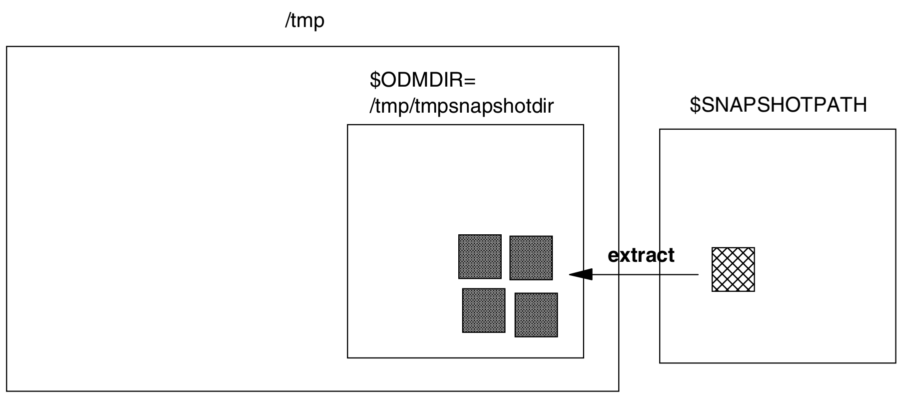

Figure 202. Extract snapshot to the directory /tmp/tmpsnapshotdir

3. `clconvert_snapshot` calls `cl_convert`. `cl_convert` copies ODM classes from /tmp/tmpsnapshotdir to /tmp/tmpodmdir as shown in Figure 203 on page 253.

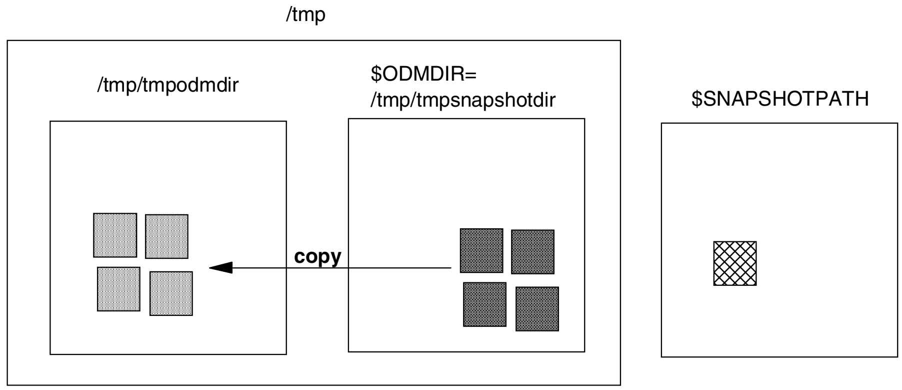

Figure 203. Copy extracted ODM file to the directory /tmp/tmpodmdir

4. Conversions are applied in the directory /tmp/tmpodmdir as shown in Figure 204.

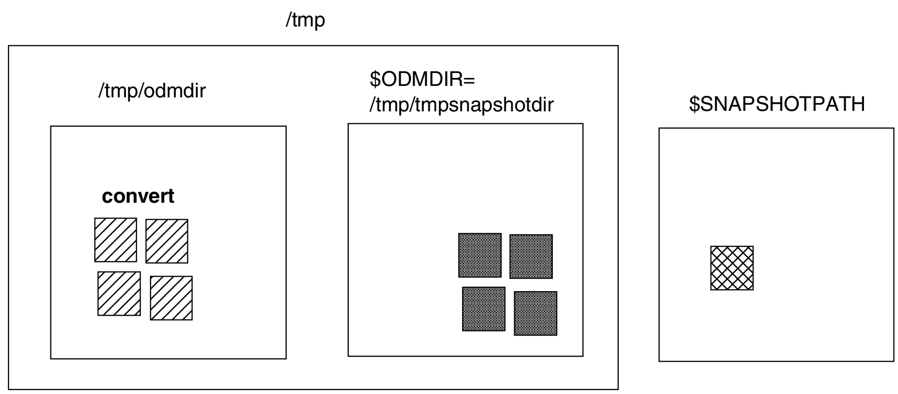

Figure 204. Conversions are applied in the /tmp/tmpodmdir directory

5. When the conversions are complete, the ODM files are copied back into /tmp/tmpsnapshotdir as shown in Figure 205 on page 254.

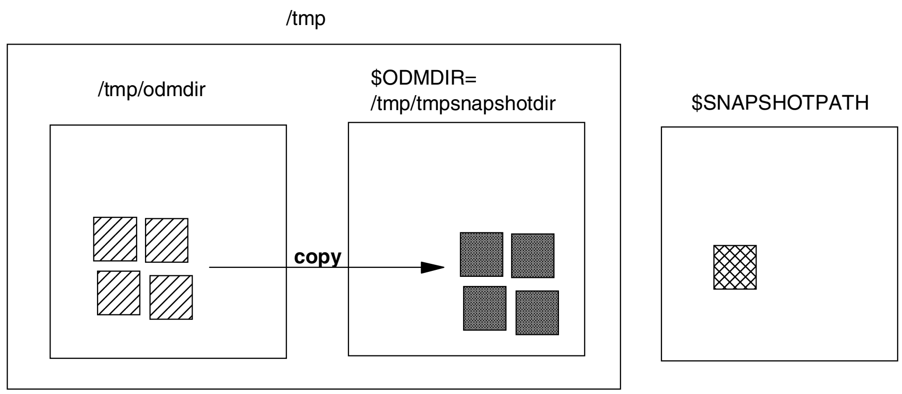

Figure 205. Copy back ODM files to /etc/objrepos

6. When all the ODM files are copied back to /tmp/tmpsnapshotdir, the original snapshot file is moved to *.old, and the new snapshot file is rebuilt from converted ODM files as shown in Figure 206.

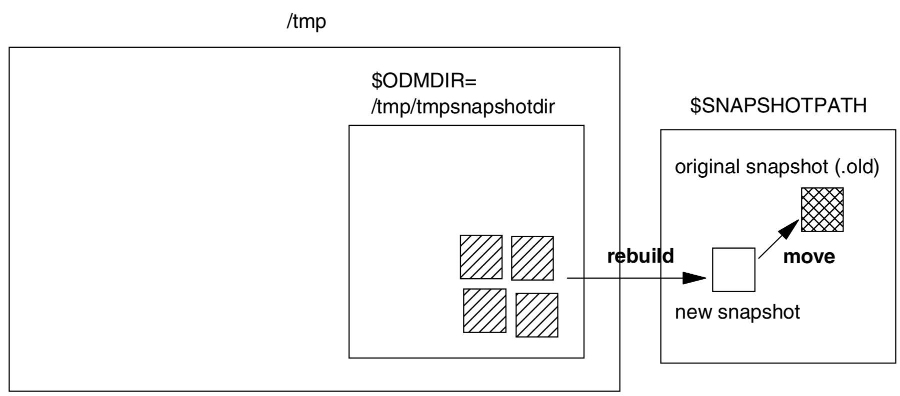

Figure 206. Rebuild new snapshot

7. When the new snapshot file is rebuilt, the temporary ODM directory is removed and $ODMDIR is set to original ODM directory as shown in Figure 207 on page 255.

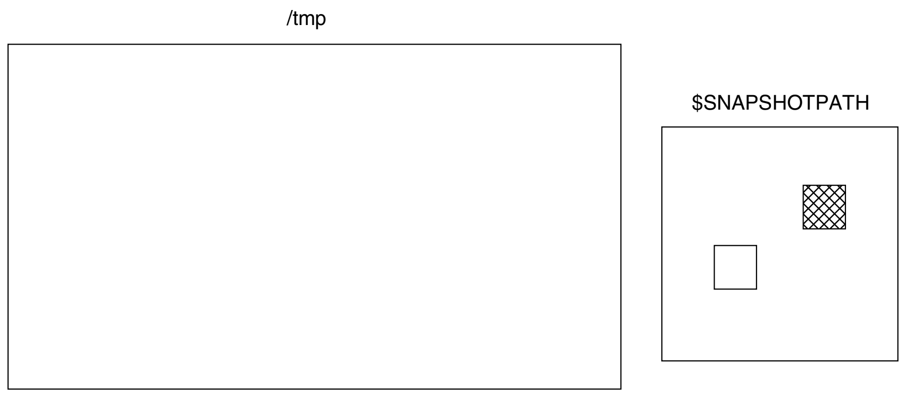

Figure 207. Delete temporary files and directory

8. /tmp/clconvert.log records the status of conversions.

9.2.3 Conversion log

You can make sure whether conversion has completed or not in a log file. /tmp/clconvert.log records the status of conversions. For example, Figure 208 shows the conversion log file written by the clconvert_snapshot.

```
arthur# more /tmp/clconvert.log

--------- log file for clconvert_snapshot: Wed Jul 19 11:20:31 EDT 2000

Command line is:
clconvert_snapshot -C -v 4.4 -s has44July19th

Parameters read in from command line are:
    Source Product is HACMP.
    Source Version is 4.4.0.
    Target Product is HAES.
    Target Version is 4.4.0.1.
    Snapshot File Flag is set: /usr/es/sbin/cluster/snapshots/has44July19th.odm

Setup:
    Create temporary directory: /tmp/tmpsnapshotdir
    Original directory: /etc/objrepos
    Changing ODMDIR to /tmp/tmpsnapshotdir.

Initiating extraction of snapshot file /usr/es/sbin/cluster/snapshots/has44July
19th.odm to /tmp/tmpsnapshotdir.
    Copying HA* from /etc/objrepos to /tmp/tmpsnapshotdir.
clconvert.log (3%)
```

Figure 208. /tmp/clconvert.log

This log file is regenerated each time `cl_convert` or `clconvert_snapshot` is executed. Therefore, if you need to remain previous conversion log, you must move or copy the log file to another directory before launching the next conversion.

For more information on `cl_convert` and `clconvert_snapshot`, refer to the respective manual pages, or to Appendix A, "HACMP for AIX Commands" in *HACMP V4.3 AIX: Administration Guide*.

9.2.4 Customizing conversions

In HACMP 4.4, conversion utilities provide easy conversion between the HACMP versions and products listed in Table 6 on page 244. In the case of upgrading from a non-supported version, you cannot use the conversion utilities, but you can customize the conversion scripts to convert ODM from earlier version to HACMP 4.4. However, we recommend you manually configure the cluster in that case.

Figure 209 illustrates the structure of conversion scripts. All conversion scripts are located in directory /usr/sbin/cluster/conversion.

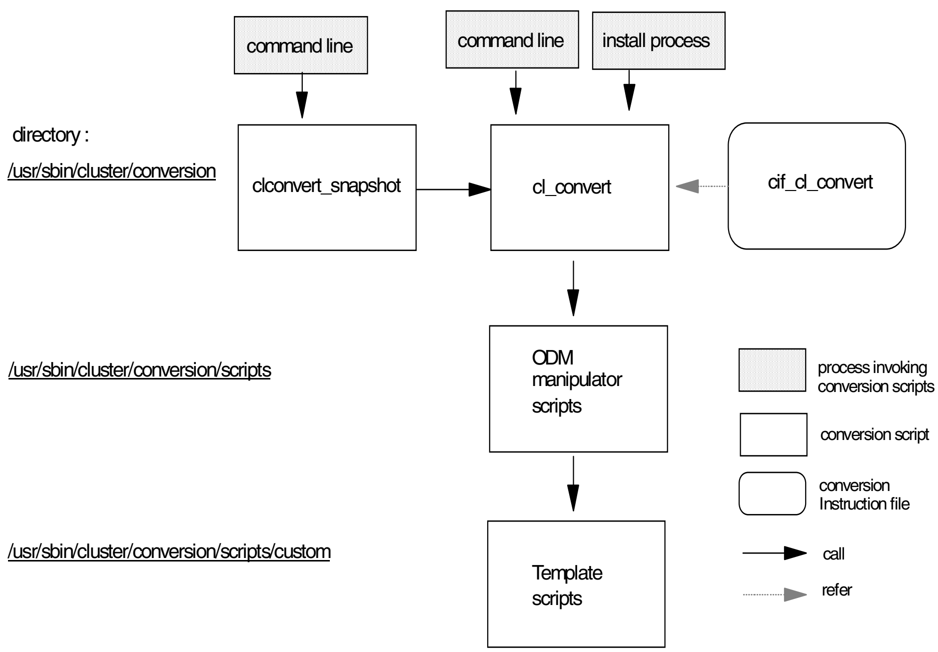

Figure 209. Structure of conversion utilities

`cl_convert` reads /usr/sbin/cluster/conversion/cif_cl_convert, the conversion instruction file, to get information about conversion paths. The file is shown in Figure 210.

```
arthur# cat /usr/sbin/cluster/conversion/cif_cl_convert
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
#
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1999,2000
# 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
#
# Note:  The all-capital-letters, spacing, colons, and semicolons are
# necessary for the program to run correctly.

VER = 4.4.0
HACMP:4.2.2:HACMP:4.4.0;HACMP422toHACMP44
HACMP:4.3.1:HACMP:4.4.0;HACMP431toHACMP44
HAES:4.2.2:HAES:4.4.0;HAES422toHAES44
HAES:4.3.1:HAES:4.4.0;HAES431toHAES44
HACMP:4.4.0:HACMP:4.4.0;NULL
HAES:4.4.0:HAES:4.4.0;NULL
HACMP:4.4.0:HAES:4.4.0;HACMP44toHAES44
HANFS:4.3.1:HACMP:4.4.0;HANFS431toHACMP44
HACMP:4.2.2:HAES:4.4.0;HACMP422toHACMP44,HACMP44toHAES44
HACMP:4.3.1:HAES:4.4.0;HACMP431toHACMP44,HACMP44toHAES44
arthur#
```

Figure 210. /usr/sbin/cluster/conversion/cif_cl_convert

In HACMP 4.4, conversion utilities can perform 10 patterns of conversions listed in cif_cl_convert. If your migration or upgrade pattern is not included in this file, you cannot use the conversion utilities. Figure 211 on page 258 shows the cif_cl_convert file format. You can add the new entry in cif_cl_convert, or create your own cif_cl_convert.

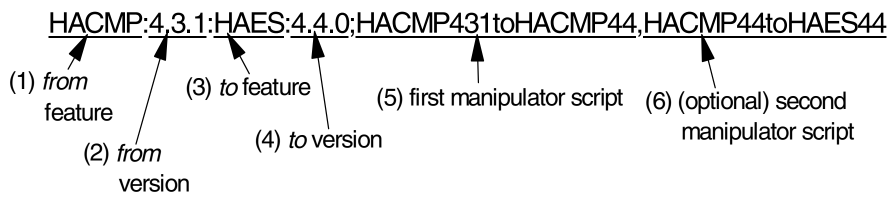

Figure 211. cif_cl_convert file format

When `cl_convert` runs automatically during the installation procedure, all necessary parameters are set by the installation process. First, `cl_convert` analyzes the parameters (first, second, third, and fourth column, separated by colon and semicolon). Next, it calls corresponding manipulate shell scripts (fifth and sixth column, separated by comma). In the case of Figure 211, in order to convert ODM files from HAS 4.3.1 to ES 4.4, `cl_convert` executes two conversion shell scripts, `HACMP431toHACMP44` and `HACMP44toHAES44`.These scripts described in cif_cl_convert are located in the directory /usr/sbin/cluster/conversion/scripts as shown in Figure 212. They adjust the particular version ODM to fit the target version ODM.

```
arthur# ls -l /usr/sbin/cluster/conversion/scripts
total 144
-r-x------   1 root     system      5278 May 10 11:24 HACMP422toHACMP431
-r-x------   1 root     system      6409 May 10 11:24 HACMP422toHACMP44
-r-x------   1 root     system      1193 May 10 11:24 HACMP431toHACMP44
-r-x------   1 root     system      6622 May 10 11:24 HACMP431toHAES431
-r-x------   1 root     system      7491 May 10 11:24 HACMP44toHAES44
-r-x------   1 root     system      6185 May 10 11:24 HAES422toHAES431
-r-x------   1 root     system     15438 May 10 11:24 HAES422toHAES44
-r-x------   1 root     system      5962 May 10 11:24 HAES431toHAES44
drwxr-xr-x   2 root     system       512 Jul 19 10:33 custom
arthur#
```

Figure 212. List of ODM manipulator scripts

Depending on migration path, some shell scripts located in the directory /usr/sbin/cluster/conversion/scripts/custom are called by manipulator scripts as shown in Figure 213 on page 259.

```
arthur# ls -l /usr/sbin/cluster/conversion/scripts/custom
total 256
-r-x------   1 root     system      6650 May 10 20:00 AddResourceStanza
-r-x------   1 root     system      7481 May 10 20:00 AddResourceStanzaNFS
-r-x------   1 root     system     10197 May 10 20:00 AssignInterface
-r-x------   1 root     system      5721 May 10 20:00 ConvertEventPaths
-r-x------   1 root     system      5988 May 10 20:00 ConvertLogPathsES
-r-x------   1 root     system      5834 May 10 20:00 GetHighestNetwork
-r-x------   1 root     system      6072 May 10 20:00 GetHighestNode
-r-x------   1 root     system      6892 May 10 20:00 GetInstanceNum
-r-x------   1 root     system      1347 May 10 20:00 GetNewCommands
-r-x------   1 root     system      7010 May 10 20:00 GetNodeIdList
-r-x------   1 root     system     10527 May 10 20:00 GetNodeIdandHandle
-r-x------   1 root     system      5987 May 10 20:00 GetSecurityMode
-r-x------   1 root     system      5697 May 10 20:00 SetNimCycle
-r-x------   1 root     system      8260 May 10 20:00 UpdateEventtoCustomScript
-rwxr-xr-x   1 root     system      5970 May 10 20:00 VerifyLogValue
arthur#
```

Figure 213. List of template scripts

These scripts are written by Perl. If you have adequate knowledge of ODM and Perl, you may be able to modify the scripts according to your own environment. You can see brief outlines of template scripts as follows:

- `AddResourceStanza` adds CASCADE_WO_FALLBACK stanza in HACMPresource ODM class.
- `AddResourceStanzaNFS` adds NFS stanza in HACMPresource ODM class.
- `AssignInterface` assigns the interfacename for the HACMPadapter ODM class.
- `ConvertEventPaths` changes path related to ES cluster events from /usr/sbin/... to /usr/es/sbin/...
- `ConvertLogPathsES` changes path related to ES log files from /usr/sbin/... to /usr/es/sbin/...
- `GetHighestNetwork` gets the highest network_id to put in HACMPcluster ODM class.
- `GetHighestNode` gets the highest node_id to put in HACMPcluster ODM class.
- `GetInstanceNum` gets instanceNum to put in HACMPtopsvcs ODM class.
- `GetNewCommands` outputs just an empty strings, if it is called from `clconvert_snapshot`. Otherwise it outputs the installed HACMPcommand ODM class for an upgrade.

- `GetNodeIdandHandle` gets node_id and node_handle to put in HACMPnode ODM class, and last_node_ids, highest_node_id, and node_handle, to input HACMPcluster ODM class. It is used for SP nodes.
- `GetNodeIdList` makes a list of unique node id in HACMPnode ODM class and puts it in last_node_ids in HACMPcluster ODM class. Also, it gets security level and puts in HACMPcluster ODM class.
- `GetSecurityMode` gets security level and puts in HACMPcluster ODM class.
- `SetNimCycle` sets cycle to “4”, if HACMP feature is ES and NIM is not “atm”.
- `UpdateEventtoCustomScript` updates HACMP event scripts to customized scripts.
- `VerifyLogValue` verifies value in HACMPlogs ODM class. If not appropriate, it sets the default value.

9.2.5 Conversion using a snapshot

Using the conversion utilities, you can perform migration or upgrade to HACMP 4.4.

To do this, use the following steps:

1. Create a snapshot to save the current HACMP cluster configuration on one node in the cluster.
2. Stop the cluster services on all the nodes in the cluster gracefully.
3. Remove the HACMP software. To do this operation by SMIT, enter:

   ```
   # smit install_remove
   ```

   Modify the fields in the SMIT menu as follows:

```
                         Remove Installed Software

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

                                                        [Entry Fields]
* SOFTWARE name                                      [cluster*]            +
  PREVIEW only? (remove operation will NOT occur)     no                   +
  REMOVE dependent software?                          no                   +
  EXTEND file systems if space needed?                yes                  +
  DETAILED output?                                    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
```

4. Install HACMP 4.4.
5. Repeat steps 3 through 4 for all other nodes in the cluster.
6. Convert the snapshot you created in step 1 to HACMP 4.4 version. Use `clconvert_snapshot` on one node where you created snapshot. See Section 9.2.2, “clconvert_snapshot” on page 250.
7. If you have modified /etc/inittab and /etc/rc.net, check to ensure that these files exist appropriately as specified in Appendix E, *HACMP V4.3 AIX: Install Guide*, before rebooting the node.
8. Reboot the nodes after installing HACMP 4.4 on all the cluster nodes.
9. To apply the snapshot converted to HACMP 4.4 version, enter:

   ```
   # smit cm_apply_snap.select
   ```

   Select the snapshot name converted in Step 6 as shown in Figure 214 on page 262.

```
                              Apply a Cluster Snapshot

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

                                                        [Entry Fields]
  Cluster Snapshot Name                                 mysnapshot
  Cluster Snapshot Description                          This -- is a HACMP sna>
  Un/Configure Cluster Resources?                       [Yes]                  +
  Force apply if verify fails?                          [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
```

Figure 214. Apply a Cluster Snapshot SMIT menu

10. Verify the cluster topology on all nodes using the following utilities:

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

    or

    `smit clverify.dialog`

11. Start the HACMP software using the `smit clstart` fastpath. And verify that the node successfully joins the cluster. When the node joins the cluster without failure, you can find the following message in /tmp/hacmp.out:

    `Jul 26 09:16:58 EVENT COMPLETED: node_up_complete arthur`

9.3 Considerations about upgrade and migration

In this section, we mention basic considerations of upgrade and migration. Before you begin to upgrade or migrate to HACMP 4.4, you should read through this section for reliable planning.

9.3.1 Preparation

Preparation of upgrade or migration is one of the most important administrative tasks.

The following is the check points for preparation:

- It is recommended to perform a full system backup (mksysb).

- If any nodes in the cluster are currently set to start cluster services automatically on reboot, change this setting before beginning the migration process. If the HACMP cluster services entry exists in the /etc/inittab file, you must comment out or remove this entry.
- Save the HACMP cluster configuration in a snapshot. You can create a snapshot file using the `smit cm_add_snap.dialog` fastpath as shown in Figure 215.

```
                              Add a Cluster Snapshot

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

                                                        [Entry Fields]
* Cluster Snapshot Name                              [mysnapshot]              /
  Custom Defined Snapshot Methods                    []                       +
* Cluster Snapshot Description                       [This is a HACMP snapsh>










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
```

Figure 215. Add a Cluster Snapshot SMIT menu

The snapshot file will be created in directory /usr/sbin/cluster/snapshots as shown in Figure 216 on page 264. For more information, see Chapter 1, “Saving and Restoring Cluster Configurations” *HACMP V4.3 AIX: Administration Guide*.

```
arthur# ls -l /usr/sbin/cluster/snapshots/
total 15688
-r--r--r--   1 root     system     19588 May 10 11:22 Configuration1.odm
-r--r--r--   1 root     system     21216 May 10 11:22 Configuration2.odm
-r--r--r--   1 root     system     19269 May 10 11:22 Configuration3.odm
-r--r--r--   1 root     system     21217 May 10 11:22 Configuration4.odm
-r--r--r--   1 root     system     21348 May 10 11:23 Configuration5.odm
-rw-r--r--   1 root     system     24195 Jul 21 17:03 active.0.odm
-rw-r--r--   1 root     system   7868216 Jul 26 15:07 mysnapshot.info
-rw-r--r--   1 root     system     24224 Jul 26 15:06 mysnapshot.odm
arthur#
```

Figure 216. Snapshot files in /usr/sbin/cluster/snapshots

Note

Because the directories /usr/sbin/cluster, /usr/es/sbin/cluster, and /usr/sbin/lpp/cluster are deleted and recreated during the installation of HACMP, move the snapshot files in these directories to another.

- Save customized event scripts. During the installation of HACMP software, any event scripts in /usr/[es]/sbin/cluster are overwritten.

9.3.2 During the upgrade and migration

This section identifies problems that you may encounter while you install, update, or migrate the HACMP, and offers possible solutions.

9.3.2.1 Unsuccessful installation

If you experience problems during the installation, the installation program automatically performs a cleanup process. If, for some reason, the cleanup is not performed after an unsuccessful installation:

1. Enter the `smit maintain_software` fastpath.
2. Select **Clean Up After Failed or Interrupted Installation**.
3. Review the SMIT output (or examine the /smit.log file) for the interruption's cause.
4. Fix any problems and repeat the installation process.

9.3.2.2 cl_convert does not run due to failed installation

When you install HACMP, `cl_convert` runs automatically as a part of installation. However, if HACMP LPP installation fails, `cl_convert` cannot run as a result. Therefore, conversion from the ODM of a previous HACMP version to the ODM of the current version will also fail. If you find the following error message in /tmp/clconvert.log:

```
"Exiting with error code 1. Errors encountered."
```

you must run `cl_convert` from the command line to convert ODM (See Section 9.2.1, "cl_convert" on page 247).

> **Note**
>
> Before converting from HAS 4.4 to ES 4.4, be sure that your ODMDIR environment variable is set to /etc/es/objrepos.

9.3.2.3 clverify gives warning message

If you get the following message in `clverify`, even though you have not configured *Auto Error Notification*;

```
"Remember to redo automatic error notification if configuration has
changed."
```

ignore this message.

9.3.2.4 config_too_long message appears

When the migrating or upgrading process has completed, you may see the message "config_too_long" in the HACMP log file.

This message appears when the cluster manager detects that an event has been processing for more than the six minutes allowed by default. "config_too_long" messages will continue to be appended to the hacmp.out log every 30 seconds until the event completes. If you observe these messages, you should periodically check that the event is indeed still running and has not failed.

You can avoid these messages by increasing the time to wait before calling `config_too_long`, using the following command:

```
# chssys -s clstrmgr -a "-u miliseconds_to_wait"
```

For example:

```
# chssys -s clstrmgr -a "-u 60000"
```

This sets the time to 600 seconds (10 minutes), instead of the default six minutes.

Note

If you do change the time, you should change it back to the default time when the migration or upgrade is complete.

9.3.2.5 Connection to remote host refused

During synchronizing or verifying the cluster, you may see the following message on the SMIT menu or in the smit.log:

```
"Connection to remote host refused"
```

In order to verify and synchronize the configuration, you must verify the configuration related to TCP/IP communication.

1. Verify that the .rhosts file is configured correctly. Usually, TCP/IP problems result from improper format of the .rhosts file or from inaccurate name resolution.

   The proper format of the .rhosts can be one or more of the following:

   ```
   ip_label root
   ip_label.fully.qualified.name root
   ip_address root
   ```

2. Verify that name resolution is configured and functioning properly. Every HACMP IP label should be defined in both the /etc/hosts file and any other source of name resolution that you may use in Domain Name services (DNS) or Network Information System (NIS). Use the following HACMP command to see the IP addresses and their corresponding IP labels that are defined to HACMP:

   ```
   # cd /usr/sbin/cluster/utilities
   # cllsif -Sc | grep -v tty | awk -F: '{ print $7 "    " $1 }'
   ```

   Then make sure that both the /etc/hosts file and other name services are consistent with the listing given by the command.

3. Verify the HACMP ports and SNMP are properly defined. Use the following list:

   - The /etc/services file should have the following entries:

   ```
   clinfo_deadman    6176/tcp
   clm_keepalive     6255/udp
   cllockd           6100/udp
   clm_pts           6200/tcp
   clsmuxpd          6270/tcp
   clm_lkm           6150/tcp
   clm_smux          6175/tcp
   ```

```
godm              6177/tcp
```

- The /etc/snmpd.conf file should have the following entry:

```
smux 1.3.6.1.4.1.2.3.1.2.1.5 clsmuxpd_password # HACMP for
AIXclsmuxpd
```

- The /etc/snmpd.peers file should have the following entry:

```
clsmuxpd 1.3.6.1.4.1.2.3.1.2.1.5 "clsmuxpd_password" # HACMP for AIX
clsmuxpd
```

- The /etc/inetd.conf file should have the following entry:

```
godm stream tcp nowait root /usr/[es]/sbin/cluster/godmd
```

4. Make sure that there are no duplicate IP addresses on the same network.
5. Verify that /etc/hosts has the IP Label defined to HACMP as the last entry or alias for a given IP address.

In addition, you must check the status of GODM. If you can not find GODM entry in the output of the `netstat -a` command as follows, reboot the machine or restart inetd:

```
arthur# netstat -a | grep *.godm
tcp4  0     0  *.godm           *.*                LISTEN
arthur#
```

Note

If you have changed root user's home directory from default (/) to another, the .rhosts file must be located in both the directory / and root user's home directory.

9.3.2.6 Backout of migration from HAS to ES

If you decide not to complete the migration process from HAS to ES, you can uninstall the ES on the nodes where you have installed it at any point in the process before starting the cluster services on the last node. To do this, use the following steps:

1. On each node in turn (one at a time), stop cluster services using the `clstop -gr` command or the `smit clstop` fastpath.

   Check that the cluster services are stopped on the node and that its cluster resources have been transferred to takeover nodes before proceeding.

2. When you are sure the resources on the node have been properly transferred to a takeover node, remove the ES using the `smit install_remove` fastpath.

Note

Be sure *not* to remove the manual pages or the C-SPOC messages software; these are shared with the HAS.

3. Start the cluster services on this node. When you are certain the resources have transferred properly (if necessary) back to this node, repeat these steps on the next node.
4. Continue this process until ES has been removed from all nodes in the cluster.

9.3.2.7 Backout of migration from HANFS to HAS

If you decide not to complete the migration process from HANFS to HAS, you can uninstall the HAS on the nodes where you have installed it at any point in the process before starting the cluster services on the last node. To do this use the following steps:

1. On each node in turn (one at a time), stop cluster services using the `clstop -gr` command or the `smit clstop` fastpath.

   Check that the cluster services are stopped on the node and that its cluster resources have been transferred to takeover nodes before proceeding.
2. When you are sure the resources on the node have been properly transferred to a takeover node, remove the HACMP using the `smit install_remove` fastpath.
3. Reinstall the HANFS 4.3.1 software (and all applicable PTF levels).
4. Synchronize the cluster from the node that is still running HANFS.
5. Start HANFS on this node.

Note

Starting cluster services on the last node is a point of no return. Once you have restarted HACMP on the last node and the migration has commenced, you cannot reverse it. If you wish to return to the previous configuration after this point, you will have to reinstall the HACMP and apply the saved snapshot.

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 configuration

Figure 217 illustrates our hardware configuration.

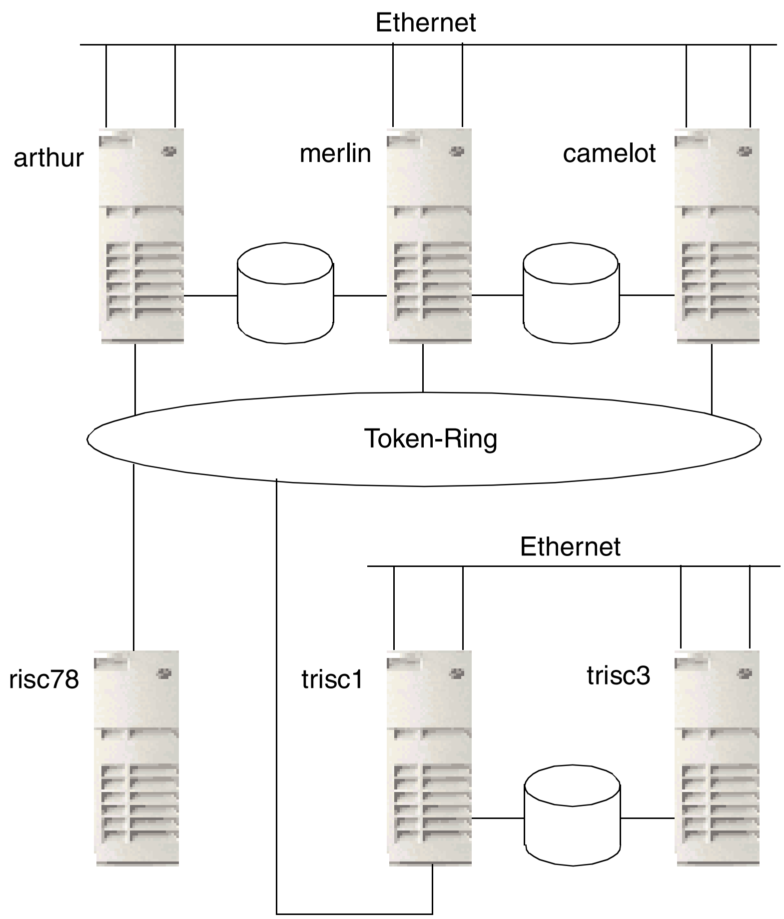

Figure 217. Hardware configuration

A.2 Software configuration

A.2.1 AIX software

We used AIX 4.3.3 with maintenance level 2.

A.2.2 HACMP software

We used HACMP 4.4 with the following PTFs:

- IY10563
- IY10564

A.2.3 Tivoli software

We used the following Tivoli software:

- Framework 3.6.1
- Distributed Monitoring 3.6.1
- AEF 3.6.1

A.2.3.1 Prerequisite software

You need the following software installed to use Tivoli software:

- X11.Dt.compat 4.3.1.0
- X11.compat.adt.Motif12 4.3.3.0
- X11.compat.fnt.pc 4.3.0.0
- X11.compat.lib.Motif10 4.3.0.0
- X11.compat.lib.Motif114 4.3.0.0
- X11.compat.lib.X11R3 4.3.0.0
- X11.compat.lib.X11R4 4.3.0.0
- X11.compat.lib.X11R5 4.3.3.0
- bos.compat.NetInstl 4.3.3.0
- bos.compat.cmds 4.3.3.0
- bos.compat.imk 4.3.0.0
- bos.compat.lan 4.3.3.0
- bos.compat.libs 4.3.0.0
- bos.compat.links 4.3.1.0
- bos.compat.net 4.3.3.0
- bos.compat.termcap 4.3.3.0
- bos.compat.NetInstl 4.3.3.0
- bos.compat.links 4.3.1.0
- bos.compat.net 4.3.3.0
- bos.compat.termcap 4.3.3.0

- bos.compat.termcap.data 4.3.0.0
- Java.adt.docs 1.1.8.0
- Java.adt.includes 1.1.8.0
- Java.adt.src 1.1.8.0
- Java.rmi-iiop.bin 1.1.8.0
- Java.rmi-iiop.docs 1.1.8.0
- Java.rmi-iiop.lib 1.1.8.0
- Java.rmi-iiop.samples 1.1.8.0
- Java.rte.Dt 1.1.8.0
- Java.rte.bin 1.1.8.0
- Java.rte.classes 1.1.8.0
- Java.rte.lib 1.1.8.0
- Java.samples.AIXDemos 1.1.8.0
- Java.samples.demos 1.1.8.0
- Java.samples.examples 1.1.8.0

A.3 Softcopy Manuals

The following publications are supplied on CD-ROM with the basic machine-readable material.

- *IBM High Availability Cluster Multi-Processing for AIX: Concepts and Facilities*, SC23-4276
- *IBM High Availability Cluster Multi-Processing for AIX: Planning Guide*, SC23-4277
- *IBM High Availability Cluster Multi-Processing for AIX: Installation Guide*, SC23-4278
- *IBM High Availability Cluster Multi-Processing for AIX: Administration Guide*, SC23-4279
- *IBM High Availability Cluster Multi-Processing for AIX: Troubleshooting Guide*, SC23-4280
- *IBM High Availability Cluster Multi-Processing for AIX: Programming Client Applications*, SC23-4282
- *IBM High Availability Cluster Multi-Processing for AIX: Programming Locking Applications*, SC23-4281
- *IBM High Availability Cluster Multi-Processing for AIX: Enhanced Scalability Installation and Administration Guide, Vol. 1*, SC23-4284
- *IBM High Availability Cluster Multi-Processing for AIX: Enhanced Scalability Installation and Administration Guide, Vol. 2*, SC23-4306

- *RS/6000 Cluster Technology: Event Management Programming Guide and Reference*, SA22-7354
- *RS/6000 Cluster Technology: Group Service Programming Guide and Reference*, SA22-7355
- *RS/6000 Cluster Technology: First Failure Data Capture Programming Guide and Reference*, SA22-7454

The publications for HACMP 4.4 are included as installation images on the installation media. After installation, the publications may be viewed or printed. All publications are provided in PostScript and PDF format. Selected publications are also available in HTML.

Appendix B. Application server scripts

This appendix shows the start and stop scripts used in the definition of the application server described in Chapter 2, “Application Monitoring” on page 5.

B.1 The start_imagedemo script

```
#!/bin/ksh
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
#
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1990,1999
# 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
# @(#)57  1.7  src/43haes/usr/sbin/cluster/events/utils/start_imagedemo.sh, hacmp.events,
43haes_rmoh, rmoht5dp1 11/29/99 13:16:21

###############################################################################
#
#   COMPONENT_NAME: EVENTUTILS
#
#   FUNCTIONS: none
#
###############################################################################

###############################################################################
#
# Name: start_imagedemo
#
# The following code is used to start the Image Server demo
#
# Arguments: -d directory_containing_images
#      -a Service_Address
#
# Returns:      0 - success or program already running
#               1 - failed (lock manager is not running)
#      2 - usage
#      3 - failed (given image path does not exist)
#
# Environment:  VERBOSE_LOGGING,PATH,IMSERV_IMAGE_LOCATION
###############################################################################

###############################################################################
#
# Usage
#
###############################################################################
usage()
{
    [[ "$VERBOSE_LOGGING" = "high" ]] && set -x
```

```
    cl_echo 306  "Usage: start_imagedemo [ -d directory_containing_images ] [ -a
Service_Address ]\n"
    exit 2
}

################################################################################
#
#   Main Entry Point
#
################################################################################

PROGNAME=$(basename ${0})
PATH="$($(dirname ${0})/../../utilities/cl_get_path all)"
[[ "$VERBOSE_LOGGING" = "high" ]] && set -x
[[ "$VERBOSE_LOGGING" = "high" ]] && version='1.7'
HA_DIR="$(cl_get_path)"

IMAGE_DIR="/usr/${HA_DIR}/sbin/cluster/demos/image"
export PATH=${IMAGE_DIR}:$PATH

SERVICE_ADDRESS=""
IMSERV_IMAGE_LOCATION=""

# Get command line options
set -- $(getopt "d:a:" $*)
if [ $? -ne 0 ]
then
    usage
fi

# Parse command line.
while [ $1 != -- ]
do
    case $1 in
    -d)
        IMSERV_IMAGE_LOCATION="$2"
        shift ; shift
        ;;
    -a)
    IP_LABEL="$2"
    #
    # Convert ip label to dot address
    # Doesn't matter if the address is already in dot format
    #
        SERVICE_ADDRESS=$(host $IP_LABEL | cut -d' ' -f3 | sed s/,//g)
        shift ; shift
        ;;
    *)
        usage
        ;;
    esac
done

shift   # lose the --

if [ "$IMSERV_IMAGE_LOCATION" != "" ]
then
    ls $IMSERV_IMAGE_LOCATION >/dev/null 2>&1
    if [ $? -ne 0 ]
    then
       cl_echo 515 "$PROGNAME: $IMSERV_IMAGE_LOCATION does not exist.\n" \
$PROGNAME $IMSERV_IMAGE_LOCATION
       exit 3
```

```
    fi

    # Export image location.  If not given image location,
    # imserv program will pick up the default path
    export IMSERV_IMAGE_LOCATION
fi

set -u

# Check to see if cllockd is running
ps -e | grep -s cllockd
if [ "$?" -ne 0 ]
then
    cl_echo 50 "$PROGNAME: cllockd must be running for demo" $PROGNAME
    exit 1
fi

#  Check to see if imserv already running
if [ "$SERVICE_ADDRESS" = "" ]
then
    LINE=$(ps -ef | egrep -e "imserv" | grep -v egrep)
else
    LINE=$(ps -ef | egrep -e "imserv" | grep "$SERVICE_ADDRESS" | grep -v egrep)
fi

#  If not, start the server
if [ -z "$LINE" ]
then
    imserv $SERVICE_ADDRESS 2>&1 > /tmp/imserv
else
    cl_echo 51 "$PROGNAME: imserv already running." $PROGNAME
fi

exit 0
```

B.2 The stop_imagedemo script

```
#!/bin/ksh
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
#
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1990,1999
# 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
# @(#)591.5  src/43haes/usr/sbin/cluster/events/utils/stop_imagedemo.sh, hacmp.events,
43haes_rmoh, rmoht5dp1 11/29/99 13:24:45
# $Id: stop_imagedemo.sh,v 8.2.2.1 1996/12/06 21:34:20 suad Exp $
#
#   COMPONENT_NAME: EVENTUTILS
#
#   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: stop_imagedemo
#
# The following code is used to stop the Image Server demo
#
# Arguments: -a Service_Address
#
# Returns:      0 - success or process not found
#      2 - usage
#
# Environment:  VERBOSE_LOGGING, PATH
################################################################################

################################################################################
#
# Usage
#
################################################################################
usage()
{
    [[ "$VERBOSE_LOGGING" = "high" ]] && set -x
    cl_echo 1045  "Usage: stop_imagedemo [ -a Service_Address ]\n"
    exit 2
}

################################################################################
#
#       Main Entry Point
#
################################################################################

PROGNAME=$(basename ${0})
[[ "$VERBOSE_LOGGING" = "high" ]] && set -x
[[ "$VERBOSE_LOGGING" = "high" ]] && version='1.5'

SERVICE_ADDRESS=""

# Get command line options
set -- `getopt "a:" $*`
if [ $? -ne 0 ]
then
    usage
fi

# Parse command line.
while [ $1 != -- ]
do
    case $1 in
    -a)
    IP_LABEL="$2"
        #
        # Convert ip label to dot address
        # Doesn't matter if the address is already in dot format
        #
```

```
        SERVICE_ADDRESS=`host $IP_LABEL | cut -d' ' -f3 | sed s/,//g`
        shift ; shift
        ;;
    *)
        usage
        ;;
    esac
done

shift   # lose the --

set -u

#  Get the pid (with the given Service Address)
if [ "$SERVICE_ADDRESS" = "" ]
then
    PID=`ps auxww | egrep -e "imserv" | grep -v egrep | awk -F' ' '{print $2}'`
else
    PID=`ps auxww | egrep -e "imserv" | grep -v egrep | grep $SERVICE_ADDRESS | awk -F' '
'{print $2}'`
fi

if [ -n "$PID" ]
then
   kill -9 $PID
fi
exit 0
```

Appendix C. Special notices

This publication is intended to help IBM customers, IBM business partners, IBM sales professionals, and IBM I/T specialists wishing to obtain a reference for High Availability Cluster Multi-Processing for AIX Version 4.4.0 Enhancements. The information in this publication is not intended as the specification of any programming interfaces that are provided by High Availability Cluster Multi-Processing for AIX. See the PUBLICATIONS section of the IBM Programming Announcement for High Availability Cluster Multi-Processing for AIX 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, North Castle Drive, Armonk, NY 10504-1785.

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 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.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

IBM ®	AIX
AT	CT
Current	HACMP/6000
Lotus	Netfinity
NetView	OS/2
Redbooks	RedbooksLogo
RS/6000	SP
System/390	Tivoli
TME	XT

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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 IBM Redbooks

For information on ordering these publications see "How to get IBM Redbooks" on page 285.

- *HACMP/ES Customization Examples* , SG24-4498
- *Migrating to HACMP/ES* , SG24-5526
- *IBM Certification Study Guide AIX HACMP* , SG24-5131
- *HACMP Enhanced Scalability Handbook* , SG24-5328
- *HACMP Enhanced Scalability: User-Defined Events* , SG24-5327
- *HACMP Enhanced Scalability* , SG24-2081

D.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at ibm.com/redbooks for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
IBM System/390 Redbooks Collection	SK2T-2177
IBM Networking Redbooks Collection	SK2T-6022
IBM Transaction Processing and Data Management Redbooks Collection	SK2T-8038
IBM Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
IBM AS/400 Redbooks Collection	SK2T-2849
IBM Netfinity Hardware and Software Redbooks Collection	SK2T-8046
IBM RS/6000 Redbooks Collection	SK2T-8043
IBM Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

D.3 Other resources

These publications are also relevant as further information sources:

- *HACMP V4.3: Concepts and Facilities*, SC23-4276
- *HACMP V4.3 AIX: Planning Guide*, SC23-4277

- *HACMP V4.3 AIX: Install Guide*, SC23-4278
- *HACMP V4.3 AIX: Administration Guide*, SC23-4279
- *HACMP V4.3 AIX: Troubleshooting Guide*, SC23-4280
- *HACMP V4.3 AIX: Program Client Applications*, SC23-4282
- *HACMP V4.3 AIX: Program Locking Applications*, SC23-4281
- *HACMP V4.3 AIX: Enhanced Scalability & Administration Guide, Vol. 1*, SC23-4284
- *HACMP V4.3 AIX: Enhanced Scalability & Administration Guide, Vol. 2*, SC23-4306
- *RSCT: Event Management Programming Guide and Reference*, SA22-7354
- *RSCT: Group Services Programming Guide and Reference*, SA22-7355
- *RSCT: First Failure Data Capture Programming*, SA22-7454
- *TME 10 Framework 3.6 Planning & Installation Guide*, SC31-8432

D.4 Referenced Web sites

These Web sites are also relevant as further information sources:

- `http://www.ibm.com/servers/aix/products/ibmsw/list`
  IBM Application Availability Guide (Alphabetical Software Listing)
- `http://techsupport.services.ibm.com/rs6000/fixes`
  Fixes, drivers, updates, tools

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** **ibm.com**/redbooks

  Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

  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.

- **E-mail Orders**

  Send orders by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States or Canada	pubscan@us.ibm.com
Outside North America	Contact information is in the "How to Order" section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the "How to Order" section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the "How to Order" section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbooks fax 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.

Index

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at **ibm.com**/redbooks
- Fax this form to: USA International Access Code + 1 845 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number **Redbook Title**	SG24-5979-00 Exploiting HACMP 4.4: Enhancing the Capabilities of Cluster Multi-Processing
Review	
What other subjects would you like to see IBM Redbooks address?	
Please rate your overall satisfaction:	O Very Good O Good O Average O Poor
Please identify yourself as belonging to one of the following groups:	O Customer O Business Partner O Solution Developer O IBM, Lotus or Tivoli Employee O None of the above
Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	O Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. **ibm.com**/privacy/yourprivacy/

IBM

Redbooks

Exploiting HACMP 4.4: Enhancing the Capabilities of Cluster

Exploiting HACMP 4.4:
Enhancing the Capabilities of Cluster Multi-Processing

Determining the state of your application with Application Monitoring

Getting the most out of Tivoli Cluster Monitoring

Utilizing Cascading without Fallback

IBM High Availability Cluster Multi-Processing for AIX (HACMP) is designed to detect system failures and manage fallover to a recovery processor with a minimal loss of end-user time. HACMP Version 4.4.0 offers improved usability, more flexible installation options, and additional hardware and software support for RS/6000 customers with mission-critical applications.

This IBM Redbook provides information on Application Monitoring, Tivoli cluster monitoring, cascading without fallback, cluster verification enhancements, tuning parameters, administrative task enhancements, NFS function of HACMP 4.4, and upgrading/migrating to HACMP 4.4.

This IBM Redbook is intended to help IBM customers, IBM business partners, IBM sales professionals, and IBM I/T specialists wishing to obtain a reference for HACMP 4.4 enhancements.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-5979-00 ISBN 0738418684