IBM

Guide to IBM PowerHA SystemMirror for AIX Version 7.1.3

Outlines the latest PowerHA enhancements

Describes clustering with unicast communications

Includes migration scenarios

Dino Quintero
Alex Abderrazag
Bernhard Buehler
Primitivo Cervantes
Bharathraj Keshavamurthy
Kunal Langer
Luciano Martins
Ashish Nainwal
Minh Pham
Katharina Probst
Matt Radford
Bjorn Roden
Michael Schmut
Isac Silva
Yefei Song
Ben Swinney
Ashraf Ali Thajudeen
Marian Tomescu
Sascha Wycisk

Redbooks

ibm.com/redbooks

International Technical Support Organization

Guide to IBM PowerHA SystemMirror for AIX, Version 7.1.3

August 2014

SG24-8167-00

Note: Before using this information and the product it supports, read the information in “Notices” on page ix.

First Edition (August 2014)

This edition applies to IBM AIX 7.1 TL3 SP1, IBM PowerHA SystemMirror 7.1.2 SP3, IBM PowerHA SystemMirror 6.1 running on IBM AIX 6.1, IBM PowerHA SystemMirror 7.1.2 running on IBM AIX 7.1.2, IBM DB2 9.7.0.8, GSKit 8.0.50.10, TDS 6.3.0.24, IBM Tivoli Monitoring 6.2.2.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
BladeCenter®
DB2®
developerWorks®
Domino®
DS8000®
eServer™
FileNet®
GDPS®
Geographically Dispersed Parallel Sysplex™
Global Technology Services®
GPFS™
HACMP™
HyperSwap®
IBM®
Lotus®
Parallel Sysplex®
POWER®
Power Systems™
POWER6®
POWER7®
PowerHA®
PowerVM®
PureFlex®
Rational®
Redbooks®
Redpaper™
Redbooks (logo) ®
System p®
System p5®
System Storage®
System x®
System z®
SystemMirror®
Tivoli®
WebSphere®
XIV®

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redbooks® publication for IBM Power Systems™ with IBM PowerHA® SystemMirror® Standard and Enterprise Editions (hardware, software, practices, reference architectures, and tools) documents a well-defined deployment model within an IBM Power Systems environment. It guides you through a planned foundation for a dynamic infrastructure for your enterprise applications.

This information is for technical consultants, technical support staff, IT architects, and IT specialists who are responsible for providing high availability and support for the IBM PowerHA SystemMirror Standard and Enterprise Editions on IBM POWER® systems.

Authors

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

Dino Quintero is a Complex Solutions Project Leader and an IBM Senior Certified IT Specialist with the ITSO in Poughkeepsie, NY. His areas of knowledge include enterprise continuous availability, enterprise systems management, system virtualization, technical computing, and clustering solutions. He is an Open Group Distinguished IT Specialist. Dino holds a Master of Computing Information Systems degree and a Bachelor of Science degree in Computer Science from Marist College.

Alex Abderrazag is a Consulting IT Specialist on the worldwide IBM education events team. Alex has more than 20 years experience working with UNIX systems and has been responsible for managing, teaching, and developing the IBM Power, IBM AIX®, and Linux education curriculum. Alex is a Chartered Member of the British Computer Society and a Fellow of the Performance and Learning Institute. Alex holds a BSc (Honors) degree in Computer Science and has many AIX certifications, including IBM Certified Advanced Technical Expert and IBM Certified Systems Expert for IBM PowerHA (exam developer).

Bernhard Buehler is an IT Specialist for availability solutions on IBM Power Systems in Germany. He works for IBM STG Lab Services in La Gaude, France. He has worked at IBM for 33 years and has 24 years of experience in AIX and the availability field. His areas of expertise include AIX, IBM PowerHA, HA architecture, shell script programming, and AIX security. He is a co-author of several IBM Redbooks publications and of several courses in the IBM AIX curriculum.

Primitivo Cervantes is a certified Senior IT Specialist in the USA. He is a high-availability and disaster-recovery (HA and DR) specialist with high level of experience in dealing with complex environments and applications. He has been working with IBM clients, designing, implementing and educating at client locations with HA and DR since the early 1990s, including many multisite clustered environments.

Bharathraj Keshavamurthy is an IBM enterprise solutions Performance Architect who designs solutions that meet the nonfunctional requirements of the system. He is also involved in end-to-end performance engineering for the solution as a whole, working with cross-brand products, with Java virtual machine performance as his primary area of research and work. His writing experience includes IBM developerWorks® articles about IBM Rational® Performance Tester, IBM Redbooks publications about IBM PowerVM® analytics optimization, and an IBM Redpaper™ publication about cloud performance. He focuses on writing articles based on real-world experience that help solve the problems in the worldwide technical community.

Kunal Langer is a Technical Solutions Architect for Power Systems in STG Lab Services, India. He has more than seven years of experience in IBM Power Systems, with expertise in the areas of PowerHA, PowerVM, and AIX security. He conducts PowerHA Health Checks, AIX Health Checks, and IBM PowerCare Services Availability Assessments for IBM clients in the ISA and EMEA regions. He has co-authored a few IBM Redbooks publications and written articles for IBM developerWorks and IBM Systems Magazine. He holds a Bachelor's in Engineering degree in Computer Science and Technology.

Luciano Martins is a is a Senior IT Specialist in IBM Global Technology Services® in IBM Brazil. He has 13 years of experience in the IT industry, focused mainly on IBM Power and IBM Storage System solutions. He is a Certified Technical Expert for AIX and PowerHA and POWER Systems and is also an IBM Certified Cloud Computing Infrastructure Architect. His areas of expertise include AIX, PowerVM, PowerHA, IBM General Parallel File System (GPFS™), cloud computing, and storage systems. Luciano holds a degree in Computer Science from Universidade da Amazônia, in Brazil, with academic work in virtualization and high availability.

Ashish Nainwal is a Managing Consultant for Power Systems in Systems Technology Group Lab Services, ASEAN. With almost nine years of experience in IBM Power Systems, he has expertise in the Power suite, including PowerHA, PowerVM, and AIX Security. He is a vetted PowerCare Services consultant and has conducted performance, availability, and security engagements for IBM clients in all countries of ASEAN region. He has published articles on IBM developerWorks and holds an IBM patent for security architecture. Ashish holds a Bachelors in Technology degree from the National Institute of Technology, India, and an MBA from Symbiosis International University, in India.

Minh Pham is a Development Support Specialist for PowerHA and Cluster Aware AIX in Austin, Texas. She has worked for IBM for 13 years, including six years in System p® microprocessor development and seven years in AIX development support. Her areas of expertise include core and chip logic design for IBM System p and AIX with PowerHA. Minh holds a Bachelor of Science degree in Electrical Engineering from the University of Texas at Austin.

Katharina Probs is a member of the IBM/SAP porting team from the IBM Lab in Böblingen, Germany. Her focus is the enablement and optimization of SAP solutions on the IBM infrastructure. She is a recognized worldwide expert on high availability and disaster recovery solutions for SAP landscapes.

Matt Radford is the Team Leader for the Front Office European Support team for PowerHA. He has seven years of experience in AIX support and PowerHA. He holds a Bsc (Honors) degree in Information Technology from the University of Glamorgan. He is co-authored previous Redbooks publications on PowerHA versions 7.10 and 7.12.

Bjorn Roden works for IBM Systems Technology Group Lab Services. He is based in Dubai and is a member of IBM worldwide PowerCare Services teams for availability, performance, and security, specializing on high-end Power Systems and AIX. Bjorn holds MSc, BSc, and DiplSSc in Informatics, plus a BCSc and a DiplCSc in Computer Science from Lund and Malmo Universities in Sweden. He is an IBM Redbooks Platinum Author, IBM Certified Specialist (Expert), and has worked in leader and implemented roles with designing, planning, implementing, programming, and assessing high-availability, resilient and secure, and high-performance systems and solutions since 1990.

Michael Schmut is a a Software Engineer at SAP, working on AIX development. His focus is on running SAP on PowerHA installations. Before he started at the IBM development lab, he worked on many client projects related to SAP and AIX in the role of Software Architect.

Isac Silva is an IT Specialist and IT Infrastructure Architect with more than 14 years of experience in IBM Power Systems. His areas of expertise are IBM AIX and IBM PowerHA. He is a Technical Leader and, along those years of experience, he has had roles and responsibility in support, deployment, development, installation, problem determination, disaster recovery, infrastructure, and networking (TCP/IP, firewall, QoS). He is currently a Development Support Specialist at IBM in Austin, Texas. He is working with the worldwide Level 2 IBM PowerHA and Cluster Aware AIX technology.

Yefei Song is an Advisory IT Specialist in IBM Singapore. He is an IBM Certified Advanced Technical Expert, with more than six years of experience in infrastructure solution delivery and technical support services for IBM POWER and storage. He provides consultation, planning, and support services for AIX, PowerHA, and server and storage virtualization.

Ben Swinney is a Senior Technical Specialist for IBM Global Technology Services in Melbourne, Australia. He has more than 14 years of experience in AIX, IBM Power Systems, PowerVM, and PowerHA. He has worked for IBM for more than six years, both within IBM UK and IBM Australia. His areas of expertise include infrastructure design and implementation, high availability solutions, server administration, and virtualization.

Ashraf Ali Thajudeen is an Infrastructure Architect in IBM Singapore Global Technologies Services. He has more than eight years of experience in High Availability and Disaster Recovery Architectures in UNIX environments. He is an IBM Master Certified IT Specialist in Infrastructure and Systems Management and is TOGAF 9-certified in Enterprise architecture. He has wide experience in designing, planning, and deploying PowerHA solutions across ASEAN strategic outsourcing accounts. His areas of expertise include designing and implementing PowerHA and IBM Tivoli® automation solutions.

Marian Tomescu has 15 years of experience as an IT Specialist and currently works for IBM Global Technologies Services in Romania. Marian has nine years of experience in Power Systems. He is a certified specialist for IBM System p Administration, PowerHA for AIX, and for Tivoli Storage Management Administration Implementation, an Oracle Certified Associated, an IBM eServer™ and Storage Technical Solutions Certified Specialist, and a Cisco Information Security Specialist. His areas of expertise include Tivoli Storage Manager, PowerHA, PowerVM, IBM System Storage®, AIX, General Parallel File System (GPFS), IBM VMware, Linux, and Microsoft Windows. Marian has a Master's degree in Electronics Images, Shapes and Artificial Intelligence from Polytechnic University - Bucharest, Electronics and Telecommunications, in Romania.

Sascha Wycisk is a Senior IT Architect in Germany. He has 15 years of experience in IBM client engagements and more than seven years of experience with IBM PowerHA on Power systems and AIX. His areas of expertise include high availability and disaster recovery architectures in Unix and Linux environments.

Special acknowledgement goes to Ken Fleck and his team at the Poughkeepsie Benchmark Center for lending the team hardware to create sample scenarios for this publication.

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

David Bennin, Ella Buchlovic, and Richard Conway
International Technical Support Organization, Poughkeepsie Center
IBM USA

Esdras Cruz-Aguilar, Dwip Banerjee, Shawn Bodily, Michael Coffey, Glen Corneau, Paul Desgranges, Zhi-Wei Dai, Ken Fleck, P. I. Ganesh, Michael Herrera, Kam Lee, Gary Lowther, Robert Luther, Bill Miller, Philip Moore, Paul Moyer, Suriyan Ramasami, Brian Rousey, and Ravi Shankar
IBM USA

Shivendra Ashish, Chennakesavulu Boddapati, Lakshmipriya Kanduru, Subramaniam Meenakshisundaram, Arun H. Nagraj, Dimpu K. Nath, and Vijay Yalamuri
IBM India

Peter Juerss, Stephen Lutz, Michael Mueller, Donal O'Connell, Jorge Rodriguez, and Ingo Weber
IBM Germany

Jon Kowszun
IBM Switzerland

Ahmad Y. Hussein and David Kgabo
IBM South Africa

Jorge Gómez García
IBM Mexico

Angie Nobre Cocharero
IBM Brazil

Xiao Er Li
IBM China

Hiroyuki Tanaka
IBM Japan

Philippe Hermès
IBM France

Tony Steel
IBM Australia

Now you can become a published author, too

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time. Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us.

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form:

  ibm.com/redbooks

- Send your comments by email:

  redbooks@us.ibm.com

- Mail your comments:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

Introduction to IBM PowerHA SystemMirror for AIX 7.1.3, Standard and Enterprise Editions

In this chapter, we describe the new features, differences, and disaster recovery offerings in the IBM PowerHA SystemMirror for AIX, Standard and Enterprise Editions, version 7.1.3. This book provides useful scenarios and examples that demonstrate these capabilities and their use in resolving challenging disaster recovery situations.

We cover the following topics:

- How IBM PowerHA SystemMirror helps
- Disaster recovery planning criteria and solution tiers
- Storage replication and mirroring

This book is helpful to IBM Power Systems specialists who use the PowerHA SystemMirror solution for high availabiity and went to align their resources with the best disaster recovery model for their environment. With each technology refresh and new server consolidation, it is not unreasonable to consider using your existing servers in your recovery environment. If you are looking for an entry point into a high-availability solution that incorporates disaster recovery, you can use your existing hardware and select the replication mechanism that fits your needs.

1.1 How IBM PowerHA SystemMirror helps

Data center and services availability are some of the most important topics for IT infrastructure, and each day draws more attention. Not only natural disasters affect normal operations, but human errors and terrorist acts might affect business continuity. Even with fully redundant infrastructure, services are vulnerable to such disasters.

One of the PowerHA SystemMirror main goals is to help continuous business services operations even after one (or more) components fails. Unexpected failures can be related to human errors or other errors. Either way, the PowerHA SystemMirror design phase is intended to remove any single point of failure (SPOF) from the environment by using redundant components and automated PowerHA SystemMirror procedures.

It is important to remember that any hardware component can fail and cause application disruptions. So, when you plan a high availability environment, you must check all components, from disk access to power circuits, for redundancy.

Replication of data between sites is a good way to minimize business disruption because backup restores can take too long to meet business requirements or equipment might be damaged, depending on the extent of the disaster, and not available for restoring data. Recovery options typically range in cost, with the least expensive involving a longer time for recovery to the most expensive providing the shortest recovery time and being the closest to having zero data loss. A fully manual failover normally requires many specialists to coordinate and perform all of the necessary steps to bring the services up to another site. Even with a good disaster recovery plan, it can take longer than business requirements allow. High availability software minimizes downtime of services by automating recovery actions when failures are detected on the various elements of the infrastructure.

Figure 1-1 on page 3 shows an example of an environment that has no redundant hardware components, so it would not tolerate failure of any component.

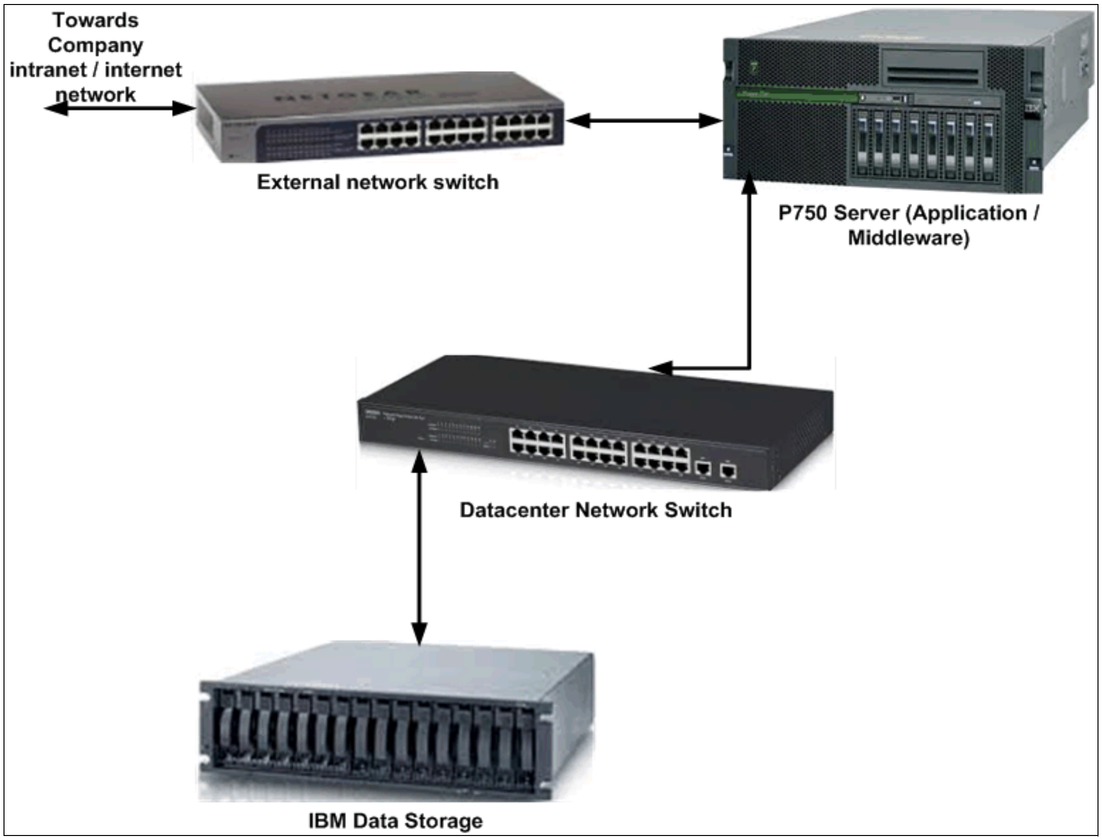


Figure 1-1 Environment with no redundancy of hardware components

With this configuration, if any component fails, for example the data center network switch or the SAN switch, the application that runs on the IBM Power 750 server becomes unavailable becaused it lacks redundancy, or a failover alternative. The IBM Power 750 server experiences a disruption until all failing components are replaced or fixed. Depending on which component fails, it can take from hours to weeks to fix it, which affects service availability and, in the worst case, data availability.

Figure 1-2 on page 4 shows a sample client environment with redundant network connections via dual network switches to ensure connectivity between server and storage.

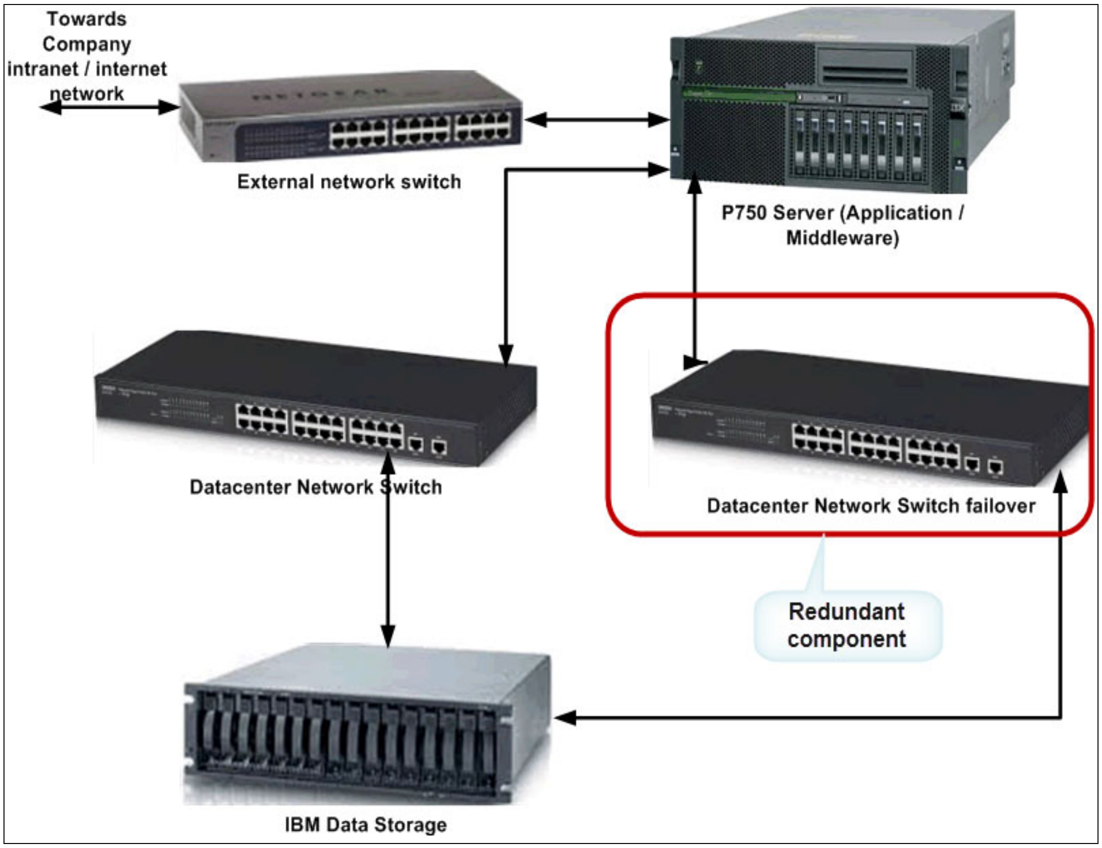


Figure 1-2 Environment with a redundant network switch

The configuration in Figure 1-2 enables the IBM Power 750 server to be more resilient in response to environmental issues. This resiliency keeps business services available even with failures in parts of the company infrastructure.

Note: High availability solutions help eliminate SPOFs through appropriate design, planning, selection of hardware, configuration of software, and carefully controlled change management discipline. "High availability" does not mean that there is no interruption to the application. Therefore, it is called *fault resilient* rather than *fault tolerant*.

Documentation: For more information, see the IBM PowerHA SystemMirror for AIX V7.1 documentation in the IBM Knowledge Center:

http://ibm.co/1t5pZ9p

1.2 Disaster recovery

Data centers are susceptible to outages due to natural events, such as earthquakes, severe storms, hurricanes, and other factors, such as fires, power outages, and related events.

As a result of these outages, your business might incur losses due to damages to the infrastructure and costs to restore systems to operation. Even a bigger cost is the data loss caused by outages. Millions of bytes of valuable information can never be restored if there is no proper planning during the design phase of the deployment. Proper planning might include making regular data backups, synchronizing or replicating data with different data storage systems in different geographical zones, and planning for a redundant data storage system that comes online if the primary node goes down due to outages.

Note: In some instances, the application might also manage the replication of the data to the disaster recovery site.

PowerHA SystemMirror 7.1.3 for AIX, Standard and Enterprise Editions, helps automate the recovery actions when failures are detected on the nodes.

1.2.1 High-availability criteria for designing your systems deployment

The idea of a fast failover in the event of a problem, or the *recovery time objective* (RTO), is important, but that should not be the only area of focus. Ultimately, the consistency of the data and whether the solution meets the *recovery point objective* (RPO) are what make the design worth the investment. Do not enter a disaster recovery planning session expecting to truly achieve the Five Nines of Availability solely by implementing a clustering solution.

Table 1-1 The Five Nines of Availability

Criteria	Uptime percentage in a year	Maximum downtime per year
Five nines	99.999%	5 minutes 35 seconds
Four nines	99.99%	52 minutes 33 seconds
Three nines	99.9%	8 hours 46 minutes
Two nines	99%	87 hours 36 minutes
One nine	90%	36 days 12 hours

There are certain questions to ask when planning a disaster recovery solution to achieve an adequate return on investment. For example, does it account for the time for planned maintenance? If so, have you backtracked to make sure that you understand the planned maintenance or downtime window?

The Five Nines of Availability (Table 1-1) give us performance criteria only for *unplanned* downtime, but it is essential to plan for *planned* downtime each year, too. Version 7 of SystemMirror does not support a nondisruptive upgrade. Therefore, you must consider the impact of other service interruptions in the environment that often require the services to go offline for a certain amount of time, such as upgrades to the applications, the IBM AIX operating system, and the system firmware. These must be included in the planned downtime considerations. For more information on the difference between planned and unplanned downtime, see the shaded box titled “Planned downtime versus unplanned downtime”, which follows.

The Standard and Enterprise Editions of PowerHA SystemMirror for AIX 7.1.3 reliably orchestrate the acquisition and release of cluster resources from one site to another. They also provide quick failover if there is an outage or natural disaster.

Solutions in the other tiers can all be used to back up data and move it to a remote location, but they lack the automation that the PowerHA SystemMirror provides. By looking over the recovery time axis (Figure 1-3 on page 7), you can see how meeting an RTO of less than four hours can be achieved with the implementation of automated multisite clustering.

Planned downtime versus unplanned downtime

Planned downtime is a period of time during which all system operations are shut down and turned off in a graceful manner, with the intent to implement upgrades to the hardware or software or to do repairs or make changes. Planned downtime occurs when the infrastructure specialists have clearly demarcated a period of time and reserved that time for carrying out these environmental changes. During planned downtime, the IBM client is typically aware and has predetermined the cost of upgrades and revenue losses due to unavailability of IT services.

Unplanned downtime is when all system operations shut down after a catastrophe or accident, such as fires, power outages, earthquakes, or hurricanes. *Unplanned downtimes* are unexpected and incur undetermined repair costs and revenue losses due to service unavailability. Unplanned downtime can occur any time during any period for many reasons. Therefore, infrastructure architects should include unplanned downtime during the design and deployment phases of IT solutions.

High availability and disaster recovery requires a balance between recovery time requirements and cost. Various external studies are available that cover dollar loss estimates for every bit of downtime that is experienced as a result of service disruptions and unexpected outages. Decisions must be made about what parts of the business are important and must remain online to continue business operations.

Beyond the need for secondary servers, storage, and infrastructure to support the replication bandwidth between two sites, it is important to answer the following questions:

- Where does the staff go in the event of a disaster?
- What if the technical staff that manages the environment is unavailable?
- Are there facilities to accommodate the remaining staff, including desks, phones, printers, desktop PCs, and so on?
- Is there a documented disaster recovery plan that can be followed by non-technical staff, if necessary?

1.2.2 Differences in disaster recovery solution tiers

Figure 1-3 shows various tiers of a disaster recovery solution and why the PowerHA SystemMirror Enterprise Edition is considered a Tier 7 recovery solution. The key point is that there are many tiers of disaster recovery. Depending on you high availability requirements and downtime sensitivity, IBM PowerHA Enterprise Edition provides an efficient and automated business recovery solution.

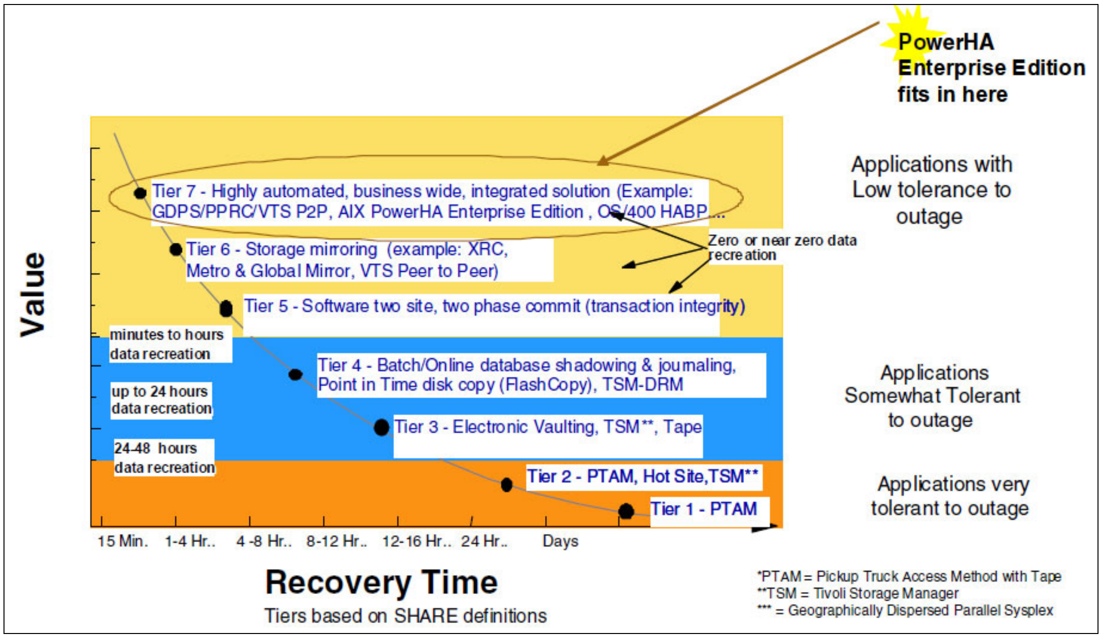


Figure 1-3 Tiers of disaster recovery solutions, IBM PowerHA SystemMirror 7.1.3 Enterprise Edition

1.3 Storage replication and mirroring

Storage replication always presents us with the problem of data integrity in case of a failover. There are good chances of database corruption or incorrect data getting replicated to the target copy. This causes data integrity issues at both source and target sites.

Replicating the data addresses only one problem. In a well-designed disaster recovery solution, a backup and recovery plan must also exist. Tape backups, snapshots, and flash memory copies are still an integral part of effective backup and recovery. The frequency of these backups at both the primary and remote locations must also be considered for a thorough design.

Tip: An effective backup and recovery strategy should leverage a combination of tape and point-in-time disk copies to protect unexpected data corruption. Restoration is very important, and regular restore tests need to be performed to guarantee that the disaster recovery is viable.

There are two types of storage replication: *synchronous* and *asynchronous*:

- Synchronous replication considers only the I/O completed after the write is done on both storage repositories. Only synchronous replication can guarantee that 100% of transactions were correctly replicated to the other site. But because this can add a considerable amount of I/O time, the distance between sites must be considered for performance criteria.
- This is the main reason that asynchronous replication is used between distant sites or with I/O-sensitive applications. In synchronous mirroring, both the local and remote copies must be committed to their respective subsystems before the acknowledgment is returned to the application. In contrast, asynchronous transmission mode allows the data replication at the secondary site to be decoupled so that primary site application response time is not affected.

  Asynchronous transmission is commonly selected when it is known that the secondary site's version of the data might be out of sync with the primary site by a few minutes or more. This lag represents data that is unrecoverable in the event of a disaster at the primary site. The remote copy can lag behind in its updates. If a disaster strikes, it might never receive all of the updates that were committed to the original copy.

Although every environment differs, the farther that the sites reside from each other, the more contention and disk latency are introduced. However, there are no hard-set considerations that dictating whether you need to replicate synchronously or asynchronously. It can be difficult to provide an exact baseline for the distance to delineate synchronous versus asynchronous replication.

2

Basic concepts

This chapter introduces the basic concepts of high availability and describes the IBM PowerHA SystemMirror for AIX functions. It also provides information about the basics of virtualization and the Virtual I/O Server (VIOS).

This chapter covers the following topics:

- High availability and disaster recovery
- PowerHA architecture
- PowerHA SystemMirror in a virtualized environment
- Virtualization in IBM Power Systems
- Important considerations for VIOS
- SAN- or FC-based heartbeat configuration in virtualized environment

2.1 High availability and disaster recovery

IBM PowerHA SystemMirror 7.1.3 for AIX helps automate fallover and recovery actions on node failures and provides application monitoring events for high availability. PowerHA SystemMirror Enterprise Edition helps automate recovery actions on storage failures for selected storage, controls storage replication between sites, and enables recovery after failure of an entire site to help ensure that data copies are consistent.

For both Standard and Enterprise Editions, the IBM Systems Director server can be enabled to manage clusters with its integrated GUI by installing the PowerHA plug-in which was enhanced to support the disaster recovery enablement features added in PowerHA SystemMirror version 7.1.2 Enterprise Edition (for example, storage replication). The PowerHA SystemMirror Enterprise Edition gives you the ability to discover the existing PowerHA SystemMirror clusters, collect information and a variety of reports about the state and configuration of applications and clusters, and receive live and dynamic status updates for clusters, sites, nodes, and resource groups. A single sign-on capability gives you full access to all clusters with only one user ID and password, access and search log files. You can display a summary page where you can view the status of all known clusters and resource groups, create clusters, add resource groups with wizards, and apply updates to the PowerHA SystemMirror Agent by using the Systems Director Update Manager.

2.2 PowerHA architecture

Before starting to review the PowerHA SystemMirror features, it helps to understand the PowerHA goals and concepts.

One of the main goals of the PowerHA SystemMirror is to provide continuous business services even after multiple component failures. Unplanned or unexpected failures can occur at any time. They can be related to human errors, or not. Either way, the intention of the PowerHA design phase is to remove any *single point of failure (SPOF)* by using redundant components wherever possible.

It is important to understand that any component can fail and cause application disruptions. When planning a high availability environment, you must provide redundancy and check all components.

Figure 2-1 on page 11 shows an environment without fault tolerance or redundancy. If any component fails (for example, a network switch or a SAN switch), workloads running on an IBM Power server become unavailable.

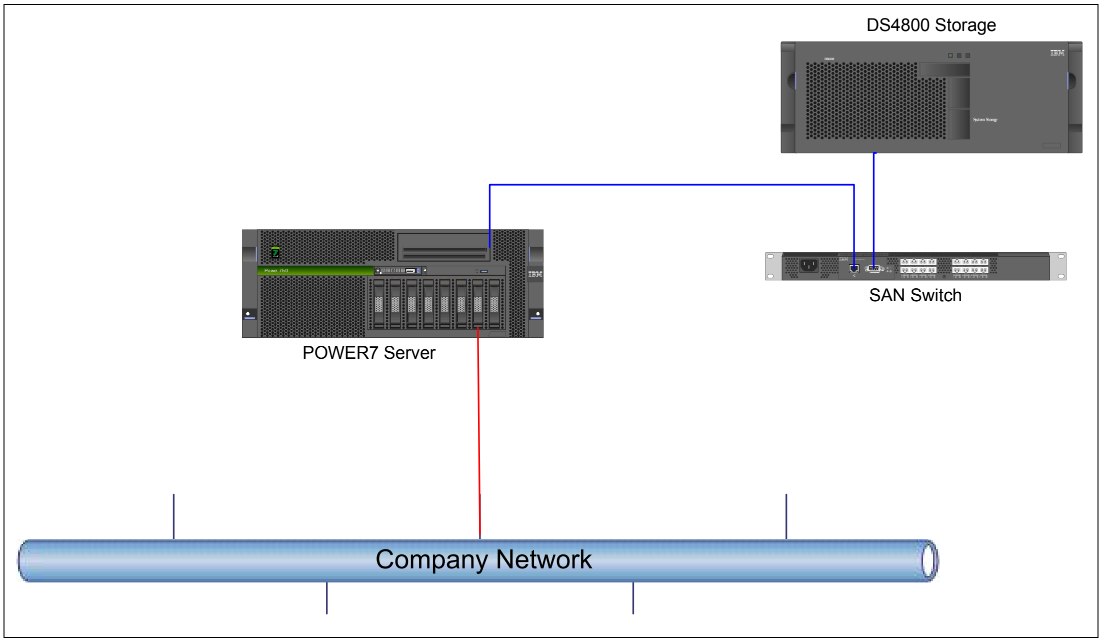


Figure 2-1 Sample environment without fault tolerance

If a failure occurs in this environment, users experience disruption in services until all failing components are replaced or fixed. Depending on which component has failed, it might take from a few hours to a few weeks to fix, so it impacts service or data availability.

Figure 2-2 on page 12 shows a sample cluster environment with redundant network connections, and dual SAN switches for disk access. This configuration enables the Power server to be more resilient to failures, keeping business services available even with some service issues in part of the company infrastructure.

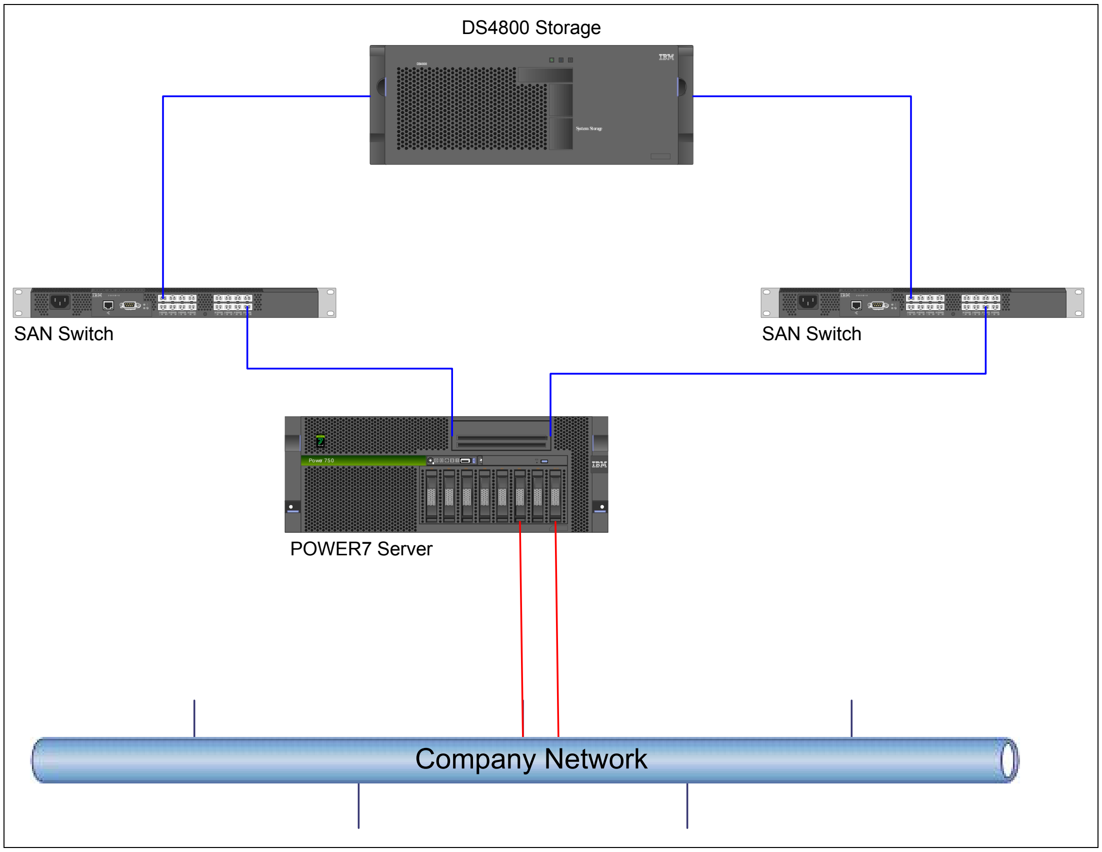


Figure 2-2   Sample environment with redundant components

Even without using PowerHA, this configuration (Figure 2-2) is resilient to some possible failures. If an IP network switch goes down, the server has a secondary network connection on a redundant switch. If a SAN switch goes down, the server can get storage access through a secondary SAN switch. This makes the customer environment more resilient and flexible to unexpected issues, allowing business services to be active and continue.

PowerHA SystemMirror for AIX requires redundancy for most of its components, for example:

- Network access
- SAN disk access
- Local disk
- SAN disk formatting (RAID)

When you plan to migrate a current production environment to a PowerHA cluster infrastructure, all possible components must be assessed to address all necessary redundancies before cluster startup. This avoids issues caused by a SPOF.

> **Note:** A high availability solution, such as PowerHA SystemMirror, ensures that the failure of any component of the solution, whether hardware, software, or other, does not cause the application and its data to be inaccessible. This is achieved through the elimination or masking of both planned and unplanned downtime. High availability solutions must eliminate all single points of failure wherever possible through design, planning, selection of hardware. and carefully controlled change management.

Before proceeding with the virtualization and other concepts discussed in this chapter, we review the fundamental concepts of PowerHA. This helps you better understand all scenarios, configurations, and concepts in this book.

Note: For more information about PowerHA architecture and concepts, download the *PowerHA SystemMirror Concepts* document from the PowerHA SystemMirror 7.1 for AIX PDFs page in the IBM Knowledge Center:

http://ibm.co/1nTups9

2.2.1 Reliable Scalable Cluster Technology

Reliable Scalable Cluster Technology, or RSCT (Figure 2-3 on page 14), is a set of low-level operating system components that allow implementation of cluster technologies, such as PowerHA SystemMirror, General Parallel File Systems (GPFS), and so on.

All of the RSCT functions are based on the following components:

- **Resource Monitoring and Control (RMC) subsystem:** This is considered the backbone of RSCT. The RMC runs on each single server and provides a common abstraction layer of server resources (hardware or software components).
- **RSCT core resource manager:** This is a software layer between a resource and RMC. The resource manager maps the abstraction defined by RMC to real calls and commands for each resource.
- **RSCT security services:** These provide the security infrastructure required by RSCT components to authenticate the identity of other parties.
- **Topology service subsystem:** This provides the infrastructure and mechanism for the node and network monitoring and failure detection.

  Important: Starting with PowerHA 7.1.0, the RSCT topology services subsystem is deactivated, and all of its functions are performed by Cluster Aware AIX (CAA) topology services.

- **Group services subsystem:** This coordinates cross-node operations in a cluster environment. The subsystem is responsible for spanning changes across all cluster nodes and making sure that all of them finish properly, with all modifications performed.

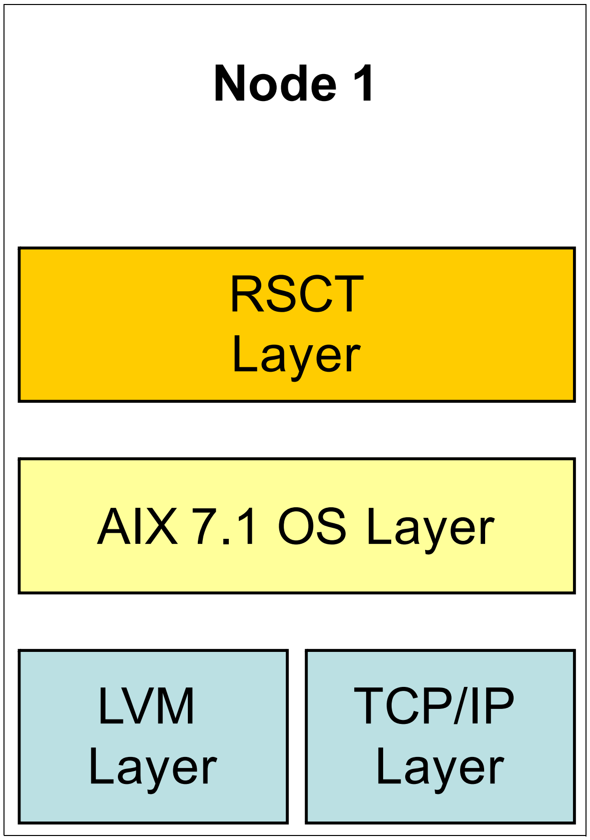


Figure 2-3 RSCT placement on an IBM AIX server

> **Note:** For more information,see the IBM RSCT: Administration Guide(SA22-7889-20):
> http://www.ibm.com/support/docview.wss?uid=pub1sa22788920

2.2.2 Cluster Aware AIX (CAA)

AIX 7.1 and AIX 6.1 TL6 introduced a built-in cluster capability called *Cluster Aware AIX (CAA)*. This feature enables system administrators to create clusters from a group of AIX servers by using commands and programming APIs. CAA provides a kernel-based heartbeat, monitoring, and event management infrastructure. Table 2-1 shows the features by release.

Table 2-1 CAA release history

Release	AIX version	Fileset	PowerHA version
R1	AIX 6.1 TL7 AIX 7.1 TL1	bos.cluster.rte 6.1.7.XX or 7.1.1.XX	PowerHA 7.11
R2	AIX 6.1 TL8 AIX 7.1 TL2	bos.cluster.rte 6.1.8.XX or 7.1.2.XX	PowerHA 7.11 PowerHA 7.12
R3	AIX 6.1 TL9 AIX 7.1 TL3	bos.cluster.rte 6.1.9.XX or 7.1.3.XX	PowerHA 7.13

Even though CAA is primarily intended to provide a reliable layer of clustering infrastructure to high-availability software, such as PowerHA, you can directly use the CAA layer functions to aid your management tasks in your own computer environment.

CAA includes a component called a *cluster repository disk*, which is required for PowerHA cluster environments. This is a central repository for all cluster topology-related information and must be shared by all servers in the cluster. The repository disk is also used for the heartbeat mechanism.

In PowerHA 7.1.0, if a repository disk fails, the nodes shut down automatically. In PowerHA 7.1.1, enhancements were implemented for CAA, and a new feature called *repository disk resilience* was introduced to enable administrators to perform cluster maintenance tasks even after the failure of the repository disk.

CAA also supports online repository disk replacement with no cluster impact. Release 7.1.2 of PowerHA introduced the concept of a backup repository disk, which allows administrators to define an empty disk to be used for rebuilding the cluster repository in case the current repository disk encounters any failure. For more information about repository disk resilience or backup repository, see the IBM Redbooks publications titled *IBM PowerHA SystemMirror Standard Edition 7.1.1 for AIX Update*, SG24-8030, and the *IBM PowerHA SystemMirror 7.1.2 Enterprise Edition for AIX*, SG24-8106.

The following products or components use the CAA technology:

- Reliable Scalable Cluster Technology (RSCT) 3.1 and later
- IBM PowerHA 7.1 and later
- Virtual I/O Server (VIOS) 2.2.0.11, FP-24 SP-01 and later

Figure 2-4 shows a high-level architectural view of how PowerHA uses the Reliable Scalable Clustering Technology (RSCT) and the CAA architecture.

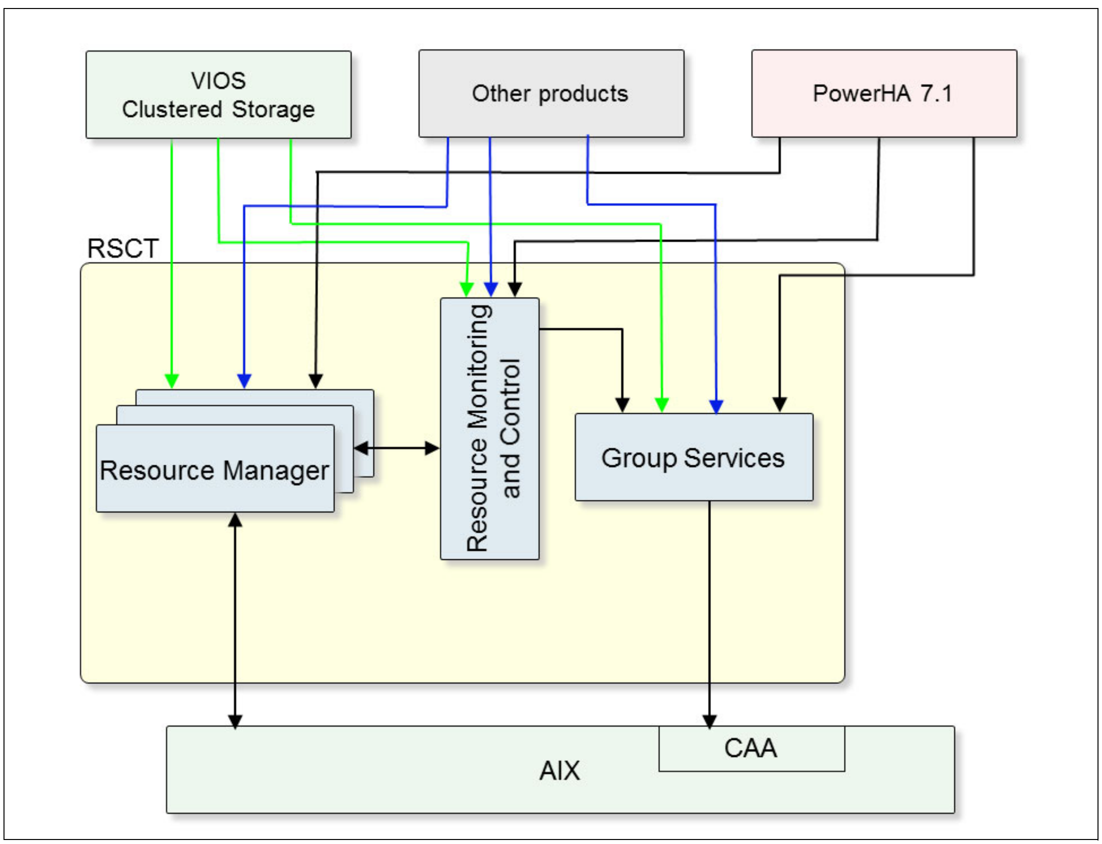


Figure 2-4 HA applications that use RSCT and CAA

2.2.3 Synchronous storage-based mirroring for the repository disk

The repository disk stores some of the configuration information centrally and provides the disk heartbeat function. Currently, only one disk is supported as a repository disk in a stretched cluster environment. Therefore, this disk should be highly available.

Note: The cluster repository disk can be re-created but cannot make cluster changes if the disk is not available. Implement mirroring if you want to make changes to the cluster while the disk is not available.

One possibility is to make the repository disk highly available by mirroring it at the hardware level over multiple storage servers, as shown in Figure 2-5.

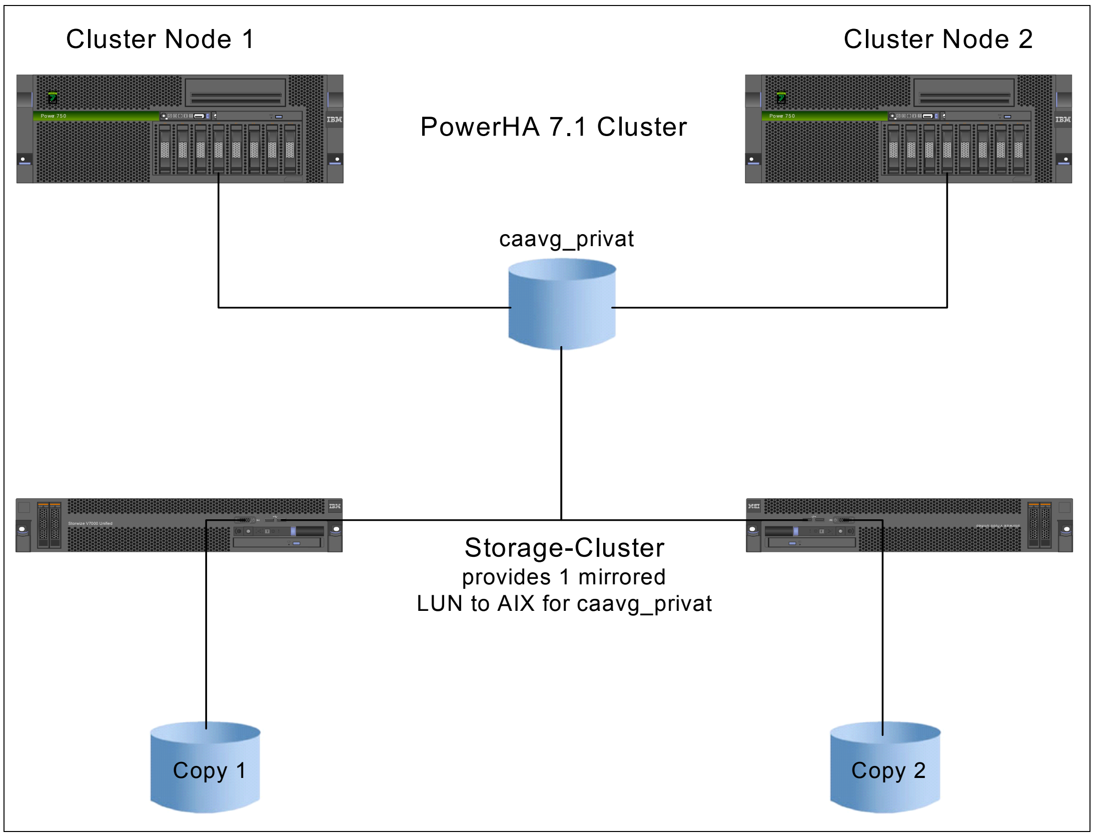


Figure 2-5 Mirroring the repository disk

2.2.4 PowerHA cluster components

This section describes the PowerHA cluster components.

PowerHA cluster

A cluster is set of computer systems connected together and sharing application data. They can all be in the same geographic place, in the same data center. or they can be in distant places, even worldwide.

By adopting cluster technologies, companies can increase service availability and reliability to their customers or even make disasters not visible to their customers. A clustered environment can help presentyour business as a better service providers.

Note: In a PowerHA cluster, many components are stored together (servers, applications, storage, and so on).

When a PowerHA cluster is set up, a logical name (cluster name) must be assigned, as shown in Figure 2-6. This name is used by PowerHA procedures to deal with specific groups of servers, services, and information.

```
                              COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

[TOP]
Cluster Name: oracluster
Cluster Connection Authentication Mode: Standard
Cluster Message Authentication Mode: None
Cluster Message Encryption: None
Use Persistent Labels for Communication: No
Repository Disk: hdisk2
Cluster IP Address: 228.1.1.30
There are 2 node(s) and 2 network(s) defined

NODE sapnfs1:
        Network net_ether_01
                oracle_svc1      172.16.21.65
[MORE...21]

F1=Help            F2=Refresh         F3=Cancel          F6=Command
F8=Image           F9=Shell           F10=Exit           /=Find
n=Find Next
```

Figure 2-6 Cluster name

Figure 2-6 shows the cluster topology. It can be checked by using the **`smitty sysmirror`** command and selecting **Cluster Nodes and Networks** → **Manage the Cluster** → **Display PowerHA SystemMirror Configuration**.

The same output is shown by using this command:

```
/usr/es/sbin/cluster/utilities/cltopinfo
```

PowerHA cluster nodes

A PowerHA cluster node can be any AIX based IBM Power server or LPAR that is running PowerHA services.

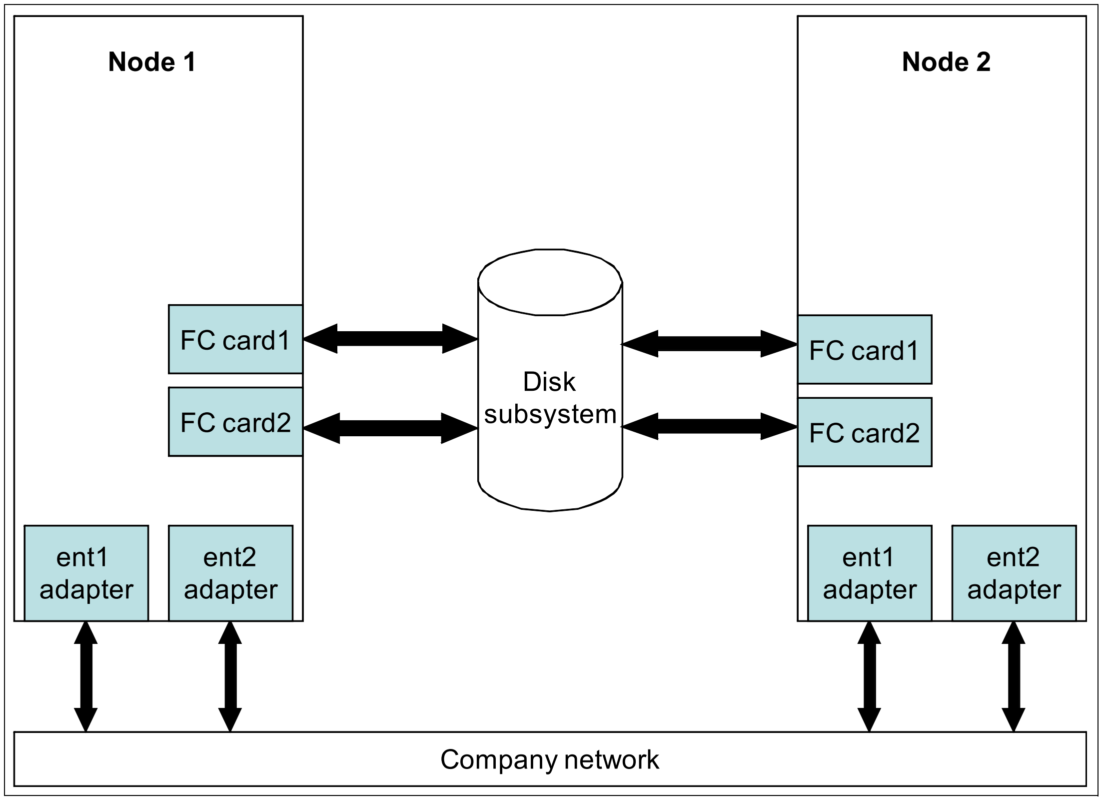


Figure 2-7 Standard two-node PowerHA cluster hardware

Figure 2-7 shows a standard cluster configuration with two nodes and redundant network and SAN access. The data is shared with the use of a shared disk subsystem.

In PowerHA version 7.1.3, up to 16 nodes can be included in a single cluster. PowerHA supports cluster nodes, such as IBM Power servers, Power blades, IBM PureFlex® Systems, or a combination of them.

PowerHA networks

For PowerHA, networks are paths through which cluster nodes communicate with each other and with the outside world. CAA heartbeat messages are also sent.

When defining a network, you can choose any name for the network, making it easier to identify it within PowerHA architecture. If you do not specify a name, PowerHA automatically assigns a network name by using the *net_ether_XX* pattern, as shown in Figure 2-8 on page 19.

Starting with PowerHA 7.1.1, the networks can be *public* or *private*. The main difference between public and private networks is that CAA does not perform heartbeat operations over a private network.

To change the network behavior, you can use **smitty sysmirror**, and select **Cluster Nodes and Networks** → **Manage Networks and Network Interfaces** → **Networks** → **Change/Show a Network**. Then, select the network that you want to change, as shown in Figure 2-8.

```
                            Change/Show a Network

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

                                                        [Entry Fields]
* Network Name                                        net_ether_01
  New Network Name                                   []
* Network Type                      [ether]
+
* Netmask(IPv4)/Prefix Length(IPv6)                  [255.255.254.0]
* Network attribute                   public
+

F1=Help                  F2=Refresh               F3=Cancel
F4=List
F5=Reset                 F6=Command               F7=Edit
F8=Image
F9=Shell                      F10=Exit                     Enter=Do
```

Figure 2-8 Cluster network configuration

PowerHA IP addresses and IP labels

The PowerHA cluster calls any IP that is used inside the cluster environment an *IP label*. In other words, an IP label is a name assigned to an IP address that is used in the cluster configuration. In PowerHA, different IP labels are used:

- **Boot (or base) IP label:** This is related to the IP address that is physically assigned to the Ethernet adapters. It is the IP address configured on nodes when they finish the boot process.
- **Service IP label:** This refers to the IP address used by the application services user to get into the application functions and data. The service IP label usually moves across cluster nodes. depending on which node currently hosts the application.
- **Persistent IP label:** In many cluster configurations, the boot IP addresses are part of non-routed network. For specific operating system maintenance tasks, system administrators need to reach specific nodes. Using the service IP is not a good choice because it might be a node where you do not want to perform the task. To make sure that a system administrator is able to log in and reach the node to perform the maintenance tasks, persistent IP addresses are used. A persistent address remains on the node even if the cluster services are down and the systems have been rebooted.

Figure 2-9 on page 20 shows a common network configuration on a two-node cluster.

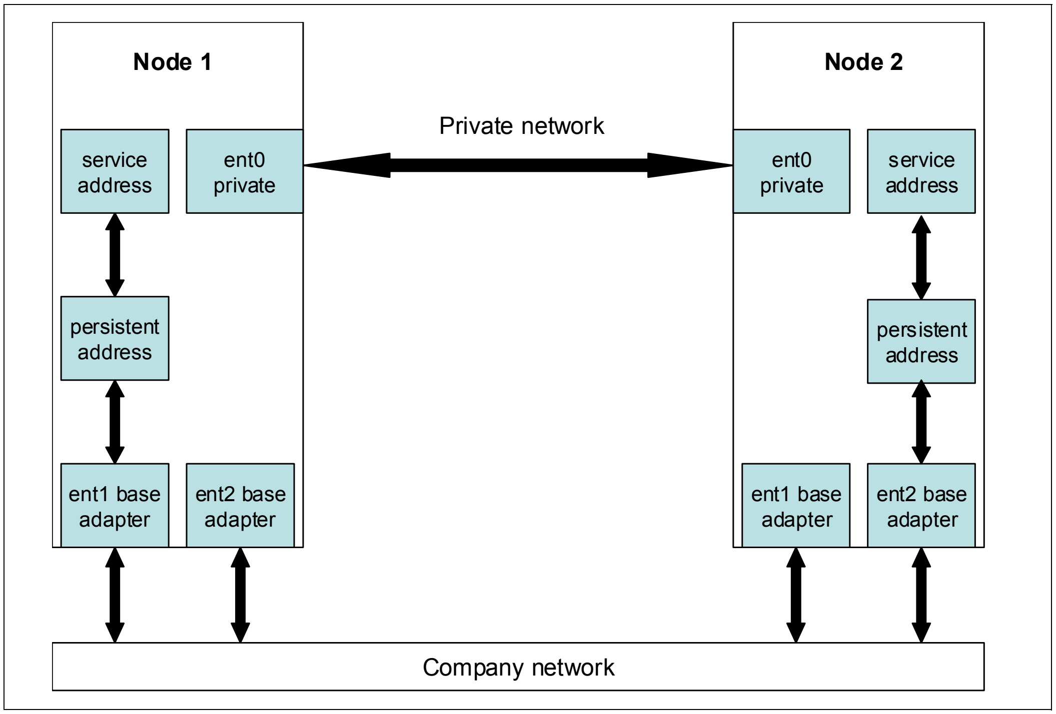


Figure 2-9 Common network configuration in a two-node cluster

PowerHA applications control

To make definitions clearer from the PowerHA cluster perspective, any process or service that is running and providing information to users is called an *application* when it is included in the cluster structure.

Also, PowerHA treats applications use the same approach, as explained previously. Because each application can have specific procedures for startup and shutdown, PowerHA requires specific shell scripts to perform applications' start and stop operations. This PowerHA control structure is called *application controller scripts*.

You need to specify which scripts will be used to start and stop the application services when brought up or down by the PowerHA cluster, as shown in Figure 2-10.

```
                          Add Application Controller Scripts

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

                                                        [Entry Fields]
* Application Controller Name                        [application01]
* Start Script                                       [/fs1/app01_start.ksh]
* Stop Script                                        [/fs1/app01_stop.ksh]
  Application Monitor Name(s)                                               +
  Application startup mode                           [background]           +

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 2-10 Defining application controller scripts by using smitty menus

You can also create *application monitoring methods* for each application. Basically, those are scripts that automatically check the applications to make sure that all application functions are working correctly.

PowerHA applications can be created by using the **`smitty sysmirror`** command and selecting **Cluster Nodes and Networks** → **Cluster Applications and Resources** → **Resources** → **Configure User Applications (Scripts and Monitors)** → **Application Controller Scripts** → **Add Application Controller Scripts** or by using fast path:

```
smitty cm_add_app_scripts
```

> **Note:** Handle application monitoring methods carefully because, normally, a resource group failover (fallover) operation is started when a monitoring script ends with an error. If there is any inconsistency in the scripts, unexpected and unnecessary failover operations might occur.

PowerHA resources and resource groups

Typically, when youare considering purchasing a clustering solution, the main concern is keeping any business-critical application highly available (databases, applications, or middleware).

A *resource* is any component that is required to bring one service application up. Using PowerHA, the resource is able to move from one cluster node to another. A resource can be any of the following components:

- File systems
- Raw devices
- Volume groups
- IP addresses
- NFS shares
- Applications
- Workload partitions (WPARs)
- Custom-defined component

To start one application, a set of these components is usually required, and they need to be grouped together. This logical entity (combined set of resources) in PowerHA is known as a *resource group*.

Figure 2-11 on page 22 shows a sample cluster with shared components (IP address, file systems, volume groups, and so on.).

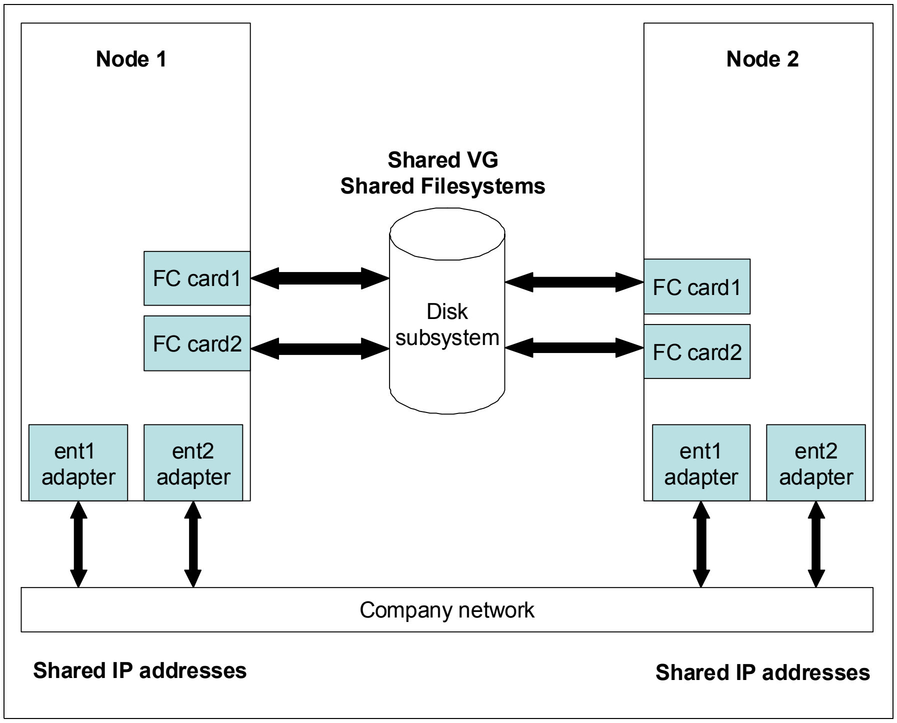


Figure 2-11 A sample cluster with shared components

PowerHA resource group policies

When designing an environment in a PowerHA cluster, you need to plan how you want the cluster to behave when a failure occurs. To make the design task easier, PowerHA uses methods to manage it automatically.

When defining a resource group by using System Management Interface Tool (SMIT), you see the panel shown in Figure 2-12.

```
                           Add a Resource Group

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

                                                     [Entry Fields]
* Resource Group Name                               [rg1]
* Participating Nodes (Default Node Priority)       [node1 node2]         +

  Startup Policy                                     Online On Home Node O> +
  Fallover Policy                                    Fallover To Next Prio> +
  Fallback Policy                                    Fallback To Higher Pr> +

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
```

Figure 2-12 Resource group policies definitions on a smitty SystemMirror menu

Figure 2-12 shows three types of resource group policies that must be configured during cluster implementation.

Startup policy

The first policy to be configured is called the *startup policy*. It defines when and on which node the resource group is brought online when the cluster is started. These are the options for the startup policy:

- **Online on home node only:** When this policy is chosen, the resource group is brought online only on the node called *home node*. This node is the first one from the left in the *Participant Nodes* field. Using Figure 2-12 on page 22 as example, if this policy is chosen, the resource group *rg1* will be online only when *node1* is online.
- **Online on first available node:** When this policy is chosen, the resource group is brought online on the first participant node that comes online. Using Figure 2-12 on page 22 as example, if this policy is chosen, the resource group *rg1* will be online on *node1* if it is the first available node, or it will be online on node2 if that node becomes available first.
- **Online using distribution policy:** When this policy is chosen, the resource groups are brought online through one of these methods:
  - They are distributed, trying to keep only one resource group online on each participant node online (*node-based* resource group distribution)
  - They are distributed, trying to keep only one resource group per node and per network (*network-based* resource group distribution)
- **Online on all available nodes:** When this policy is chosen, the resource group is brought online on all available nodes in the cluster. To avoid data corruption or any kind of application problem, ensure that the components included on the resource group can be used concurrently.

Also, regarding resource group startup, there is a parameter that can be customized called the *settling time*. When using the settling time, any cluster node waits for the configured time to make sure that any other higher-priority node is not about to join the cluster. It is an interesting parameter that can be helpful to use when you have a multiple node cluster, and all of them start simultaneously.

Fallover policy

The second mandatory policy is called the *fallover policy* (or *failover* policy). In a running cluster, this policy defines the behavior of resource groups when the resource group that owns the node fails. These are the options for the *fallover policy*:

1. **Fallover to the next priority node in the list:** When the node that owns an online resource group fails, if the resource group is not online on all available nodes, it is brought online on the next node according to the resource groups *participant nodes* list (Figure 2-12 on page 22).
2. **Fallover using dynamic node priority:** When the node that owns an online resource group fails, the resource group is moved to another node according to the *dynamic node priority* policy that is defined. These policies are based on RSCT variables, such as the node with the most memory available. Keep in mind that if you choose this option without a dynamic node priority policy defined, you will encounter an error when you synchronize a cluster configuration.
3. **Bring offline (on the error node only):** When the node that owns an online resource fails, no fallover action will be taken. If the resource group is online at one node per time, the services will be unavailable until an administrator action. When the resource group is online on all available nodes, the resource will be offline only on the failing node, and the resource continues to work properly on all another nodes.

Fallback policy

The third policy to be configured is called the *fallback policy*. It defines what happens with a resource group when a higher-priority node that experienced a failure joins the cluster. These are the options for the fallback policy:

1. **Fall back to a higher-priority node in the list:** When using this policy, if a higher priority node returns to the cluster from a previous failure, the resource group is brought offline anywhere it is and is brought online on the higher priority node. When using this automatic fallback method, it is important to remember that if there is an intermittent issue on the higher-priority node, the cluster applications start an infinite loop of moves between nodes.
2. **Never fall back:** When using this policy, even if a higher priority node returns from a previous failure, the resource group remains on the lower priority node until a manual resource group move is performed by a cluster administrator. This is an important configuration to be considered when designing the cluster, because it allows a small disruption. The only disruption period is while the resource groups are being moved to next node, and the fallback must be done manually later. But you must consider that the contingency node can be over stressed with more load than it can be designed for.

(Optional) Fallback timer policy

An optional policy that can be configured when creating a resource group is called *fallback timer policy*. Using this policy, you can configure on which specific frequency a fallback operation can be performed. These are the options:

- **Daily:** Fallback operations are performed daily on the hour and date determined by the system administrator.
- **Weekly:** Fallback operations are performed weekly on the day, hour, and time specified by the system administrator. Only one weekday can be chosen.
- **Monthly:** Fallback operations are performed monthly on the day of the month, hour, and time specified by the system administrator. Only one day per month can be chosen.
- **Yearly:** Fallback operations are performed annually on the month, day of the month, hour, and time that are specified by the system administrator. Only a single year date and time can be chosen.

> **Note:** For fallback timer policy configurations, use `smitty sysmirror`, and then select **Cluster Applications and Resources** → **Resource Groups** → **Configure Resource Group Run-Time Policies** → **Configure Delayed Fallback Timer Policies** or use the `smitty cm_timer_menu` fast path.

PowerHA cluster events

Considering all involved components, the PowerHA solution provides ways to monitor almost any part of the cluster structure. Also, according to the output of these monitoring methods, the PowerHA cluster itself takes an automatic action, which can be a notification or even a resource group fallover.

PowerHA allows customization of predefined cluster events and creation of new events. When creating new events, it is important to check first whether there is any standard event that covers the action or situation.

All standard cluster events have their own meanings and functions. Table 2-2 on page 25 lists examples of cluster events.

Table 2-2 Examples of standard cluster events

Event name	Event type	Summary
node_up	Nodes joining or leaving cluster	A **node_up** event starts when a node joins or rejoins the cluster.
node_down	Nodes joining or leaving cluster	A **node_down** event starts when a cluster is not receiving heartbeats from a node. It considers the node gone and starts a node_down event.
network_up	Nodes joining or leaving cluster	A **network_up** event starts when a cluster detects that a network is available and ready for use (for a service IP address activation, forexample).
network_down	Network-related events	A **network_down** event starts when a specific network is not reachable anymore. It can be **network_down_local**, when only a specific node has lost its connectivity for a network, or **network_down_global**, when all nodes have lost connectivity.
swap_adapter	Network-related events	A **swap_adapter** event starts when the interface thathosts one service IP address experiences a failure. If there are other boot networks available on the same node, the swap_adapter event moves the service IP address to another boot interface and refreshesthe network routing table.
fail_interface	Interface-related issues	A **fail_interface** event starts when any node interface experiences a failure. If the interface has no service IP defined, onlythe fail_interface event runs. If the failing interface hosts a service IP address and there is no other boot interface available to host it, an **rg_move** event is triggered.
join_interface	Interface-related issues	A **join_interface** event starts when a boot interface becomes available or when it recovers itself from a failure.
fail_standby	Interface-related issues	A **fail_standby** event starts when a boot interface, hosting no service IP address, faces a failure.

Event name	Event type	Summary
join_standby	Interface-related issues	A **join_standby** event starts when a boot interface becomes available or when it recovers from a failure.
rg_move	Resource group changes	An **rg_move** event starts when a resource group operation from one node to another starts.
rg_up	Resource group changes	An **rg_up** event starts when a resource group is successfully brought online at a node.
rg_down	Resource group changes	An **rg_down** event starts when a resource group is brought offline.

Note: All events have detailed use description in the script files. All standard events are in the /usr/es/sbin/cluster/events directory.

2.2.5 PowerHA cluster configurations

PowerHA provides many possible ways to configure a cluster environmentto make different high availability solutions possible. Some of the possible configurations are listed in this section, with some examples for better understanding of how the solutions works.

Standby configuration

The simplest cluster configuration is when a physical node is running all services for a resource group while the other nodes are idle, ready to host resource group services in case of a main node failure.

Figure 2-13 on page 27 shows that when the sample standby cluster starts, all *DB Prod RG* resource group services are brought online at Node 1. However, Node 2 remains idle with no production service running on it. It is only in the case of a Node 1 failure that the DB Prod RG resource group will be automatically moved to Node 2.

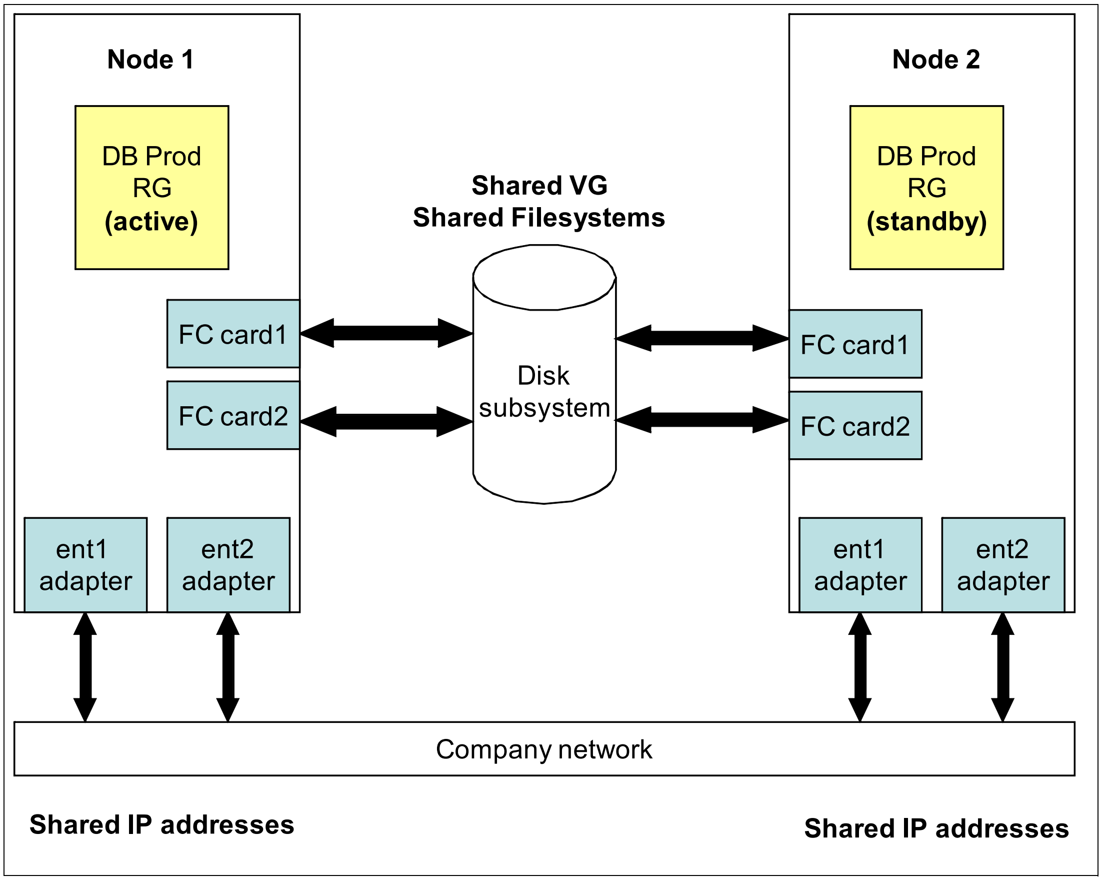


Figure 2-13 Sample standby cluster configuration

Takeover configuration

This allows a more efficient hardware use when all cluster nodes are running parts of the production workload. A takeover configuration can be split into two possible sub-configurations: *One-sided takeover* or *mutual takeover*. Details of these possibilities are shown in Figure 2-14 on page 28 and in Figure 2-15 on page 29.

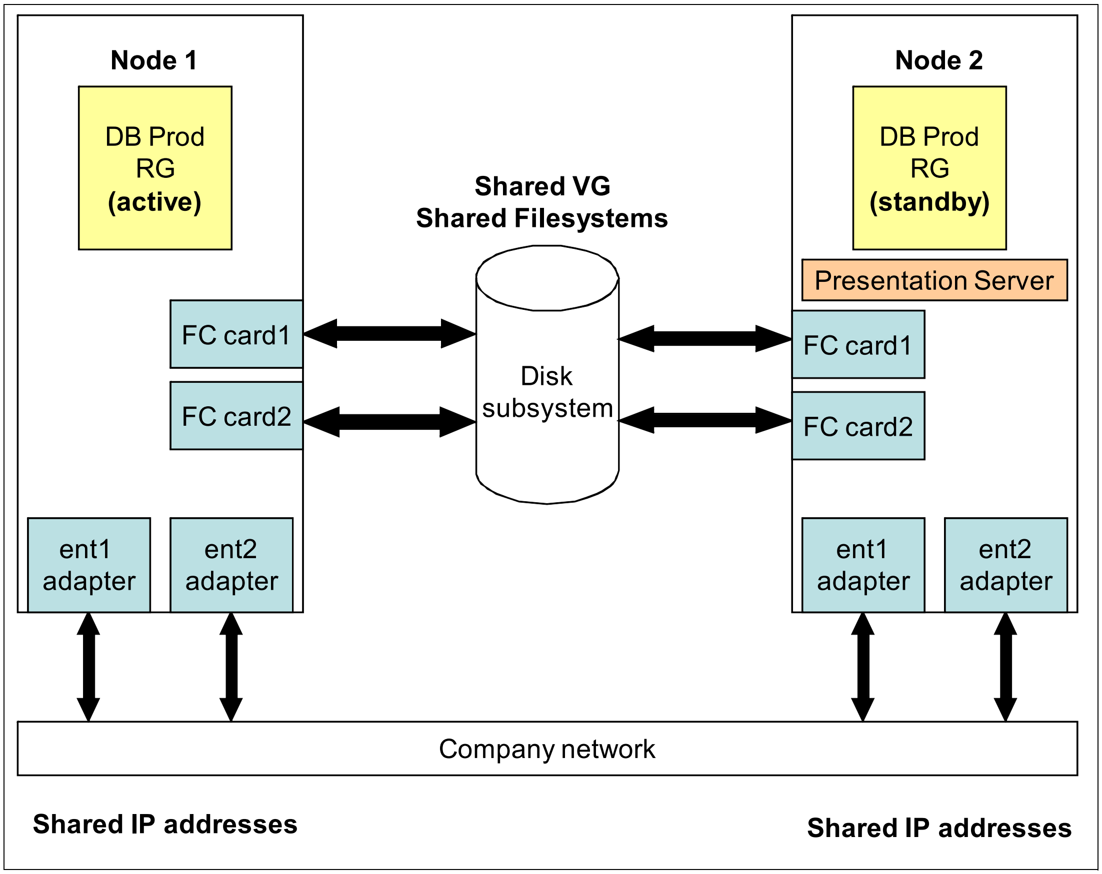


Figure 2-14 Sample one-sided takeover cluster configuration

As shown in Figure 2-14, on a *one-sided takeover* cluster configuration, some application parts are made highly available, for example, being managed by a resource group. In this example, *DB Prod RG* and some application parts run stand-alone, with no high availability behavior running outside of the cluster structure. This means that in a Node 1 failure, its services will be automatically brought online on Node 2. But in a Node 2 failure, its services will remain unavailable until it is manually brought up again in production.

Note: PowerHA does not use the shared disk capability of CAA.

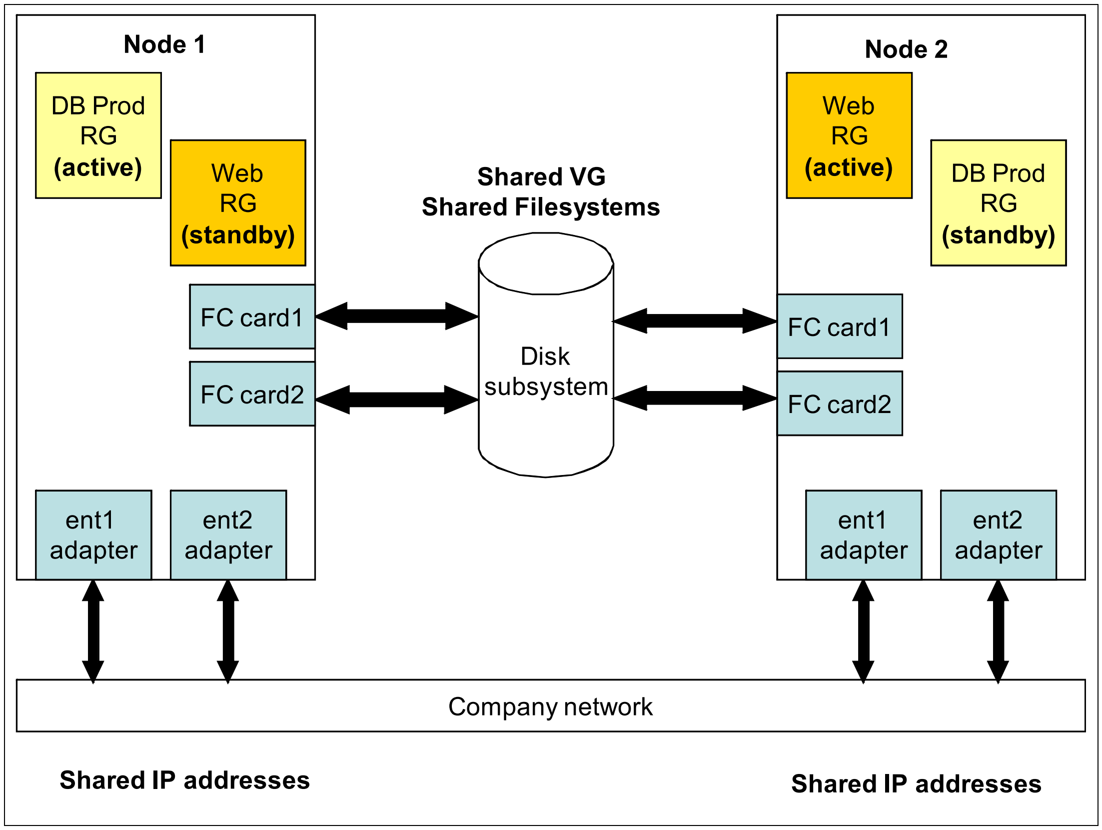


Figure 2-15 Sample mutual takeover cluster configuration

As shown in Figure 2-15, in a *mutual takeover* cluster configuration, all application parts are highly available and managed by resource groups (DB Prod RG and Web RG). When Node 1 has services running in it and that node fails, its services are moved automatically to Node 2. And in a *Node 2 failure*, services will be brought online automatically on Node 1. So, any kind of node crash can be covered by the PowerHA cluster structure with minimal impact to users.

PowerHA cluster single point of control

Sometimes, when managing a cluster environment, some basic administration tasks become harder to perform because of the number of managed clusters or managed nodes. As a result, inconsistencies can appear in customer environments, especially inconsistencies that are related to the LVM structure or user and group ID management.

To avoid these issues, PowerHA provides a way to facilitate administrative tasks on all nodes inside a PowerHA cluster. This is called the *Cluster Single Point of Control (C-SPOC)*.

Using C-SPOC, you can do the following tasks on all cluster nodes:

- Control PowerHA services:startup and shutdown
- Manage cluster resource groups and applications
- Manage cluster nodes communication interfaces
- Manage file collections
- View and manage logs
- Manage AIX user and groups across all cluster nodes
- Perform Logical Volume Manager (LVM)tasks
- Handle IBM General Parallel File System (IBM GPFS) file system tasks
- Open a smitty session on any specific node

Note: Throughout this book, many tasks are performed by using C-SPOC functions to show specific PowerHA features and behaviors. For more information about C-SPOC features and use, see the *PowerHA SystemMirror system management C-SPOC* topic in the IBM Knowledge Center:

http://ibm.co/1s4CRe1

PowerHA SmartAssists

SmartAssists are PowerHA tools that help system administrators include applications in a cluster infrastructure. Using SmartAssists, you can configure the application in a highly available cluster and manage the availability of the application with start and stop scripts.

SmartAssists works specifically with each application, so individual SmartAssist packages must be installed in addition to PowerHA base software to support particular applications. If an application that needs to be included in a cluster environment has no specific SmartAssist product, PowerHA provides a General Application SmartAssist (GASA), which helps include these applications in the clustered environment.

These requirements must be addressed before start using SmartAssists:

- The SmartAssist fileset to be used must be installed on all cluster nodes.
- Before using the SmartAssist tool, a basic cluster must be created by using **smitty** or the IBM System Director interface.
- Before configuring an application inside the cluster by using SmartAssist, you must ensure that the application is already able to run manually with no issues on all cluster nodes.
- We strongly recommend that you configure the application with the SmartAssist on the cluster node where the application is currently running.

There are many SmartAssist versions available, as shown in Figure 2-16 on page 31.

```
          Make Applications Highly Available (Use Smart Assists)

  +--------------------------------------------------------------------------+
  |      Select a Smart Assist From the List of Available Smart Assists      |
  |                                                                          |
  | Move cursor to desired item and press Enter.                             |
  |                                                                          |
  |   DB2 UDB non-DPF Smart Assist     # smass1 smass2                       |
  |   DHCP Smart Assist                # smass1 smass2                       |
  |   DNS Smart Assist                 # smass1 smass2                       |
  |   Lotus Domino smart assist        # smass1 smass2                       |
  |   FileNet P8 Smart Assist          # smass1 smass2                       |
  |   IBM HTTP Server Smart Assist     # smass1 smass2                       |
  |   SAP MaxDB Smart Assist           # smass1 smass2                       |
  |   Oracle Database Smart Assist     # smass1 smass2                       |
  |   Oracle Application Server Smart Assist # smass1 smass2                 |
  |   Print Subsystem Smart Assist     # smass1 smass2                       |
  |   SAP Smart Assist                 # smass1 smass2                       |
  |   Tivoli Directory Server Smart Assist # smass1 smass2                   |
  |   TSM admin smart assist           # smass1 smass2                       |
  |   TSM client smart assist          # smass1 smass2                       |
  |   TSM server smart assist          # smass1 smass2                       |
  |   WebSphere Smart Assist           # smass1 smass2                       |
  |   WebSphere MQ Smart Assist        # smass1 smass2                       |
  +--------------------------------------------------------------------------+
```

Figure 2-16 Smart Assists available in PowerHA 7.1.3

2.3 PowerHA SystemMirror in a virtualized environment

PowerHA SystemMirror supports high availability solutions for virtualized environments in Power System servers. There are special considerations thatare described in this section.

2.3.1 Virtualization in IBM Power Systems

IBM Power Systems, combined with IBM PowerVM, are designed to help IBMclients maximize the returns from IT infrastructure investments. The virtualization capabilities of PowerVM help consolidate multiple workloads. As a result, IBM gives special importance to maintaining higher availability of physical hardware because more than one workload is dependent on the server's uptime. Built on the principles of RAS (reliability, availability and serviceability), Power Systems servers and PowerVM have several components to improve the availability of operating systems.

PowerHA can be used with both virtual and physical devices. It can detect hardware failures on these servers but there are special considerations when you are designing the virtual infrastructure:

- Use a dual Virtual I/O Server (VIOS) setup for redundancy (strongly recommended).
- Configure shared Ethernet adapter fallover.
- Configure the `netmon.cf` file to check the status of the network behind the virtual switch.
- Use multiple paths for network and storage devices (strongly recommended).

These configurations are generic for maintaining high availability in a virtualized server environment.

2.3.2 Important considerations for VIOS

PowerHA 7.1.3 supports a virtualized environment. You can use virtual components, such as virtual Ethernet adapters, virtual SCSI disks, and N-Port ID Virtualization (NPIV).

For cluster nodes that use virtual Ethernet adapters, there are multiple configurations possible for maintaining high availability at the network layer. Consider these suggestions:

- Configure dual VIOS to ensure high availability of virtualized network paths.
- Use the servers that are already configured with virtual Ethernet settings because no special modification is required. For a VLAN-tagged network, the preferred solution is to use SEA fallover; otherwise, consider using the network interface backup.
- One client-side virtual Ethernet interface simplifies the configuration; however, PowerHA might miss network events. For a more comprehensive cluster configuration, configure two virtual Ethernet interfaceson the cluster LPAR to enable PowerHA. Two network interfaces are required by PowerHA to track network changes, similar to physical network cards. It is recommended to have two client-side virtual Ethernet adapters that use different SEAs. This ensures that any changes in the physical network environment can be relayed to the PowerHA cluster using virtual Ethernet adapters, such as in a cluster with physical network adapters.

Theseare a few configurations to explore:

- **Two Ethernet adapters in PowerHA network with no SEA fallover or NIB:** In this configuration, each VIOS provides a virtual network adapter to the client on a separate VLAN. Without SEA fallover or NIB, the redundancy is provided by PowerHA. such as in clusters with physical network adapters.
- **NIB and a single Ethernet adapter in PowerHA network:** This configuration is similar to previous configuration but with NIB on the client side. However, using `netmon.cf` is still recommended.
- **NIB and two Ethernet adapters per PowerHA network:** This configuration is an improvement over the previous configuration. It can provide redundancy and load balancing across VIOS servers. Also, PowerHA can track network events in this scenario.
- **SEA fallover and one virtual Ethernet adapter on the client side:** PowerHA configuration with shared Ethernet adapter fallover is helpful when VLAN tagging is being used. Only one Ethernet adapter exists on the client side, and redundancy is provided by SEA fallover. PowerHA cannot detect network events because there is only a single Ethernet adapter on each cluster node.
- **SEA fallover with two virtual Ethernet adapters in the cluster LPAR:** This is a comprehensive setup that supports VLAN tagging and load sharing between VLANS. Two networks are defined and two virtual Ethernet adapters are configured per network. Dual redundancy is provided with SEA fallover and PowerHA. PowerHA can track network events also.

For more information: Architectural details of some of the possible PowerHA solutions using virtual Ethernet are mentioned in section 3.4.1 of the *IBM PowerHA SystemMirror Standard Edition 7.1.1 for AIX Update*, SG24-8030.

2.3.3 SAN- or FC-based heartbeat configuration in virtualized environment

A storage area network (SAN)-based path is a redundant, high-speed path of communication that is established between the hosts by using the SAN fabric that exists in any data center between hosts. Cluster Aware AIX (CAA) provides an additional heartbeat path over SAN or Fibre Channel (FC) adapters. It is not mandatory to set up a FC- or SAN-based heartbeat path. However, if it is configured, SANComm (sfwcomm, as seen in `lscluster -i` output) provides an additional heartbeat path for redundancy.

> **Important:** You can perform LPM on a PowerHA SystemMirror LPAR that is configured with SAN communication. However, when you use LPM, the SAN communication is not automatically migrated to the destination system. You must configure the SAN communication on the destination system before you use LPM. Full details can be found at:
>
> http://www-01.ibm.com/support/knowledgecenter/SSPHQG_7.1.0/com.ibm.powerha.admngd/ha_admin_config_san.htm

PowerHA SystemMirror 7.1.3 supports SAN-based heartbeat within a site.The SAN heartbeat infrastructure can be created in two ways, depending on the configuration of the nodes that are members of the cluster:

- Using real or physical adapters on cluster nodes and enabling the storage framework capability (sfwcomm device) of the HBAs. Currently, FC and SAS technologies are supported. See “Setting up cluster storage communication” in the IBM Knowledge Center for more information about the HBAs and the required steps to set up the storage framework communication:

  http://ibm.co/1o5IxTv

- In a virtual environment, where the nodes in the clusters are VIO Clients. Enabling the sfwcomm interface requires activating the target mode (the tme attribute) on the real adapters in the VIOS and defining a private virtual LAN (VLAN) with VLAN ID 3358 for communication between the partitions that contain the sfwcomm interface and VIOS. The real adapter on VIOS needs to be a supported HBA.
- Using FC for SAN heartbeat requires zoning of the corresponding FC adapter ports (real FC adapters or virtual FC adapters on VIOS).

Configure two types of zones:

- Heartbeat zones:
  - These contain VIOS physical WWPNs.
  - The VIOS on each machine must be zoned together.
  - The virtual WWPNs of the client LPARs must not be zoned together.
- Storage zones:
  - Contains the LPARs’ virtual WWPNs.
  - Contains the storage controller’s WWPNs.

Steps for creating the zones (or “zoning”):

1. Log in to each of the VIOS (both VIOS on each managed system). Verify that the FC adapters are available. Capture the WWPN information for zoning.
2. From the client LPAR, capture the WWPNs for the *fcsX* adapter.
3. Create the zones on switch fabrics:
   a. Zone the LPARs virtual WWPN to the storage ports on the storage controller that is used for shared storage access.

b. Create the zones that contain VIOS physical ports, which will be used for heartbeats.

Target mode enablement

After the zoning is complete, the next step is to enable the *target mode enabled* (tme) attribute. The tme attribute for a supported adapter is available only when the minimum AIX level for CAA is installed (AIX 6.1 TL6 or later or AIX 7.1 TL0 or later). This needs to be performed on all VIOSes. Follow these configuration steps:

1. Configure the FC adapters for SAN heartbeats on VIOS:

   ```
   # chdev -l fscsiX -a tme=yes
   ```

2. Set dynamic tracking to yes and FC error recovery to fast_fail:

   ```
   # chdev -l fscsiX -a dyntrk=yes -a fc_err_recov=fast_fail
   ```

3. Reboot the VIOS.
4. Repeat steps 1 - 4 for all the VIOSes that serve the cluster LPARs.
5. On the HMC, create a new virtual Ethernet adapter for each cluster LPAR and VIOS. Set the VLAN ID to 3358. Do not put another VLAN ID or any other traffic on this interface.
6. Save the LPAR profile.
7. On the VIO server, run the **cfgmgr** command, and check for the virtual Ethernet and sfwcomm device by using the **lsdev** command:

   ```
   #lsdev -C | grep sfwcomm
   ```

   Command output:

   ```
   sfwcomm0 Available 01-00-02-FF Fibre Channel Storage Framework Communication.

   sfwcomm1 Available 01-01-02-FF Fibre Channel Storage Framework Communication.
   ```

8. On the cluster nodes, run the **cfgmgr** command, and check for the virtual Ethernet adapter and sfwcomm with the **lsdev** command.
9. No other configuration is required at the PowerHA level. When the cluster is configured and running, you can check the status of SAN heartbeat by using the **lscluster -i** command:

   ```
   # lscluster -i sfwcomm
   ```

3

What's new in IBM PowerHA SystemMirror 7.1.3

This chapter covers the following topics:

- New features in Version 7.1.3
- Cluster Aware AIX enhancements
- Embedded hyphen and leading digit support in node labels
- Native HTML report
- Syntactical built-in help
- Applications supported by Smart Assist
- Cluster partition (split and merge policies)

3.1 New features in Version 7.1.3

PowerHA 7.1.3 introduced the following features:

- Unicast-based heartbeat

  The Cluster Aware AIX (CAA) environment does have the option to select IP unicast or IP multicast for heartbeat exchanges.

- Dynamic host name change

  Offers two types for dynamically changing the host name: Temporary or permanent.

  For more about how to use it, see Chapter 10, “Dynamic host name change (host name takeover)” on page 359.

- Cluster split and merge handling policies

  Operator-managed manual failover policy for multisite linked clusters.

- **`clmgr`** enhancements

  The following items are enhancements to the **`clmgr`** command:

  – Embedded hyphen and leading digit support in node labels

    In PowerHA 7.1.3, the node labels can start with a number or have a hyphen as part of the name. For example: *2ndnode* or *first-node*.

    Details are described in section 3.3, “Embedded hyphen and leading digit support in node labels” on page 39

  – Native HTML report

    This is part of the base product. The main benefits are:

    • Contains more cluster configuration information than any other report.
    • Can be scheduled to run automatically via AIX core functionality like `cron`.
    • Portable, so it can send by email without loss of information.
    • Fully translated.
    • Allows for inclusion of a company name or logo into the report header.

    Details are described in section 3.4, “Native HTML report” on page 40

  – Cluster copying

    Allows the administrator to take a snapshot from a fully configured and tested cluster which can then be restored on a new hardware, or LPAR.

  – Syntactical built-in help, with these main features:

    • Lists all possible inputs for an operation.
    • Shows valid groupings.
    • Provides complete required versus optional input information.
    • Provides standard versus verbose modes.

  – Split and merge support

    For **`clmgr`** full split/merge policy control was added.

    Details are described in section 3.5, “Syntactical built-in help” on page 41.

- Cluster Aware AIX (CAA) enhancements:

  – Scalability

    CAA now supports up to 32 nodes.

  – Dynamic host name and IP address support

  – Unicast support (supports IP unicast and IP multicast)

- IBM HyperSwap® enhancements

  The items listed as follows are new to HyperSwap in PowerHA 7.1.3:

  – Active-active sites

    This supports active-active workloads across sites for continuous availability of site level compute and storage outages. It includes support for Oracle RAC long-distance deployment.

  – One node HyperSwap

    Support for the storage HyperSwap for one AIX LPAR. No need for second a node in the cluster.

  – Auto resynchronization of mirroring

    Support for automatic resynchronization of metro mirroring when needed.

  – Node level unmanage mode support

    HyperSwap adapts to the cluster status unmanage of PowerHA and stops HyperSwap for the affected node.

  – Enhanced repository disk swap management

    Administrator can avoid specifying standby disk for repository swap handling.

  – Dynamic policy management support

    Administrator can modify the HyperSwap policies across the cluster. For instance: Expand or delete mirror groups.

  – Enhanced verification and RAS

    New command **`phakedb`** (in kdb) can be used to display important control blocks and data structures.

> **Note:** To implement the HyperSwap functionality with the IBM PowerHA SystemMirror Enterprise Edition 7.1.3 a DS88xx and higher is required.

- Enhancements for PowerHA plug-in for Systems Director

  The major enhancements for the PowerHA plug-in for Systems Director are:

  – Restore snapshot wizard

    There are two ways how to do the restore:

    • Restore snapshot on the same set of nodes where snapshot was captured.
    • Restore snapshot on a different set of nodes from where snapshot was captured.

  – Cluster split/merge support

    Support for splitting or merging cluster nodes

  – Cluster simulator

    It provides a supported, portable demonstration tool and a portable demonstration tool.

- Smart Assist Enhancements for SAP

  The main enhancements for SmartAssist are for SAP environments as follows:

  – Support for SAP instance installation variations supported by SAP

    If more than one SAP instance share the same virtual-IP or VG, SAP SA groups them in a single resource group.

  – Support for local configuration installation for SAP instances

SAP SA supports the local configuration deployment of SAP instances. The local file system or local VGs are not monitored by PowerHA SystemMirror.

– Pure Java stack support

  SAP SA is enhanced to support pure Java stack deployments.

– Multiple SIDs support

  Users can configure SAP instances of different SIDs in same PowerHA SystemMirror cluster. Sharing resources across SIDs (virtual IP or VGs) is not supported.

– SAP configuration tunables customization

  SAP SA monitoring can be tuned by setting its attributes.

– Support for internal/external NFS

  SAP SA discovers/supports both internal and external NFS deployments. External deployments are not monitored by SAP SA.

– Manual configuration enhancements

  Updated support for multiple SIDs deployments, networks, database RG and local resources.

– Updated support for IBM Systems Director

– Resource group configuration enhancements (miscellaneous data, dependencies, etc.) for SAP Instances

– Single app server script (start/stop/monitor) to handle all types of SAP Instances

– Adaptive failover enhancements

– Support for migration from previous versions of PowerHA SystemMirror

– Option to explicitly define dependency with database with SAP instances from smitty

– New logging utility *KLIB_SAP_SA_Logmsg* to enhance logging

– Operator controlled Recovery support

For more information, see Chapter 7, “Smart Assist for SAP 7.1.3” on page 131.

3.2 Cluster Aware AIX enhancements

This section describes the enhancements to Cluster Aware AIX (CAA) in more detail.

3.2.1 Unicast clustering

PowerHA SystemMirror 7.1.3 uses unicast communications by default for heartbeat and messaging between nodes in the cluster. You can choose to use multicast communication. If you use multicast communication, you must verify that the network devices are configured for multicast communication. When you create a cluster that uses multicast communication, PowerHA SystemMirror uses a default multicast IP address for your environment, or you can specify a multicast IP address.

Connectivity for communication must already be established between all cluster nodes. Automatic discovery of cluster information runs by default when you use the initial cluster setup (typical) menus found under the SMIT menu Cluster Nodes and Networks. After you have specified the nodes to add and their established communication paths, PowerHA SystemMirror automatically collects cluster-related information and configures the cluster nodes and networks based on physical connectivity. All discovered networks are added to the cluster configuration.

The cluster configuration is stored in a central repository disk, and PowerHA SystemMirror assumes that all nodes in the cluster have common access to at least one physical volume or disk. This common disk cannot be used for any other purpose such as hosting application data. You specify this dedicated shared disk when you initially configure the cluster.

In PowerHA SystemMirror 7.1.3, a new feature has been added that enables Cluster Aware AIX environment to select IP unicast or IP multicast for heartbeat exchange. This gives additional flexibility during the configuration of the CAA environment.

> **Note:** Unicast is the default for new created clusters, but multicast is the heartbeat exchange mechanism for 7.1 migrated clusters.

> **Note:** For more information on cluster setup, see "Configuring a PowerHA SystemMirror cluster" in the PowerHA SystemMirror 7.1 section of the IBM Knowledge Center:
>
> `http://ibm.co/1qhVwQp`

3.2.2 Dynamic host name change support

The host name can now be changed dynamically. The change can be either permanent or temporary (reset on a reboot) based on how the change was made. CAA supports both option and updates the node name with the current host name. However, if the node name is set to a host name that is not resolvable, the node name is not changed.

Both of these options can be used to set the host name dynamically. For more details about how to use it see Chapter 10, "Dynamic host name change (host name takeover)" on page 359.

3.2.3 Scalability enhancements

CAA can now support up to 32 nodes (PowerHA supports up to 16 nodes in the cluster in one or two sites configurations).

CAA updates also include more capabilities:

- SAN comm check utility
- Repository disk recovery

For the complete details of the Cluster Aware AIX (CAA) updates, see 2.2.2, "Cluster Aware AIX (CAA)" on page 14.

3.3 Embedded hyphen and leading digit support in node labels

The following historical restrictions for node names have been removed in the PowerHA release 7.1.3:

- No embedded hyphens in the name
- No leading digits in the name

Node names as shown in Example 3-1 on page 40 are now valid and may be used.

Example 3-1 Node names

```
first-node
300node
```

3.4 Native HTML report

The cluster manager command, **clmgr,** is now able to generate native HTML output. The output is similar to that from IBM Systems Director plug-in but has no external requirements. It is available in the base product starting with release 7.1 TL3. Consider these benefits and limitations:

- Benefits:
  - Contains more cluster configuration information than any previous native report.
  - Can be scheduled to run automatically via AIX core abilities (for example, cron).
  - Portable. Can be emailed without loss of information.
  - Fully translated.
  - Allows for inclusion of a company name or logo into the report header.
- Limitations:
  - Per-node operation. No centralized management.
  - Relatively modern browser required for tab effect.
  - Only officially supported on Internet Explorer and Firefox.

The output can be generated for the whole cluster configuration or limited to special configuration items such as:

- nodeinfo
- rginfo
- lvinfo
- fsinfo
- vginfo
- dependencies

See the top part of Figure 3-1 for a complete cluster sample output.

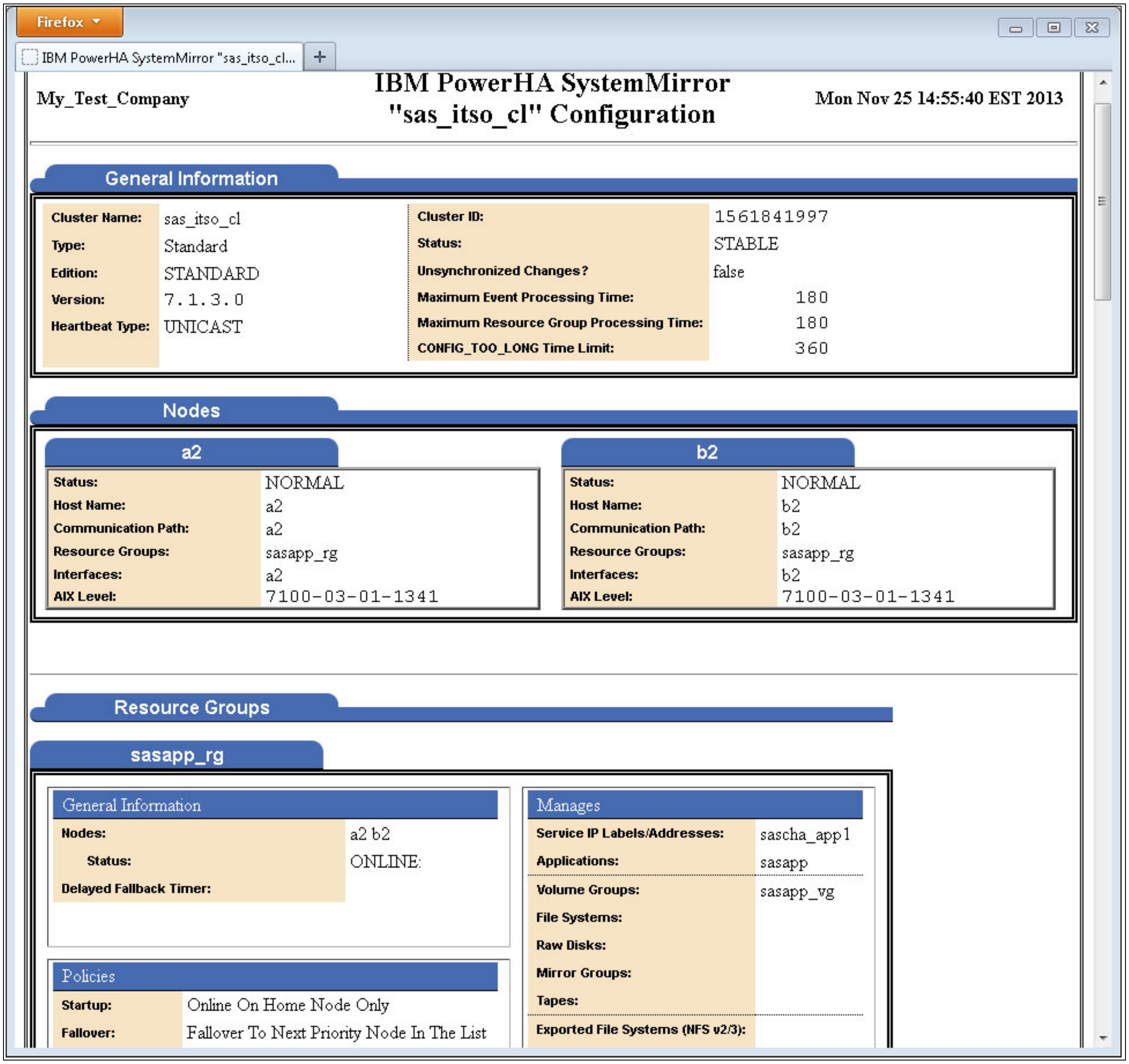


Figure 3-1 Complete cluster output example

Also, the availability report can now be generated in HTML format.

> **Tip:** For a full list of available options, use the **`clmgr`** built-in help function:
>
> `clmgr view report -h`

3.5 Syntactical built-in help

In the new PowerHA 7.1.3 release, the **`clmgr`** provides true syntactical help for every operation. The command lists all possible inputs for an operation and is also showing valid groupings. The command also shows the complete required and the optional input information. A standard and a verbose mode are available for this help.

Help requirements:

- The **clmgr** man page must be installed.
- Either the "more" or "less" pager must be installed.

If verbose mode fails, the standard mode is attempted. If the standard mode fails, the original simple help is displayed. See Example 3-2 for a syntactical help output of the **clmgr** command view report.

Example 3-2 clmgr command help output

```
[a2]:/ >clmgr -h view report

clmgr view report [<report>] \
             [ FILE=<PATH_TO_NEW_FILE> ] \
             [ TYPE={text|html} ]

 clmgr view report cluster \
             TYPE=html \
             [ FILE=<PATH_TO_NEW_FILE> ] \
             [ COMPANY_NAME="<BRIEF_TITLE>" ] \
                 [ COMPANY_LOGO="<RESOLVEABLE_FILE>" ]

 clmgr view report {nodeinfo|rginfo|lvinfo|
                   fsinfo|vginfo|dependencies} \
             [ TARGETS=<target>[,<target#2>,<target#n>,...] ] \
             [ FILE=<PATH_TO_NEW_FILE> ] \
             [ TYPE={text|html} ]

 clmgr view report availability \
             [ TARGETS=<application>[,<app#2>,<app#n>,...] ] \
             [ FILE=<PATH_TO_NEW_FILE> ] \
             [ TYPE={text|html} ] \
             [ BEGIN_TIME="YYYY:MM:DD" ] \
             [ END_TIME="YYYY:MM:DD" ]

 view => cat
```

3.6 Applications supported by Smart Assist

Table 3-1 on page 43 lists the applications that were supported when the PowerHA 7.1.3 release was announced. To get the latest information about the supported versions, check "Smart Assists for PowerHA SystemMirror" in the IBM Knowledge Center (PowerHA SystemMirror 7.1 > Smart Assists for PowerHA SystemMirror):

http://ibm.co/YMuo56

Table 3-1 Smart Assist applications

Application	Supported version
Oracle Database	11gR2
Oracle Application Server	6.1
IBM HTTP Server	6.1
IBM WebSphere® Application Server	6.1
IBM DB2®	10.1
IBM WebSphere MQ	7.0
Tivoli Storage Management - Admin	6.2
Tivoli Storage Management - Server	6.2
Tivoli Storage Management - Client	6.2
IBM Security Directory Server	6.3
IBM Lotus® Domino® Server	9
SAP Net Weaver	7.3
SAP Live Cache	7.9
Max DB	7.8
IBM FileNet® P8	4.5
Print subsystem	AIX V6.1 & AIX V7.1
DNS	AIX V6.1 & AIX V7.1
DHCP	AIX V6.1 & AIX V7.1

3.7 Cluster partition (split and merge policies)

Cluster nodes communicate with each other over communication networks. PowerHA SystemMirror supports different types of definitions for sites and site-specific policies for high availability disaster recovery (HADR). You can define multiple sites in both PowerHA SystemMirror Standard Edition for AIX and PowerHA SystemMirror Enterprise Edition for AIX.

A cluster partition is when failures isolate a subset of cluster nodes from the rest of the cluster, for example:

- Failure of the links between sites
- Multiple failures within a site (requires failures of Ethernet, SAN, and repository access)
- The process of partitioning is referred to as a *split*
- The isolated subset of nodes is referred to as a *partition*

The following are definitions to remember for the split and merge policies:

Split policy
: A cluster split event can occur between sites when a group of nodes cannot communicate with the remaining nodes in a cluster. For example, in a linked cluster, a split occurs if all communication links between the two sites fail. A cluster split event splits the cluster into two or more partitions.

Merge policy	Depending on the cluster split policy, the cluster might have two partitions that run independently of each other. You can use PowerHA SystemMirror Version 7.1.2 or later to configure a merge policy that allows the partitions to operate together again after communications are restored between the partitions. See Table 3-2 on page 45.
Tie breaker option	You can use the tie breaker option to specify a SCSI disk that is used by the split and merge policies (refer to table Table 3-2 on page 45). A tie breaker disk is used when the sites in the cluster can no longer communicate with each other. This communication failure results in the cluster splitting the sites into two, independent partitions. If failure occurs because the cluster communication links are not responding, both partitions attempt to lock the tie breaker disk. The partition that acquires the tie breaker disk continues to function while the other partition reboots or has cluster services restarted, depending on the selected action plan. The disk that is identified as the tie breaker must be accessible to all nodes in the cluster. When partitions that were part of the cluster are brought back online after the communication failure, they must be able to communicate with the partition that owns the tie breaker disk. If a partition that is brought back online cannot communicate with the tie breaker disk, it will not join the cluster. The tie breaker disk is released when all nodes in the configuration rejoin the cluster.

3.7.1 Configuring split and merge policies

You can use the SMIT interface to configure split and merge policies.

When you use the SMIT interface in PowerHA SystemMirror 7.1.3 to configure split and merge policies, you must stop and restart cluster services on all nodes in the cluster. You can stop a cluster service before you complete the following steps, or you can configure split and merge policies in an active cluster and restart cluster services after verification and resynchronization of the cluster is complete.

To configure a split and merge policy in PowerHA SystemMirror 7.1.3, or later, complete the following steps:

1. From the command line, enter `smitty sysmirror`.
2. In the SMIT interface, select **Custom Cluster Configuration** → **Cluster Nodes and Networks** → **Initial Cluster Setup (Custom)** → **Configure Cluster Split and Merge Policy**, and press Enter.
3. Complete the fields as shown in Table 3-2 on page 45, and press Enter.

> **Note:** The *Manual choice option* was added in PowerHA 7.1.3

Table 3-2 Configure Cluster Split and Merge Policy fields

Field	Description
Split handling policy	Select **None**, the default setting, for the partitions to operate independently of each other after the split occurs. Select **Tie breaker** to use the disk that is specified in the **Select tie breaker** field after a split occurs. When the split occurs, one site wins the SCSI reservation on the tie breaker disk. The site that losses the SCSI reservation uses the recovery action that is specified in the policy setting. **Note:** If you select the **Tie breaker** option in the **Merge handling policy** field, you must select **Tie breaker** for this field. Select **Manual** to wait for manual intervention when a split occurs. PowerHA SystemMirror does not perform any actions on the cluster until you specify how to recover from the split. **Note:** If you select the **Manual** option in the **Merge handling policy** field, you must select **Manual** for this field.
Merge handling policy	Select **Majority** to choose the partition with the highest number of nodes the as primary partition. Select **Tie breaker** to use the disk that is specified in the **Select tie breaker** field after a merge occurs. **Note:** If you select the **Tie breaker** option in the **Split handling policy** field, you must select **Tie breaker** for this field. Select **Manual** to wait for manual intervention when a merge occurs. PowerHA SystemMirror does not perform any actions on the cluster until you specify how to handle the merge.
Split and merge action plan	Select Reboot to **reboot** all nodes in the site that does not win the tie breaker.
Select tie breaker	Select an iSCSI disk or a SCSI disk that you want to use as the tie breaker disk.

Field	Description
Manual choice option	**Notify Method** This is a method to be invoked in addition to a message to `/dev/console` to inform the operator of the need to choose which site will continue after a split or merge. The method is specified as a path name, followed by optional parameters. When invoked, the last parameter will be either "split" or "merge" to indicate the event. **Notify Interval** The frequency of the notification time, in seconds, between the message to inform the operator of the need to choose which site will continue after a split or merge. The supported values are between 10 and 3600. **Maximum Notifications** This is the maximum number of times that PowerHA SystemMirror will prompt the operator to chose which site will continue after a split or merge. The default, blank, is infinite. Otherwise, the supported values are between 3 and 1000. However, this value *cannot* be blank when a surviving site is specified. **Default Surviving Site** If the operator has not responded to a request for a manual choice of surviving site on a "split" or "merge," this site is allowed to continue. The other site takes the action chosen under "Action Plan." The time that the operator must respond is "Notify Interval" times "Maximum Notifications+1." **Apply to Storage Replication Recovery** determines if the manual response on a split also applies to those storage replication recovery mechanisms that provide an option for "Manual" recovery. If "Yes" is selected, then the partition that was selected to continue on a split will proceed with takeover of the storage replication recovery. This *cannot* be used with DS8k and IBM XIV® replication is used.

4. Verify that all fields are correct and press Enter.
5. Verify and synchronize the changes across the cluster.

Note: You can use the SMIT interface to configure split and merge policies in PowerHA SystemMirror 7.1.2 or earlier as shown in the following website:

`http://ibm.co/1tXTdEa`

3.7.2 Responding to a cluster that uses a manual split merge policy

If your site goes offline and you have the manual choice option specified for split policy the default response is to send a notification to the console of the surviving site node as shown in Figure 3-2 on page 47. This can be responded to via the console terminal as explained in the notification. It also can be specified via the IBM Systems Director Console.

```
[shanley:root] / # May 26 13:46:11 shanley local0:crit clstrmgrES[12648580]: Mon May
26 13:46:11 Removing 5 from ml_idx
A cluster split has been detected.
You must decide if this side of the partitioned cluster is to continue.
To have it continue, enter

        /usr/es/sbin/cluster/utilities/cl_sm_continue

To have the recovery action - Reboot - taken on all nodes on this partition, enter

        /usr/es/sbin/cluster/utilities/cl_sm_recover
```

Figure 3-2 Manual operator response prompt upon site split

To manually respond to a cluster that goes offline and uses a split policy or a merge policy, using the IBM Systems Director console perform the following:

1. Log in to the IBM Systems Director console.
2. On the Welcome page, click the **Plug-ins** tab and select **PowerHA SystemMirror Management**.
3. In the Cluster Management section, click **Manage Clusters**.
4. Right-click the cluster that you do *not* want to use a split policy or a merge policy, and select **Recovery** → **Manual** response to cluster split or merge.
5. Select the site that recovers the cluster, and click **OK**.

4

Migration

This chapter covers the most common migration scenarios from IBM PowerHA 6.1 or PowerHA 7.1.x to PowerHA 7.1.3. It includes the following topics:

4.1 Introduction

This chapter presents a detailed view of the various migration options to help you determine the most appropriate migration path.

> **Note:** This chapter does not cover migration from High Availability Cluster Multi-Processing (IBM HACMP™) 5.5 or earlier versions. See 4.4.1, “Legacy rolling migrations to PowerHA SystemMirror 7.1.3” on page 53 for more information on how to migrate from earlier releases of PowerHA (HACMP).

The success of a migration depends on careful planning. There are important items to keep in mind before starting a migration:

- Create a backup of rootvg from all nodes in the cluster.
- Save the existing cluster configuration.

If necessary, save all custom user scripts:

- Application scripts
- Monitoring scripts

4.2 PowerHA SystemMirror 7.1.3 requirements

This section explains the software and hardware requirements for PowerHA SystemMirror 7.1.3.

4.2.1 Software requirements

The following are the software requirements:

- IBM AIX 6.1 TL9 SP1
- IBM AIX 7.1 TL3 SP1

Migrating from PowerHA SystemMirror 6.1 or earlier requires the installation of the following AIX filesets:

- bos.cluster.rte
- bos.ahafs
- bos.clvm.enh
- devices.common.IBM.storfwork.rte

> **Note:** These filesets are in the base AIX media.

4.2.2 Hardware requirements

IBM systems that run IBM POWER5, POWER6®, or POWER7® technology-based processors, including the following systems:

- IBM Power Systems
- IBM System p
- IBM System p5®
- IBM eServer p5
- eServer pSeries

Hardware requirements for the storage framework communications

When this book was written, the following adapters were supported by Cluster Aware AIX (CAA) for use as sfwcom CAA adapters:

- 4 GB Single-Port Fibre Channel PCI-X 2.0 DDR Adapter (FC 1905; CCIN 1910)
- 4 GB Single-Port Fibre Channel PCI-X 2.0 DDR Adapter (FC 5758; CCIN 280D)
- 4 GB Single-Port Fibre Channel PCI-X Adapter (FC 5773; CCIN 5773)
- 4 GB Dual-Port Fibre Channel PCI-X Adapter (FC 5774; CCIN 5774)
- 4 Gb Dual-Port Fibre Channel PCI-X 2.0 DDR Adapter (FC 1910; CCIN 1910)
- 4 Gb Dual-Port Fibre Channel PCI-X 2.0 DDR Adapter (FC 5759; CCIN 5759)
- 4-Port 8 Gb PCIe2 FH Fibre Channel Adapter (FC 5729)
- 8 Gb PCI Express Dual Port Fibre Channel Adapter (FC 5735; CCIN 577D)
- 8 Gb PCI Express Dual Port Fibre Channel Adapter 1Xe Blade (FC 2B3A; CCIN 2607)
- 3 Gb Dual-Port SAS Adapter PCI-X DDR External (FC 5900 and 5912; CCIN 572A)

For more information, see these IBM Knowledge Center topics:

- "Cluster communication"

  http://ibm.co/1kbEXYC
- "Setting up cluster storage communication" in the IBM Knowledge Center:

  http://ibm.co/1o05BEJ

Also check this APAR:

IV03643: DOC: CAA VLAN REQUIREMENTS FOR SAN COMMUNICATIONS

http://www.ibm.com/support/docview.wss?uid=isg1IV03643

4.2.3 Deprecated features

Starting with PowerHA SystemMirror 7.1, the following features are no longer available:

1. IP address takeover (IPAT) via IP replacement
2. Locally administered address (LAA) for hardware MAC address takeover (HWAT)
3. Heartbeat over IP aliases
4. The following IP network types:
   - ATM
   - FDDI
   - Token Ring
5. The following point-to-point (non-IP) network types:
   - RS232
   - TMSCSI
   - TMSSA
   - Disk heartbeat (diskhb)
   - Multinode disk heartbeat (mndhb)
6. Two-node configuration assistant

7. WebSMIT (replaced by the IBM Systems Director plug-in, Enterprise Edition only)

   Although PowerHA Enterprise Edition was never supported with WebSMIT, PowerHA SystemMirror Enterprise Edition 7.1.2 and later is supported with the IBM Systems Director plug-in.

> **Important:** If your cluster is configured with any of the features listed in points 1 through 4 (above), your environment cannot be migrated. You must either change or remove the features before migrating, or simply remove the cluster and configure a new one with the new version of PowerHA.

4.2.4 Migration options

The following terms and definitions are key ones to know for migrating:

Offline
: A migration type where PowerHA is brought offline on all nodes before performing the migration. During this time, the resources are not available.

Rolling
: A migration type from one PowerHA version to another during which cluster services are stopped one node at a time. That node is upgraded and reintegrated into the cluster before the next node is upgraded. It requires little downtime, mostly because the resources are moved between nodes while each node is being upgraded.

Snapshot
: A migration type from one PowerHA version to another during which you take a snapshot of the current cluster configuration, stop cluster services on all nodes, install the preferred version of PowerHA SystemMirror, and then convert the snapshot by running the **`clconvert_snapshot`** utility. Then, restore the cluster configuration from the converted snapshot.

Non-disruptive
: A node can be *unmanaged*, which allows all resources on that node to remain operational when cluster services are stopped. This generally can be used when applying service packs to the cluster. This option does *not* apply when migrating to version 7.1.x from a prior version.

> **Important:** If nodes in a cluster are running two different versions of PowerHA, the cluster is considered to be in a *mixed cluster state*. A cluster in this state does not support any configuration changes until all of the nodes have been migrated. It is highly recommended to complete either the rolling or non-disruptive migration as soon as possible to ensure stable cluster functionality.

> **Tip:** After migration is finished, the following line is added to the `/etc/syslog.conf` file:
>
> `*.info /var/adm/ras/syslog.caa rotate size 1m files 10`
>
> It is recommended to enable verbose logging by adding the following line:
>
> `*.debug /tmp/syslog.out rotate size 10m files 10`
>
> Then, issue a **`refresh -s syslogd`** command. This provides valuable information if troubleshooting is required.

4.2.5 AIX Technology Level (TL) equivalence table

Table 4-1 shows the AIX Technology Level equivalence for AIX 7.1 compared to AIX 6.1. The reason to know this is because CAA should share the same code across different AIX versions.

Table 4-1 AIX TL level equivalence

AIX 6,1 TL7	is equivalent to	AIX 7.1 TL1	bos.cluster.rte 6.1.7.XX or 7.1.1.XX
AIX 6.1 TL8	is equivalent to	AIX 7.1 TL2	bos.cluster.rte 6.1.8.XX or 7.1.2.XX
AIX 6.1 TL9	is equivalent to	AIX 7.1 TL3	bos.cluster.rte 6.1.9.XX or 7.1.3.XX

4.3 clmigcheck explained

A migration from PowerHA 6.1 to PowerHA 7.1.3 requires invoking the **`/usr/sbin/clmigcheck`** utility, which is part of bos.cluster.rte (CAA).

The first purpose of this utility is to validate the existing PowerHA cluster configuration. The tool detects deprecated features, such as the *network disk heartbeat*, so you can decide to either remove it before the migration or let the migration protocol remove it when the migration is being finished.

The second purpose of this utility is to obtain the necessary information to create the underlying CAA cluster.

The utility prompts for the following user inputs:

- The CAA disk repository
- The use of either unicast or multicast
- In case of multicast, an optional Multicast IP address

> **Note:** The last node in the cluster to run **`/usr/sbin/clmigcheck`** creates the underlying CAA cluster.

4.4 Migration options

This section further describes these migrations options:

- Rolling migration
- Offline migration
- Snapshot migration
- Non-disruptive migration

4.4.1 Legacy rolling migrations to PowerHA SystemMirror 7.1.3

Migration from before the 6.1 release (5.4.1 or 5.5) to v7.1.3 via rolling migration requires a two-stage migration:

1. Migrate to PowerHA 6.1.

2. Migrate to PowerHA 7.1.3 from 6.1.

See the example for migrating from PowerHA 6.1 to 7.1.3 in the next section, 4.4.2, “Rolling migration from PowerHA SystemMirror 6.1 to PowerHA SystemMirror 7.1.3 (AIX 7.1 TL3 or 6.1 TL9)” on page 54.

4.4.2 Rolling migration from PowerHA SystemMirror 6.1 to PowerHA SystemMirror 7.1.3 (AIX 7.1 TL3 or 6.1 TL9)

The cluster configuration in this migration example consists of the following components:

- AIX 7.1 TL3 SP0
- PowerHA 6.1 SP12
- Two node cluster and a single resource group

> **Note:** You might also find it helpful to watch the “PowerHA v6.1 to v7.1.3 Rolling Migration” demo on YouTube:
>
> https://www.youtube.com/watch?v=MaPxuK4poUw

Example 4-1 shows that the cluster topology includes a disk heartbeat network. This type of network is deprecated, and it is automatically removed when the very last node starts cluster services.

Example 4-1 Cluster information

```
#/usr/es/sbin/cluster/utilities/cllsif
Adapter              Type       Network    Net Type   Attribute  Node       IP
Address       Hardware Address Interface Name   Global Name      Netmask
Alias for HB Prefix Length

hdisk5_01            service    net_diskhb_01 diskhb     serial     hacmp37
/dev/hdisk5                       hdisk5
node37s1             boot       net_ether_01 ether      public     hacmp37
10.1.1.37                        en2                       255.255.0
24
node37s3             boot       net_ether_01 ether      public     hacmp37
10.1.3.37                        en4                       255.255.0
24
node37s2             boot       net_ether_01 ether      public     hacmp37
10.1.2.37                        en3                       255.255.0
24
node37s4             boot       net_ether_01 ether      public     hacmp37
10.1.4.37                        en5                       255.255.0
24
ha37a1               service    net_ether_01 ether      public     hacmp37
192.168.1.37                                               255.255.0
24
hdisk5_02            service    net_diskhb_01 diskhb     serial     hacmp38
/dev/hdisk5                       hdisk5
node38s3             boot       net_ether_01 ether      public     hacmp38
10.1.3.38                        en4                       255.255.0
24
```

```
node38s1            boot       net_ether_01 ether      public     hacmp38
10.1.1.38                       en2                       255.255.0
24
node38s2            boot       net_ether_01 ether      public     hacmp38
10.1.2.38                       en3                       255.255.0
24
ha37a1              service    net_ether_01 ether      public     hacmp38
192.168.1.37                                              255.255.0
24
```

Rolling migration from PowerHA 6.1 to PowerHA 7.1.3

This migration requires the following steps:

1. Stop cluster services (**smitty clstop**) on node hacmp37 (the first node to be migrated) with the option to Move Resource Groups.
2. Ensure that the values for the ODM stanza HACMPnode COMMUNICATION_PATH match the AIX **hostname** output and the AIX /etc/hosts resolution, as shown in Example 4-2.

Example 4-2 Checking the ODM stanza values

```
# odmget -q "object = COMMUNICATION_PATH" HACMPnode

HACMPnode:
        name = "hacmp37"
        object = "COMMUNICATION_PATH"
        value = "hacmp37"
        node_id = 1
        node_handle = 1
        version = 15

HACMPnode:
        name = "hacmp38"
        object = "COMMUNICATION_PATH"
        value = "hacmp38"
        node_id = 2
        node_handle = 2
        version = 15

#[hacmp37] hostname
hacmp37

#[hacmp38] hostname
hacmp38

#[hacmp37] host hacmp37
hacmp37 is 9.3.44.37,  Aliases:   hacmp37.austin.ibm.com

#[hacmp38] host hacmp38
hacmp38 is 9.3.44.38,  Aliases:   hacmp38.austin.ibm.com
```

Note: If the value of COMMUNICATION_PATH does not match the AIX hostname output, **/usr/sbin/clmigcheck** displays the following error message:

```
------------[ PowerHA System Mirror Migration Check ]-------------

ERROR: Communications Path for node hacmp37 must be set to hostname

Hit <Enter> to continue
```

This error requires user intervention to correct the environment before proceeding with the migration.

3. Verify that all nodes' host names are included in /etc/cluster/rhosts:

```
# cat /etc/cluster/rhosts
hacmp37
hacmp38
```

4. Refresh the PowerHA cluster communication daemon, **clcomd**.

```
#refresh -s clcomd
```

5. Run the command **/usr/sbin/clmigcheck** and follow the steps shown in Example 4-3.

Example 4-3 Running the /usr/sbin/clmigcheck tool

```
# /usr/sbin/clmigcheck

------------[ PowerHA System Mirror Migration Check ]-------------

Please select one of the following options:
1       = Check ODM configuration.
2       = Check snapshot configuration.
3       = Enter repository disk and IP addresses.
Select one of the above,"x"to exit or "h" for help:

<select 1>

------------[ PowerHA System Mirror Migration Check ]-------------

CONFIG-WARNING: The configuration contains unsupported hardware:  Disk
Heartbeat network.  The PowerHA network name is net_diskhb_01.  This will be
removed from the configuration during the migration to PowerHA System Mirror
7.1.

Hit <Enter> to continue

< Enter >


------------[ PowerHA System Mirror Migration Check ]-------------
The ODM has no unsupported elements.
Hit <Enter> to continue

< Enter >

------------[ PowerHA System Mirror Migration Check ]-------------
```

```
Please select one of the following options:
1       = Check ODM configuration.
2       = Check snapshot configuration.
3       = Enter repository disk and IP addresses.
Select one of the above,"x"to exit or "h" for help:

< Select 3 >

------------[ PowerHA System Mirror Migration Check ]-------------

Your cluster can use multicast or unicast messaging for heartbeat.
Multicast addresses can be user specified or default (i.e. generated by AIX).
Select the message protocol for cluster communications:


        1       = DEFAULT_MULTICAST
        2       = USER_MULTICAST
        3       = UNICAST

Select one of the above or "h" for help or "x" to exit:
```

6. Per Example 4-3 on page 56, choose one of the following CAA heartbeat mechanisms:
   - 1 DEFAULT MULTICAST

     CAA will automatically assign a cluster Multicast IP address.
   - 2 USER MULTICAST

     User will assign a cluster Multicast IP address.
   - 3 UNICAST

     The unicast mechanism was introduced in PowerHA SystemMirror 7.1.3. Select this option if the cluster network environment does not support multicast.

   Example 4-4, as part of the migration steps, shows the selection of the repository disk.

Example 4-4 Migration steps, selecting the repository disk

```
------------[ PowerHA System Mirror Migration Check ]-------------

Select the disk to use for the repository

        1       = 000262ca102db1a2(hdisk2)
        2       = 000262ca34f7ecd9(hdisk5)

Select one of the above or "h" for help or "x" to exit:


< Select the "Disk Repository" >


< Select "x" then "y" to exit >
```

Note: The following warning message always appears when UNICAST has been selected (if a repository disk has been assigned, the message can be ignored):

```
------------[ PowerHA System Mirror Migration Check ]-------------

You have requested to exit clmigcheck.
Do you really want to exit? (y) y

Note - If you have not completed the input of repository disks and multicast
IP addresses, you will not be able to install PowerHA System Mirror.
Additional details for this session may be found in
/tmp/clmigcheck/clmigcheck.log.
```

7. Install all of the PowerHA 7.1.3 filesets (use **`smitty update_all`**).
8. Start cluster services (**`smitty clstart`**).
9. Check hacmp37 node information (lssrc -ls clstrmgrES). The output of the **`lssrc -ls clstrmgrES`** command on node hacmp37 is shown in Example 4-5.

Example 4-5 hacmp37 node information

```
# lssrc -ls clstrmgrES
Current state: ST_STABLE
sccsid = "@(#)36 1.135.1.118
src/43haes/usr/sbin/cluster/hacmprd/main.C,hacmp.pe,61haes_r713,1343A_hacmp713
10/21/"
build = "Oct 31 2013 13:49:41 1344B_hacmp713"
i_local_nodeid 0, i_local_siteid -1, my_handle 1
ml_idx[1]=0     ml_idx[2]=1
There are 0 events on the Ibcast queue
There are 0 events on the RM Ibcast queue
CLversion: 11   <--- This means the migration still in progress !!!
local node vrmf is 7130
cluster fix level is "0"
```

10. On hacmp38, stop cluster services with the **Move Resource Groups** option, and move them over to **hacmp37**.
11. Verify that all nodes' hostnames are included in `/etc/cluster/rhosts`:

```
# cat /etc/cluster/rhosts
hacmp37
hacmp38
```

12. Refresh the PowerHA cluster communication daemon **clcomd**:

```
#refresh -s clcomd
```

13. Run **`/usr/sbin/clmigcheck`** on node hacmp38, as shown in Example 4-6 on page 59.

Example 4-6 Running /usr/sbin/clmigcheck on node hacmp38 output

```
# /usr/sbin/clmigcheck
Verifying clcomd communication, please be patient.
Verifying multicast IP communication, please be patient.
Verifying IPV4 multicast communication with mping.
clmigcheck: Running
/usr/sbin/rsct/install/bin/ct_caa_set_disabled_for_migration on each node in
the cluster

Creating CAA cluster, please be patient.


< then on the next screen >

------------[ PowerHA System Mirror Migration Check ]-------------

About to configure a 2 node CAA cluster, this can take up to 2 minutes.

Hit <Enter> to continue


------------[ PowerHA System Mirror Migration Check ]-------------

You can install the new version of PowerHA System Mirror.


Hit <Enter> to continue
```

14. Check for CAA cluster on both nodes as shown in Example 4-7.

Example 4-7 Checking for the CAA cluster

```
#lscluster -c
Cluster Name: cluster3738
Cluster UUID: b9b87978-611e-11e3-aa68-0011257e4371
Number of nodes in cluster = 2
        Cluster ID for node hacmp37.austin.ibm.com: 1
        Primary IP address for node hacmp37.austin.ibm.com: 9.3.44.37
        Cluster ID for node hacmp38.austin.ibm.com: 2
        Primary IP address for node hacmp38.austin.ibm.com: 9.3.44.38
Number of disks in cluster = 1
       Disk = hdisk2 UUID = 9c167b07-5678-4e7a-b468-e8b672bbd9f9 cluster_major
= 0 cluster_minor = 1
Multicast for site LOCAL: IPv4 228.3.44.38 IPv6 ff05::e403:2c26
Communication Mode: unicast
Local node maximum capabilities: HNAME_CHG, UNICAST, IPV6, SITE
Effective cluster-wide capabilities: HNAME_CHG, UNICAST, IPV6, SITE
```

15. Verify that UNICAST is in place for CAA inter-node communications on hacmp37, as shown in Example 4-8.

Example 4-8 Verifying that unicast is in place for CAA inter-node communication

```
# lscluster -m
Calling node query for all nodes...
Node query number of nodes examined: 2
```

```
        Node name: hacmp37.austin.ibm.com
        Cluster shorthand id for node: 1
        UUID for node: b90a2f9e-611e-11e3-aa68-0011257e4371
        State of node: UP  NODE_LOCAL
        Smoothed rtt to node: 0
        Mean Deviation in network rtt to node: 0
        Number of clusters node is a member in: 1
        CLUSTER NAME       SHID         UUID
        cluster3738        0            b9b87978-611e-11e3-aa68-0011257e4371
        SITE NAME          SHID         UUID
        LOCAL              1            51735173-5173-5173-5173-517351735173

        Points of contact for node: 0

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

        Node name: hacmp38.austin.ibm.com
        Cluster shorthand id for node: 2
        UUID for node: b90a3066-611e-11e3-aa68-0011257e4371
        State of node: UP
        Smoothed rtt to node: 7
        Mean Deviation in network rtt to node: 3
        Number of clusters node is a member in: 1
        CLUSTER NAME       SHID         UUID
        cluster3738        0            b9b87978-611e-11e3-aa68-0011257e4371
        SITE NAME          SHID         UUID
        LOCAL              1            51735173-5173-5173-5173-517351735173

        Points of contact for node: 1
-----------------------------------------------------------------------
        Interface     State  Protocol    Status     SRC_IP->DST_IP
-----------------------------------------------------------------------
        tcpsock->02   UP     IPv4         none   10.1.1.37->10.1.1.38
```

Note: The **lscluster -m** output on the remote node shows the reverse unicast network direction:

```
-----------------------------------------------------------------------
tcpsock->01   UP     IPv4         none   10.1.1.38->10.1.1.37
-----------------------------------------------------------------------
```

16. Install all PowerHA 7.1.3 filesets on node hacmp38 (use **smitty update_all**).
17. Start cluster services on node hacmp38 (**smitty clstart**).
18. Verify that the cluster has completed the migration on both nodes, as shown in Example 4-9.

Example 4-9 Verifying the migration has completed on both nodes

```
# odmget HACMPcluster | grep cluster_version
        cluster_version = 15

# odmget HACMPnode | grep version | sort -u
        version = 15
```

Note: These entries are shown in `/var/hacmp/log/clstrmgr.debug` (code snippet):

```
Updating ACD HACMPnode stanza with node_id = 2 and version = 15 for object
finishMigrationGrace: Migration is complete
```

19. Check for the updated `clstrmgrES` information, as shown in Example 4-10.

Example 4-10 Checking the updated clstrmgrES information

```
#lssrc -ls clstrmgrES
Current state: ST_STABLE
sccsid = "@(#)36 1.135.1.118
src/43haes/usr/sbin/cluster/hacmprd/main.C,hacmp.pe,61haes_r713,1343A_hacmp713
10/21/"
build = "Oct 31 2013 13:49:41 1344B_hacmp713"
i_local_nodeid 0, i_local_siteid -1, my_handle 1
ml_idx[1]=0     ml_idx[2]=1
There are 0 events on the Ibcast queue
There are 0 events on the RM Ibcast queue
CLversion: 15
local node vrmf is 7130
cluster fix level is "0"
```

Note: Both nodes must show CLversion: 15. Otherwise, the migration has not completed successfully. In that casel, call IBM Support.

4.4.3 Rolling migration from PowerHA SystemMirror 7.1.0 to PowerHA SystemMirror 7.1.3 (AIX 7.1 TL3 or 6.1 TL9)

The following describes the rolling migration from PowerHA 7.1.0 or later to PowerHA 7.1.3. The existing cluster configuration is the same as described on 4.4.2, "Rolling migration from PowerHA SystemMirror 6.1 to PowerHA SystemMirror 7.1.3 (AIX 7.1 TL3 or 6.1 TL9)" on page 54 except:

- No disk-heartbeat network
- PowerHA SystemMirror 7.1.0 cluster uses multicast

Note:
- The running AIX level for the following migration is AIX 7.1 TL3 SP0.
- The running PowerHA Level is PowerHA 7.1.0 SP8.
- Remember the requirements for PowerHA 7.1.3:

  AIX 6.1 TL9 SP0 or AIX 7.1 TL3 SP0

1. On node hacmp37 (the first node to be migrated), stop cluster services (**`smitty clstop`**) with the option to *Move Resource Groups* (this action moves over the resource groups to hacmp38).
2. Install all PowerHA 7.1.3 filesets (use **`smitty update_all`**).
3. Start cluster services on node hacmp37 (**`smitty clstart`**).
4. The output of the **`lssrc -ls clstrmgrES`** command on node hacmp37 is shown in Example 4-11 on page 62.

Example 4-11 lssrc -ls clstrmgrES output from node hacmp37

```
#lssrc -ls clstrmgrES
Current state: ST_STABLE
sccsid = "@(#)36 1.135.1.118
src/43haes/usr/sbin/cluster/hacmprd/main.C,hacmp.pe,61haes_r713,1343A_hacmp713
10/21/"
build = "Oct 31 2013 13:49:41 1344B_hacmp713"
i_local_nodeid 0, i_local_siteid -1, my_handle 1
ml_idx[1]=0     ml_idx[2]=1
There are 0 events on the Ibcast queue
There are 0 events on the RM Ibcast queue
CLversion: 12 <--- This means the migration still in progress !!!
local node vrmf is 7130
cluster fix level is "0"
```

5. On hacmp38, stop cluster services with the option *Move Resource Groups*.
6. Install all PowerHA 7.1.3 filesets (use **`smitty update_all`**).
7. Start cluster services on node hacmp38 (**`smitty clstart`**).
8. Verify that the cluster has completed the migration on both nodes as shown in Example 4-12.

Example 4-12 Verifying migration completion on both nodes

```
# odmget HACMPcluster | grep cluster_version
        cluster_version = 15

# odmget HACMPnode | grep version | sort -u
        version = 15
```

> **Note:** The following entries are shown in `/var/hacmp/log/clstrmgr.debug` (snippet):
>
> ```
> Updating ACD HACMPnode stanza with node_id = 2 and version = 15 for object
> finishMigrationGrace: Migration is complete
> ```

9. Check for the updated `clstrmgrES` information as shown in Example 4-13.

Example 4-13 Checking updated clstrmgrES information

```
#lssrc -ls clstrmgrES
Current state: ST_STABLE
sccsid = "@(#)36 1.135.1.118
src/43haes/usr/sbin/cluster/hacmprd/main.C,hacmp.pe,61haes_r713,1343A_hacmp713
10/21/"
build = "Oct 31 2013 13:49:41 1344B_hacmp713"
i_local_nodeid 1, i_local_siteid -1, my_handle 2
ml_idx[1]=0     ml_idx[2]=1
There are 0 events on the Ibcast queue
There are 0 events on the RM Ibcast queue
CLversion: 15 <--- This means the migration completed.
local node vrmf is 7130
```

> **Note:** Both nodes must show *CLversion: 15*. Otherwise, the migration has not completed successfully. In that case, call IBM Support.

4.4.4 Rolling migration from PowerHA SystemMirror 7.1.2 to PowerHA SystemMirror 7.1.3 (AIX 6.1 TL8 or 7.1 TL2)

The cluster configuration in this scenario consists of the following:

- AIX 7.1 TL3 SP0
- PowerHA 7.1 SP2
- Two node cluster and single resource group

> **Note:**
>
> - In the following migration example, the nodes are running AIX 7.1 TL2.
> - This migration example also applies to:
>   - Nodes running PowerHA 7.1.0 and AIX 6.1 TL8 or AIX 7.1 TL2.
>   - Nodes running PowerHA 7.1.1 and AIX 6.1 TL8 or AIX 7.1 TL2.

1. On node hacmp37 (the first node to be migrated), stop cluster services (**`smitty clstop`**) with the option to *Move Resource Groups* (moves the RGs over to node hacmp38).
2. Apply AIX TL3 - SP1 on node hacmp37, then reboot node hacmp37.
3. Install all PowerHA 7.1.3 filesets (use **`smitty update_all`**).
4. Start cluster services on node hacmp37 (**`smitty clstart`**).
5. The output of the **`lssrc -ls clstrmgrES`** command on node hacmp37 is shown in Example 4-14.

Example 4-14 lssrc -ls clstrmgrES command output of node hacmp37

```
#lssrc -ls clstrmgrES
Current state: ST_STABLE
sccsid = "@(#)36 1.135.1.118
src/43haes/usr/sbin/cluster/hacmprd/main.C,hacmp.pe,61haes_r713,1343A_hacmp713
10/21/"
build = "Oct 31 2013 13:49:41 1344B_hacmp713"
i_local_nodeid 0, i_local_siteid -1, my_handle 1
ml_idx[1]=0     ml_idx[2]=1
There are 0 events on the Ibcast queue
There are 0 events on the RM Ibcast queue
CLversion: 14 <--- This means the migration still in progress.
local node vrmf is 7130
cluster fix level is "0"
```

6. On hacmp38, stop cluster services with the option to **Move Resource Groups**.
7. Apply AIX TL3 - SP1 on node hacmp38, and then reboot node hacmp38.
8. Install all PowerHA 7.1.3 filesets (use **`smitty update_all`**).
9. Start cluster services on node hacmp38 (**`smitty clstart`**).
10. Verify that the cluster has completed migration on both nodes, as shown in Example 4-15 on page 64.

Example 4-15 Verifying completed migration

```
# odmget HACMPcluster | grep cluster_version
        cluster_version = 15

# odmget HACMPnode | grep version | sort -u
        version = 15
```

> **Note:** The following entries are shown in `/var/hacmp/log/clstrmgr.debug` (code snippet):
>
> `Updating ACD HACMPnode stanza with node_id = 2 and version = 15 for object`
> `finishMigrationGrace: Migration is complete`

11. Check for the updated `clstrmgrES` information as shown in Example 4-16.

Example 4-16 Checking for the updated clstrmgrES information

```
#lssrc -ls clstrmgrES
Current state: ST_STABLE
sccsid = "@(#)36 1.135.1.118
src/43haes/usr/sbin/cluster/hacmprd/main.C,hacmp.pe,61haes_r713,1343A_hacmp713
10/21/"
build = "Oct 31 2013 13:49:41 1344B_hacmp713"
i_local_nodeid 1, i_local_siteid -1, my_handle 2
ml_idx[1]=0     ml_idx[2]=1
There are 0 events on the Ibcast queue
There are 0 events on the RM Ibcast queue
CLversion: 15 <--- This means the migration completed !!!
local node vrmf is 7130
```

> **Note:** Both nodes must show *CLversion: 15*. Otherwise, the migration has not completed successfully. In that case, call IBM Support.

4.4.5 Snapshot migration to PowerHA SystemMirror 7.1.3

The following steps are required for a snapshot migration from PowerHA v6.1. Most of these steps can be performed in parallel because the entire cluster will be offline.

> **Tip:** A demo of performing a snapshot migration from PowerHA v6.1 to PowerHA v7.1.3 is available at:
>
> https://www.youtube.com/watch?v=1pkaQVB8r88

1. Stop cluster services on all nodes.

   Choose to bring resource groups offline.
2. Create a cluster snapshot if you have not previously created one and saved copies of it.
3. Upgrade AIX (if needed).
4. Install additional requisite filesets as listed in “Software requirements” on page 50.
5. Reboot.
6. Verify that **`clcomd`** is active:

`lssrc -s clcomd`

7. Update /etc/cluster/rhosts.

   Enter either cluster node hostnames or IP addresses, only one per line.

8. Run **Refresh -s clcomd**

9. Execute **clmigcheck** on one node.

   – Choose option 2 to verify that the cluster snapshot configuration is supported (assuming no errors).

   – Then, choose option 3.

     • Choose default multicast, user multicast, or unicast for heartbeat.
     • Choose a repository disk device to be used (for each site if applicable).
     • Exit the **clmigcheck** menu.

   – Review the contents of /var/clmigcheck/clmigcheck.txt for accuracy.

10. Uninstall the current version of PowerHA via **smitty remove**, and specify **cluster.***.

11. Install the new PowerHA version, including service packs, on both nodes.

12. Execute **clconvert_snapshot**. This command is in /usr/es/sbin/cluster/conversion.

    Syntax example:

    `clconvert_snapshot -v <version migrating from> -s <snapshot>`

13. Restore the cluster configuration from the converted snapshot.

14. Restart cluster services on each node, one at a time.

4.5 Automate the cluster migration check

As explained in previous IBM Redbooks publications, such as the *IBM PowerHA SystemMirror Standard Edition 7.1.1 for AIX Update*, SG24-8030 and the *IBM PowerHA SystemMirror 7.1.2 Enterprise Edition for AIX*, SG24-8106, the migration from PowerHA 6.1 to PowerHA 7.1 could only be done with exact requirements. The **clmigcheck** script assists you with these requirement checks and the cluster migration. A limitation of the **clmigcheck** script is that it could not be automated by using a response file.

The following sections discuss the limitations, dependencies, and steps to prepare for a cluster migration without running the **clmigcheck** script.

4.5.1 Limitations

Rolling migrations are *not* supported to run without using the **clmigcheck**. This is related to the change of the cluster service from RSCT (Reliable Scalable Cluster Technology) to CAA (Cluster Aware AIX) during a rolling migration. The migration must be done at a specific point in time to ensure successful migration without causing an outage.

4.5.2 Preparation and check

If you are not planning to use the **clmigcheck** script, it is your responsibility to ensure that the requirements are met. The following steps provide guidance for the checks and preparations:

1. Ensure that unsupported hardware is *not* used.
2. Ensure that IP address takeover (IPAT) via IP replacement is *not* used.

3. The communications path for the node must be set to `hostname` on all nodes.
4. When migrating to multicast:
   a. Choose a free multicast address or use cluster-defined multicast address (see Section 3.1.2, Network considerations, in *IBM PowerHA SystemMirror 7.1.2 Enterprise Edition for AIX*, SG24-8106.
   b. Test the communication with **mping**.
5. Choose the future repository disks and note the pvids of the disks.

4.5.3 Automated snapshot migration steps

The automated snapshot migration differs in only a few points from the standard snapshot migration described in 4.4.5, “Snapshot migration to PowerHA SystemMirror 7.1.3” on page 64. For easier reading, we outline the steps here, too, but please follow the steps in that section if you are using the **clmigcheck** script during the snapshot migration.

Most of these steps can be performed in parallel, because the entire cluster will be offline:

1. Stop cluster services on all nodes.

   Choose to bring the resource groups offline.
2. Create a cluster snapshot if you have not previously created one, and copy it to `/tmp` as backup, as shown in Example 4-17.

   Example 4-17 Creating a cluster snapshot

   ```
   root@a2: # > /usr/es/sbin/cluster/utilities/clsnapshot -c -i \
   -n'PowerHA713_mig_snap'  -d 'Snapshot for PowerHA71 migration'
   root@a2: # > cp /usr/es/sbin/cluster/snapshots/PowerHA713_mig_snap* /tmp/
   ```
3. Upgrade AIX (if needed).
4. Install the additional prerequisite filesets that are listed in 4.2.1, “Software requirements” on page 50, and then reboot.
5. Verify that **clcomd** is active:

   `lssrc -s clcomd`
6. Uninstall the current version of PowerHA.
7. Update `/etc/cluster/rhosts`.

   Enter either the cluster node hostnames or the IP addresses, only one per line.
8. Run **Refresh -s clcomd**.
9. Create `clmigcheck.txt` as described in 4.5.4, “Creating the clmigcheck.txt” on page 69.
10. Place the `climgcheck.txt` file in `/var/clmigcheck/clmigcheck.txt` on *every node* of the cluster.
11. Install the new PowerHA version, including service packs, on all nodes of the cluster.
12. Execute **clconvert_snapshot**.

    This command is in `/usr/es/sbin/cluster/conversion`. See Example 4-18.

    Syntax example:

    `clconvert_snapshot -v <version migrating from> -s <snapshot>`

Example 4-18 Executing the clconvert_snapshot

```
root@a2: # > /usr/es/sbin/cluster/conversion/clconvert_snapshot -v 6.1 -s \ PowerHA713_mig_snap
Extracting ODM's from snapshot file... done.
Converting extracted ODM's... done.
Rebuilding snapshot file... done.
```

Note: Depending on the number of managed service addresses and aliases, it could take several minutes to convert the snapshot. Please be patient when the snapshot is running. If you want to ensure that the process is still working, use the **proctree** command on the PID of the **clconvert_snapshot** several times and watch for changing output.

13. Restore the cluster configuration from the converted snapshot with the **clsnapshot** command, as shown in Example 4-19. The command also executes the **mkcluster** command that creates the CAA cluster. After the command finishes, the defined hdisk should display as caavg_private.

Example 4-19 Restoring the cluster configuration

```
root@a2: # > usr/es/sbin/cluster/utilities/clsnapshot -a -n'PowerHA71_mig_snap'
-f'false'

clsnapshot: Removing any existing temporary PowerHA SystemMirror ODM entries...

clsnapshot: Creating temporary PowerHA SystemMirror ODM object classes...
clsnapshot: Adding PowerHA SystemMirror ODM entries to a temporary directory..
clsnapshot: Verifying configuration using temporary PowerHA SystemMirror ODM
entries...
Verification to be performed on the following:
        Cluster Topology
        Cluster Resources

Retrieving data from available cluster nodes.  This could take a few minutes.

        Start data collection on node a2
        Start data collection on node b2
        Collector on node a2 completed
        Collector on node b2 completed
        Data collection complete
        Completed 10 percent of the verification checks


For nodes with a single Network Interface Card per logical
network configured, it is recommended to include the file
'/usr/es/sbin/cluster/netmon.cf' with a "pingable"
IP address as described in the 'HACMP Planning Guide'.
WARNING: File 'netmon.cf' is missing or empty on the following nodes:
a2
b2
   Completed 20 percent of the verification checks
WARNING: Network option "nonlocsrcroute" is set to 0 and will be set to 1 on
during PowerHA SystemMirror startup on the following nodes:

        a2
        b2
```

```
WARNING: Network option "ipsrcrouterecv" is set to 0 and will be set to 1 on
during PowerHA SystemMirror startup on the following nodes:

        a2
        b2

        Completed 30 percent of the verification checks
This cluster uses Unicast heartbeat
        Completed 40 percent of the verification checks

WWARNING: Application monitors are required for detecting application failures
in order for PowerHA SystemMirror to recover from them.  Application monitors
are started
by PowerHA SystemMirror when the resource group in which they participate is
activated.
The following application(s), shown with their associated resource group,
do not have an application monitor configured:

   Application Server                Resource Group
   --------------------------------  ---------------------------------
   app1_httpstart                   app_rg_1
   app2_httpstart                   app_rg_2
        Completed 50 percent of the verification checks
        Completed 60 percent of the verification checks
        Completed 70 percent of the verification checks
        Completed 80 percent of the verification checks
        Completed 90 percent of the verification checks
        Completed 100 percent of the verification checks

Verification has completed normally.

clsnapshot: Removing current PowerHA SystemMirror cluster information...
Deleting the cluster definition from "a2"...

clsnapshot: Adding new PowerHA SystemMirror ODM entries...

clsnapshot: Synchronizing cluster configuration to all cluster nodes...
/etc/es/objrepos
Timer object autoclverify already exists

Committing any changes, as required, to all available nodes...
lscluster: Cluster services are not active.
Adding any necessary PowerHA SystemMirror for AIX entries to /etc/inittab and
/etc/rc.net for IP Address Takeover on node a2.


cldare: Configuring a 2 node cluster in AIX may take up to 2 minutes. Please
wait.

Adding any necessary PowerHA SystemMirror for AIX entries to /etc/inittab and
/etc/rc.net for IP Address Takeover on node b2.

Verification has completed normally.
```

```
clsnapshot: Succeeded applying Cluster Snapshot: PowerHA71_mig_snap
```

14. Restart cluster services on each node, one at a time.

4.5.4 Creating the clmigcheck.txt

Depending on the cluster type that you have, the `clmigcheck.txt` file must be created. The following four `clmigcheck.txt` examples show different types of clusters.

Example 4-20 is used for a two-node cluster that is using unicast communication with PowerHA 7.1. Unicast communication is supported with PowerHA 7.1 TL3 and later.

Example 4-20 clmigcheck.txt for stretched cluster using unicast communication

```
CLUSTER_TYPE:STANDARD
CLUSTER_REPOSITORY_DISK:00f70c9976cc355b
CLUSTER_MULTICAST:UNI
```

Example 4-21 is used for a two-node cluster that is using multicast communication, where the operator has assigned a special multicast address.

Example 4-21 clmigcheck.txt for stretched cluster using user-defined multicast communication

```
CLUSTER_TYPE:STRETCHED
CLUSTER_REPOSITORY_DISK:00c4c9f2eafe5b06
CLUSTER_MULTICAST:224.10.10.65
```

Example 4-22 can be used for a stretched cluster with multicast communication. But this time, the cluster itself defines the multicast address during migration. This is done by a clearly defined process that is explained in 3.1.2 Network Considerations, in the IBM Redbooks publication titled *IBM PowerHA SystemMirror 7.1.2 Enterprise Edition for AIX*, SG24-8106.

Example 4-22 clmigcheck.txt for stretched cluster using cluster defined multicast communication

```
CLUSTER_TYPE:STANDARD
CLUSTER_REPOSITORY_DISK:00c0fb32fcff8cc2
CLUSTER_MULTICAST:NULL
```

> **Note:** This is a preferred way for migrating several clusters of the same application type:
>
> Use the **`clmigcheck`** script on one of the clusters to ensure compliance with the requirements, and generate files for the other cluster.

4.5.5 Offline migration to PowerHA SystemMirror 7.1.3

The following steps are required to perform an offline migration. These steps can often be performed in parallel, because the entire cluster will be offline.

> **Tip:** A demo of performing an offline migration from PowerHA v6.1 to PowerHA v7.1.2 is available on YouTube. The only difference compared to v7.1.3 is that the order of choosing repository disk and IP address is the opposite, and there is a new option to choose unicast.
>
> `http://youtu.be/7kl0JtcL2Gk`

1. Stop cluster services on all nodes.

Choose to bring resource groups offline.

2. Upgrade AIX (if needed).
3. Install the additional requisite filesets that are listed in 4.2.1, “Software requirements” on page 50.

   Reboot.
4. Verify that **clcomd** is active:

   `lssrc -s clcomd`
5. Update `/etc/cluster/rhosts`.

   Enter either cluster node hostnames or IP addresses, only one per line.
6. Run **Refresh -s clcomd**.
7. Execute **clmigcheck** on one node.
   a. Choose option 1 to verify that the cluster configuration is supported (assuming no errors).
   b. Then choose option 3.
      i. Choose default multicast, user multicast, or unicast for heartbeat.
      ii. Choose a repository disk device to be used (for each site, if applicable).
      iii. Exit the **clmigcheck** menu.
   c. Review the contents of `/var/clmigcheck/clmigcheck.txt` for accuracy.
8. Upgrade PowerHA on one node.
   a. Install base-level mages only (apply service packs later).
   b. Review the `/tmp/clconvert.log` file.
9. Execute **clmigcheck** and upgrade PowerHA on the remaining node.

   When executing **clmigcheck** on each additional node, the menu does not appear and no further actions are needed. On the last node, it creates the CAA cluster.
10. Restart cluster services.

4.5.6 Non-disruptive migration from SystemMirror 7.1.2 to 7.1.3

The existing cluster configuration is the same as described on 4.4.2, “Rolling migration from PowerHA SystemMirror 6.1 to PowerHA SystemMirror 7.1.3 (AIX 7.1 TL3 or 6.1 TL9)” on page 54 except in the following situations:

- No disk heartbeat network
- PowerHA SystemMirror 7.1.2 cluster uses multicast
- AIX 7.1 TL3 or PowerHA 7.1.2 SP

> **Note:** A demo of performing a non-disruptive *update* (not *upgrade*) on PowerHA v7.1.2 is available on YouTube. The process is identical; it’s just not a full upgrade.
>
> http://youtu.be/fZpYiu8zAZo

1. On node hacmp37 (the first node to be migrated), stop cluster services (**smitty clstop**) with the option to *Unmanage Resource Groups* as shown in Example 4-23.

Example 4-23 Stopping cluster services on hacmp37

```
# lssrc -ls clstrmgrES
Current state: ST_STABLE
sccsid = "@(#)36 1.135.9.1
src/43haes/usr/sbin/cluster/hacmprd/main.C,hacmp.pe,61haes_r712,1304C_hacmp712
2/21/13 "
build = "Jul 12 2013 14:07:16 1323C_hacmp712"
i_local_nodeid 0, i_local_siteid -1, my_handle 1
ml_idx[1]=0     ml_idx[2]=1
Forced down node list: hacmp37
There are 0 events on the Ibcast queue
There are 0 events on the RM Ibcast queue
CLversion: 14
local node vrmf is 7123
cluster fix level is "3"
```

2. Install all PowerHA 7.1.3 filesets (use **smitty update_all**).
3. Start cluster services on node hacmp37 (**smitty clstart**).
4. The output of the **lssrc -ls clstrmgrES** command on node hacmp37 is shown in Example 4-24.

Example 4-24 Output of lssrc -ls clstrmgrES on node hacmp37

```
# lssrc -ls clstrmgrES
Current state: ST_STABLE
sccsid = "@(#)36 1.135.1.118
src/43haes/usr/sbin/cluster/hacmprd/main.C,hacmp.pe,61haes_r713,1343A_hacmp713
10/21/"
build = "Nov  7 2013 09:13:10 1345A_hacmp713"
i_local_nodeid 0, i_local_siteid -1, my_handle 1
ml_idx[1]=0     ml_idx[2]=1
There are 0 events on the Ibcast queue
There are 0 events on the RM Ibcast queue
CLversion: 14
local node vrmf is 7130
cluster fix level is "0"
```

5. On node hacmp38, stop cluster services with the option to *Unmanage Resource Groups* as shown in Example 4-25.

Example 4-25 Stopping cluster services on hacmp38

```
# lssrc -ls clstrmgrES
Current state: ST_STABLE
sccsid = "@(#)36 1.135.9.1
src/43haes/usr/sbin/cluster/hacmprd/main.C,hacmp.pe,61haes_r712,1304C_hacmp712
2/21/13 "
build = "Jul 12 2013 14:07:16 1323C_hacmp712"
i_local_nodeid 1, i_local_siteid -1, my_handle 2
ml_idx[1]=0     ml_idx[2]=1
Forced down node list: hacmp38
There are 0 events on the Ibcast queue
There are 0 events on the RM Ibcast queue
CLversion: 14
local node vrmf is 7123
```

```
cluster fix level is "0"
```

6. Install all PowerHA 7.1.3 filesets (use **smitty update_all**).
7. Start cluster services on node hacmp38 (**smitty clstart**).
8. Verify that the cluster has completed migration on both nodes, as shown in Example 4-26.

Example 4-26 Verifying migration completion on both nodes

```
# odmget HACMPcluster | grep cluster_version
        cluster_version = 15

# odmget HACMPnode | grep version | sort -u
        version = 15
```

> **Note:** The following entries are shown in `/var/hacmp/log/clstrmgr.debug` (snippet):
>
> ```
> Updating ACD HACMPnode stanza with node_id = 2 and version = 15 for object
> finishMigrationGrace: Migration is complete
> ```

9. Check for the updated `clstrmgrES` information as shown in Example 4-27.

Example 4-27 Checking the updated clstrmgrES information

```
#lssrc -ls clstrmgrES
Current state: ST_STABLE
sccsid = "@(#)36 1.135.1.118
src/43haes/usr/sbin/cluster/hacmprd/main.C,hacmp.pe,61haes_r713,1343A_hacmp713
10/21/"
build = "Oct 31 2013 13:49:41 1344B_hacmp713"
i_local_nodeid 1, i_local_siteid -1, my_handle 2
ml_idx[1]=0     ml_idx[2]=1
There are 0 events on the Ibcast queue
There are 0 events on the RM Ibcast queue
CLversion: 15 <--- This means the migration completed !!!
local node vrmf is 7130
```

> **Note:** Both nodes must show *CLversion: 15*. Otherwise, the migration has not completed successfully. In that case, call IBM Support.

4.5.7 PowerHA SystemMirror 7.1.3 conversion from multicast to unicast

Use the following steps to convert from multicast to unicast communication mode:

1. Verify that the existing CAA communication mode is set to multicast, as shown in Example 4-28.

Example 4-28 Verifying CAA communication mode

```
# lscluster -c
Cluster Name: cluster3738
Cluster UUID: 5eeb1ae6-82c0-11e3-8eb9-0011257e4348
Number of nodes in cluster = 2
        Cluster ID for node hacmp37.austin.ibm.com: 1
        Primary IP address for node hacmp37.austin.ibm.com: 9.3.44.37
        Cluster ID for node hacmp38.austin.ibm.com: 2
```

```
        Primary IP address for node hacmp38.austin.ibm.com: 9.3.44.38
Number of disks in cluster = 1
        Disk = hdisk2 UUID = 9c167b07-5678-4e7a-b468-e8b672bbd9f9 cluster_major
= 0 cluster_minor = 1
Multicast for site LOCAL: IPv4 228.3.44.37 IPv6 ff05::e403:2c25
Communication Mode: multicast
Local node maximum capabilities: HNAME_CHG, UNICAST, IPV6, SITE
Effective cluster-wide capabilities: HNAME_CHG, UNICAST, IPV6, SITE
```

2. Change the heartbeat mechanism from multicast to unicast, as shown in Example 4-29.

Example 4-29 Changing the heartbeat mechanism

```
#smitty cm_define_repos_ip_addr

                   Define Repository Disk and Cluster IP Address

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

                                                      [Entry Fields]
* Cluster Name                                        cluster3738
* Heartbeat Mechanism                                 Unicast
  Repository Disk                                     000262ca102db1a2
  Cluster Multicast Address                           228.3.44.37
      (Used only for multicast heartbeat)

  Note: Once a cluster has been defined to AIX, all
  that can be modified is the Heartbeat Mechanism
```

3. Verify and synchronize the cluster (**smitty sysmirror > Cluster Nodes and Networks > Verify and Synchronize Cluster Configuration**).

   Verify that the new CAA communication mode is now set to unicast, as shown in Example 4-30 on page 73.

Example 4-30 Verifying the new CAA communication mode

```
# lscluster -c
Cluster Name: cluster3738
Cluster UUID: 5eeb1ae6-82c0-11e3-8eb9-0011257e4348
Number of nodes in cluster = 2
        Cluster ID for node hacmp37.austin.ibm.com: 1
        Primary IP address for node hacmp37.austin.ibm.com: 9.3.44.37
        Cluster ID for node hacmp38.austin.ibm.com: 2
        Primary IP address for node hacmp38.austin.ibm.com: 9.3.44.38
Number of disks in cluster = 1
        Disk = hdisk2 UUID = 9c167b07-5678-4e7a-b468-e8b672bbd9f9 cluster_major
= 0 cluster_minor = 1
Multicast for site LOCAL: IPv4 228.3.44.37 IPv6 ff05::e403:2c25
Communication Mode: unicast
Local node maximum capabilities: HNAME_CHG, UNICAST, IPV6, SITE
Effective cluster-wide capabilities: HNAME_CHG, UNICAST, IPV6, SITE
```

5

IBM PowerHA cluster simulator

For the PowerHA 7.1.3 release, the IBM Systems Director PowerHA team spent a significant amount of time developing the cluster simulator. We believe that it can be a very useful tool for IBM clients to explore and investigate PowerHA capabilities in a safe sandbox. No real nodes are needed to form a cluster, and the clients do not even need base PowerHA product. This simulator can run on any platform that is supported by supported by an IBM Systems Director server.

This chapter covers the following topics:

- Systems Director overview
- IBM Systems Director PowerHA cluster simulator
- Using SUSE Linux as a KVM guest system
- Importing configurations from stand-alone systems

5.1 Systems Director overview

IBM Systems Director is the IBM software that is used to managed heterogeneous environments. It supports IBM servers families (IBM BladeCenter®, IBM Power Systems, IBM PureFlex System, IBM System x®, and IBM System z®), network devices (including virtual network switches from PowerVM, KVM, Hyper-V, and VMWare), storage devices, and many applications, such as PowerHA.

5.1.1 IBM Systems Director components

IBM Systems Director operates as a framework for environment management. It works based in a structured set of components that interact and perform functions together as illustrated in Figure 5-1.

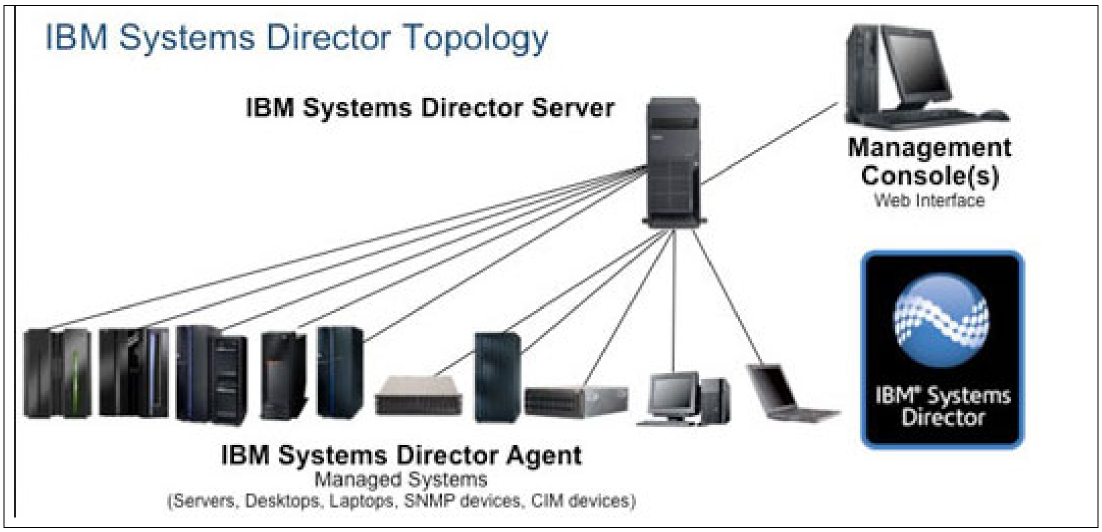


Figure 5-1 Basic IBM Systems Director topology

The following subsection that follow describe the main IBM Systems Director components.

Management server

The management server is the main entity of the topology, and it has the Systems Director server packages installed. The management server works as the central point for controlling the environment inventory, performing the operations on resources, and managing the IBM DB2 database, where all information is stored.

Common agent

The Common Agent is the basic agent for all managed servers. The agent allows the Systems Director server to view and manipulate server information and configuration, including security management and deployment functions. This component is installed on all servers that the environment administrator wants to have managed by IBM Systems Director. Each PowerHA node is considered to be a Common Agent because they run the Common Agent services (CAS). All PowerHA nodes must be discovered by the IBM Systems Director running on the Management Server.

Platform Agent

The Platform Agent acts similarly to the Common Agent but with fewer management options. The Platform Agent is designed for smaller environments, where just a subset of administrative operations are intended to be performed. We do not use Platform Agent with PowerHA.

Additional plug-ins

For specific use, there are additional plug-ins that can be downloaded from the IBM Systems Director download web page:

http://www.ibm.com/systems/director/downloads/plugins.html

> **Note:** For more information about the IBM Systems Director installation, management, and operations, see the IBM Knowledge Center:
>
> http://www.ibm.com/support/knowledgecenter/SSAV7B/welcome

5.2 IBM Systems Director PowerHA cluster simulator

In December 2013, with Systems Director 6.3.3 release, a new function for PowerHA for Systems Director plug-in was added to work in a simulation mode called the *cluster simulator*. With the cluster simulator, an administrator can simulate work (display, create, modify, delete) on PowerHA clusters with no connection to real PowerHA nodes and with no impact on real environments.

The information displayed on the console comes from local XML configuration files, and any changes performed in the PowerHA console are saved into local XML configuration files. This XML format is new with the 7.1.3 release. It is used as an interchange format between the PowerHA console and the IBM System Director database on one side and the PowerHA base product on the IBM AIX side (on the PowerHA node side). As explained later in this section, in the Planning mode of the cluster simulator, the XML configuration files that are built on the PowerHA console can be used on PowerHA nodes to deploy the corresponding configuration.

The IBM Systems Director PowerHA console works in two different modes:

- **Online mode:** In Online mode, the console works the way that it has always worked. The PowerHA console can be used to create a new PowerHA cluster or to manage an already configured and running PowerHA cluster. Management tasks are performed on a real, running PowerHA environment (real PowerHA nodes, real PowerHA cluster, and so on), the same way as before the 7.1.3 release. But new with the 7.1.3 release, this real configuration can now be exported to an XML configuration file.
- **Simulated mode:** In Simulated mode, the console works as a cluster simulator. It works in a disconnected fashion (disconnected from real PowerHA nodes) with no impact and no risk for a real PowerHA environment. Two modes are then possible:
  - **Offline mode:** This mode is entirely artificial. All information related to hostnames, IP addresses, volume groups, file systems, and services is fake and comes from a hardcoded XML environment file. In this mode, you interact only with an XML configuration file. You do not interact with a real PowerHA cluster, and you do not even need to have connection to any PowerHA nodes. In this mode, you use the Systems Director PowerHA console to create, display, change, or delete your PowerHA configuration and save it to an XML configuration file, with no possible risk to production environments. Several offline XML files are delivered, ready to use as starting points.

In this mode, the XML configuration file, which stores results of all actions from using the console, cannot be used in a real PowerHA environment. For example, it cannot really be deployed. This mode is useful only to learn and become familiar with the tool, to train others, and to demonstrate the PowerHA console and the PowerHA concepts.

– **Planning mode:** Planning mode is different from Offline mode because all information related to hostnames, IP addresses, volume groups, file systems, and services is collected from real PowerHA running nodes in an initial step. The XML environment file, which contains entirely fake and hardcoded data in Offline mode, contains real data in Planning mode. To collect this real environment from the PowerHA node, the PowerHA console needs to be connected to the PowerHA nodes. During this initial step when the XML environment file is created as a result of the collection, the PowerHA console can work in a disconnected fashion. Then, the configuration that is displayed in the console reflects a real environment.

  In this mode, as in Offline mode, you use IBM Systems Director PowerHA console to create, display, change, or delete your PowerHA configuration and save it to an XML configuration file, with no possible risk to the production environments. In this mode, the XML configuration file, which contains results of all actions while using the console, can be used in a real PowerHA environment.

  This mode is useful to prepare and plan a PowerHA configuration in a disconnected fashion. When your configuration is ready and verified, the resulting XML configuration files prepared with the console in the Planning mode can be deployed on real PowerHA nodes. In this Planning mode, you can create a new cluster configuration using real PowerHA nodes, real PVID disks, and so one, so that, at the end, the planned cluster, which is saved in an XML file, can actually be deployed.

5.2.1 Installing the PowerHA SystemMirror for Systems Director plug-in

To run the cluster simulator within Systems Director, the following requirements must be met on both the managed server and the managed agent to ensure that all functions run:

> **Note:** To run the simulator in Offline mode, only the PowerHA SystemMirror Director Server plug-in needs to be installed. Agent nodes are not needed.

- **Operating system:** To run the PowerHA plug-in for Systems Director, the minimum operating system version is AIX 6.1 TL9 or later or AIX 7.1 TL3 or later. For a managed server, any operating system supported by Systems Director 6.3 can run the plug-in.

  > **Note:** To check all supported environments to run Systems Director 6.3, see the *Operating Systems and Software Requirements* section of the IBM Knowledge Center:
  >
  > http://ibm.co/1uBLLRB

- **Systems Director server:** To support the cluster simulator feature, the minimum Systems Director server version is 6.3.2 or later.
- **PowerHA SystemMirror:** The minimum PowerHA version supported for the cluster simulator feature is PowerHA SystemMirror 7.1.3.

Installing PowerHA SystemMirror plug-in on a managed server

First, the plug-in must be installed on the Systems Director managed server to allow operations on PowerHA. To start it, download the appropriate agent version from the plug-ins download page:

http://www.ibm.com/systems/director/downloads/plugins.html

The installation is simple. After downloading and uncompressing the plug-in installation package, for AIX, Linux, or Microsoft Windows running a Systems Director server, just run the `IBMSystemsDirector_PowerHA_sysmirror_Setup.bin` binary file that is included in the package. The installation goes as Example 5-1 shows (this example is running on an AIX 7.1 operating system).

Example 5-1 Installing the PowerHA plug-in on a managed server

```
root@pokbclpar0102(/public/PowerHA_SD/AIX)#
./IBMSystemsDirector_PowerHA_sysmirror_Setup.bin
Preparing to install...
Extracting the installation resources from the installer archive...
Configuring the installer for this system's environment...

Launching installer...

Graphical installers are not supported by the VM. The console mode will be used
instead...

===============================================================================
Choose Locale...
----------------

    1- Deutsch
  ->2- English
    3- Espanol
    4- Francais
    5- Italiano
    6- Portuguese (Brasil)

CHOOSE LOCALE BY NUMBER: 2
===============================================================================
IBM PowerHA SystemMirror                        (created with InstallAnywhere)
-------------------------------------------------------------------------------

Preparing CONSOLE Mode Installation...

===============================================================================
Introduction
------------

InstallAnywhere will guide you through the installation of IBM PowerHA
SystemMirror.

It is strongly recommended that you quit all programs before continuing with
this installation.

Respond to each prompt to proceed to the next step in the installation.  If you
want to change something on a previous step, type 'back'.
```

```
You may cancel this installation at any time by typing 'quit'.

PRESS <ENTER> TO CONTINUE:
===============================================================================

    International Program License Agreement

    Part 1 - General Terms

    BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, CLICKING ON AN
    "ACCEPT" BUTTON, OR OTHERWISE USING THE PROGRAM, LICENSEE AGREES TO
    THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON
    BEHALF OF LICENSEE, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL
    AUTHORITY TO BIND LICENSEE TO THESE TERMS. IF YOU DO NOT AGREE TO
    THESE TERMS,

    - DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, CLICK ON AN "ACCEPT" BUTTON,
    OR USE THE PROGRAM; AND

    - PROMPTLY RETURN THE UNUSED MEDIA, DOCUMENTATION, AND PROOF OF
    ENTITLEMENT TO THE PARTY FROM WHOM IT WAS OBTAINED FOR A REFUND OF THE
    AMOUNT PAID. IF THE PROGRAM WAS DOWNLOADED, DESTROY ALL COPIES OF THE
    PROGRAM.


Press Enter to continue viewing the license agreement, or enter "1" to
   accept the agreement, "2" to decline it, "3" to print it, "4" to read
   non-IBM terms, or "99" to go back to the previous screen.: 1

===============================================================================
IBM Director Start
------------------

IBM Systems Director is currently running. Do you want IBM Systems Director to be
restarted automatically after IBM PowerHA SystemMirror is installed?  Although it
does not need to be stopped in order to install IBM PowerHA SystemMirror, it will
need to be restarted before IBM PowerHA SystemMirror functions are available.

    1- Yes
  ->2- No

ENTER THE NUMBER FOR YOUR CHOICE, OR PRESS <ENTER> TO ACCEPT THE DEFAULT:: 1


===============================================================================
Installing...
-------------

 [==================|==================|==================|==================]
 [------------------|------------------|------------------|------------------]
Thu Dec 19 10:07:50 CST 2013 PARMS: stop
Thu Dec 19 10:07:50 CST 2013 The lwi dir is: :/opt/ibm/director/lwi:
Thu Dec 19 10:07:50 CST 2013 localcp:
/opt/ibm/director/lwi/runtime/USMiData/eclipse/plugins/com.ibm.usmi.kernel.persist
```

```
ence_6.3.3.jar:/opt/ibm/director/lwi/runtime/USMiMain/eclipse/plugins/com.ibm.dire
ctor.core.kernel.nl1_6.3.3.1.jar:/opt/ibm/director/lwi/runtime/USMiData/eclipse/pl
ugins/com.ibm.usmi.kernel.persistence.nl1_6.3.2.jar:/opt/ibm/director/bin/..//bin/
pdata/pextensions.jar
Thu Dec 19 10:07:50 CST 2013 directorhome: /opt/ibm/director
Thu Dec 19 10:07:50 CST 2013 java_home: /opt/ibm/director/jre
Thu Dec 19 10:07:51 CST 2013 inscreenmessage STARTOFMESS --formatmessage-
-shuttingdown- -IBM Director- --
Thu Dec 19 10:07:51 CST 2013 starting value is shutting down Thu Dec 19 10:07:51
CST 2013 shutting down IBM Director
Thu Dec 19 10:07:51 CST 2013 Calling lwistop
Thu Dec 19 10:08:22 CST 2013 lwistop complete
Thu Dec 19 10:08:22 CST 2013 starting wait for shutdown on lwipid:
Thu Dec 19 10:08:22 CST 2013 Running PID: ::
Thu Dec 19 10:16:27 CST 2013 PARMS: start
Thu Dec 19 10:16:27 CST 2013 The lwi dir is: :/opt/ibm/director/lwi:
Thu Dec 19 10:16:27 CST 2013 localcp:
/opt/ibm/director/lwi/runtime/USMiData/eclipse/plugins/com.ibm.usmi.kernel.persist
ence_6.3.3.jar:/opt/ibm/director/lwi/runtime/USMiMain/eclipse/plugins/com.ibm.dire
ctor.core.kernel.nl1_6.3.3.1.jar:/opt/ibm/director/lwi/runtime/USMiData/eclipse/pl
ugins/com.ibm.usmi.kernel.persistence.nl1_6.3.2.jar:/opt/ibm/director/bin/..//bin/
pdata/pextensions.jar
Thu Dec 19 10:16:27 CST 2013 directorhome: /opt/ibm/director
Thu Dec 19 10:16:27 CST 2013 java_home: /opt/ibm/director/jre
Thu Dec 19 10:16:32 CST 2013 in screen message STARTOFMESS --formatmessage-
-starting- -IBM Director- --
Thu Dec 19 10:16:32 CST 2013 starting value is starting
Thu Dec 19 10:16:32 CST 2013 starting IBM Director
Thu Dec 19 10:16:32 CST 2013 in screen message STARTOFMESS --formatmessage-
-startingprocess- - - --
Thu Dec 19 10:16:33 CST 2013 starting value is starting process
Thu Dec 19 10:16:33 CST 2013 starting process
```

Installing the PowerHA SystemMirror plug-in on the cluster nodes

Now that the PowerHA plug-in is properly installed on the managed server, it is time to install the packages on the PowerHA cluster nodes.

Considering that the cluster nodes are servers controlled by the managed server, only two packages must be installed on them:

- Systems Director Common Agent 6.3.3 or later
- PowerHA SystemMirror for Systems Director 7.1.3 or later

> **Note:** Remember that only the PowerHA plug-in from version 7.1.3 or later allows the cluster simulator feature.

Both packages can be downloaded from the IBM Systems Director download page at:

http://www.ibm.com/systems/director/downloads/plugins.html

Install the `cluster.es.director.agent` fileset and Common Agent on each node in the cluster as you want.

Installing the Common Agent

Perform the following steps on each node that is going to be managed by the Systems Director server:

1. Extract the `SysDir6_3_3_Common_Agent_AIX.jar` file:

   ```
   #/usr/java5/bin/jar -xvf SysDir6_3_3_Common_Agent_AIX.jar
   ```

2. Assign execution permissions to the `repository/dir6.3_common_agent_aix.sh` file:

   ```
   # chmod +x repository/dir6.3_common_agent_aix.sh
   ```

3. Execute the `repository/dir6.3_common_agent_aix.sh` file:

   `# ./repository/dir6.3_common_agent_aix.sh` file

Some subsystems are added as part of the installation: *platform_agent* and *cimsys*.

Installing the PowerHA SystemMirror agent

To install the PowerHA SystemMirror agent, perform the following steps on each node:

1. Install the `cluster.es.director.agent.rte` fileset:

   ```
   #smitty install_latest
   ```

   Choose the directory and then the fileset name and execute to install.

2. Stop the Common Agent:

   ```
   # stopsrc -s platform_agent
   # stopsrc -s cimsys
   ```

3. Start the Common Agent:

   ```
   #startsrc -s platform_agent
   ```

> **Important:** The `cimsys` subsystem starts automatically with the `platform_agent` subsystem.

5.2.2 Choosing the mode on which the PowerHA console runs

This section describes how to choose the mode on which the PowerHA console runs.

Online mode

From the server side (ISD side), you can export your real configuration to an XML file by using either the console (see “Option A: Export to XML by using the console” on page 83) or the command line (see “Option B: Export to XML by using the command line” on page 88).

However, the next step is mandatory when exporting the XML configuration, whether using the console or the command line.

Option A: Export to XML by using the console

Visualize your cluster as shown in Figure 5-2.

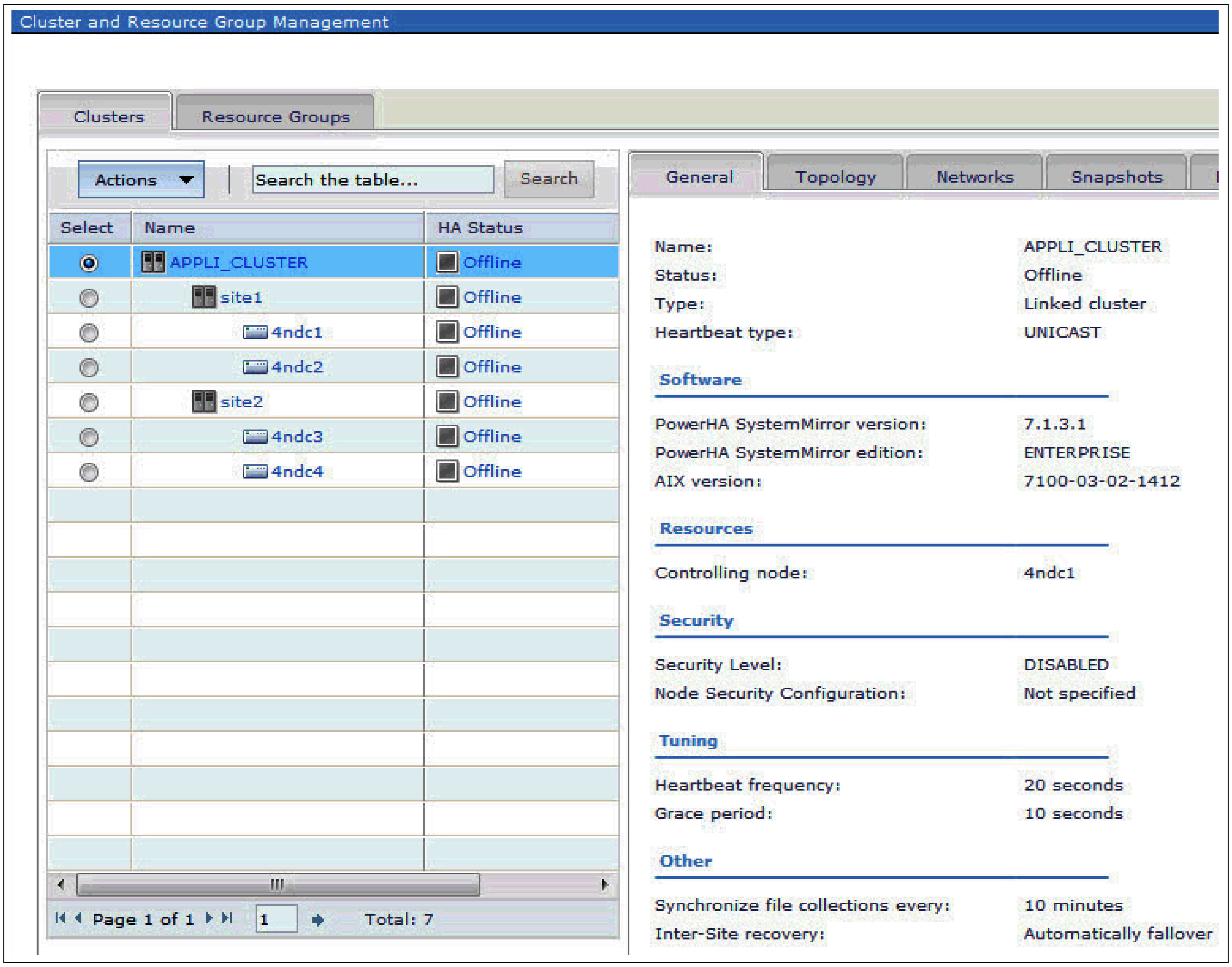


Figure 5-2 Visualizing your cluster

This is a real cluster, not a simulated one. To get to the Export cluster definition menu, click on the **Actions** menu, as shown in Figure 5-3.

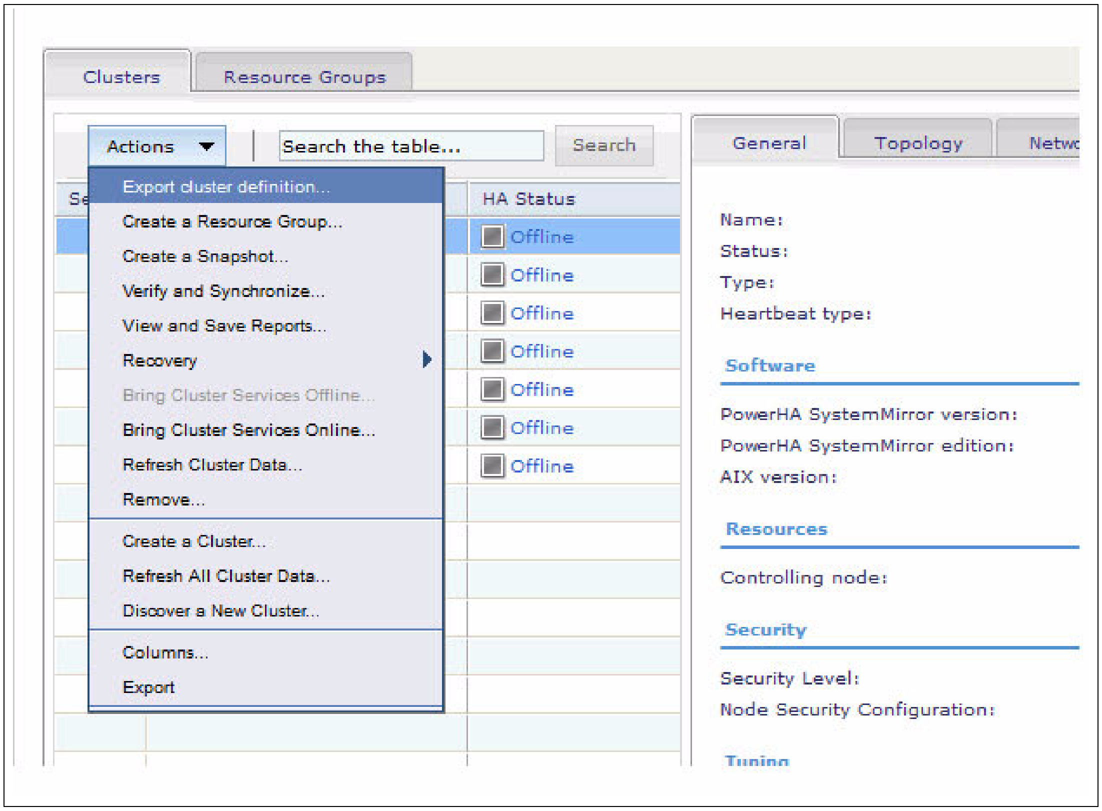


Figure 5-3 Export cluster definition

The “Export cluster definition” options panel opens, as shown in Figure 5-4.

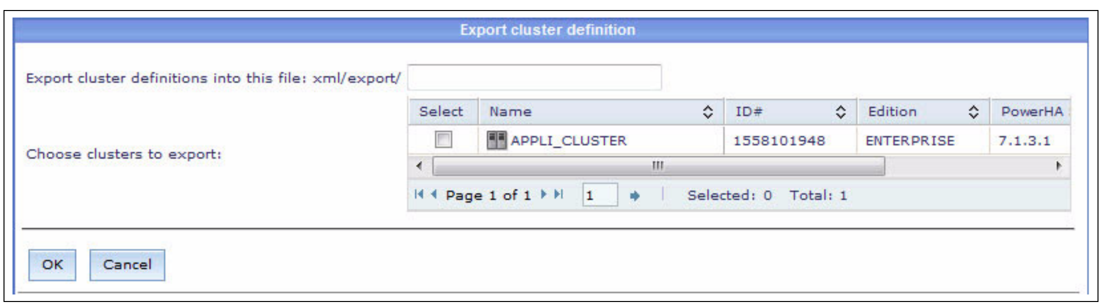


Figure 5-4 Export cluster definition menu

Fill in the “Export cluster definitions into this file: xml/export/” field, and select the cluster to export, as shown in Figure 5-5.

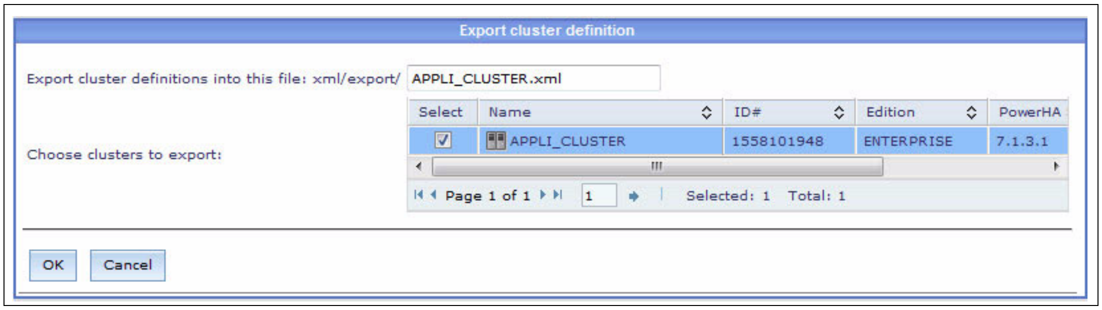


Figure 5-5 Adding the cluster name and selecting the cluster

Example 5-2 shows the results.

Example 5-2 Showing the cluster results

```
# smcli modemngt -l
PowerHA SystemMirror Cluster Simulator XML files :
 xml files dir
/opt/ibm/director/PowerHASystemMirror/eclipse/plugins/com.ibm.director.power.ha.sy
stemmirror.common_7.1.3.1/bin

 XML offline simulation files
  xml/simu/LC_713_data.xml
  xml/simu/NSC_7122_data.xml
  xml/simu/NSC_713_data.xml
  xml/simu/NSC_SC_LC_env.xml
  xml/simu/SC_713_data.xml

 XML custom offline simulation files
  xml/custom/shawnsdemo.xml
  xml/custom/shawnsdemo_env.xml

 XML planning files
  xml/planning/data.xml
  xml/planning/discdata_20140418_044421.xml
  xml/planning/discenv_20140418_044421.xml
  xml/planning/env.xml

 XML export files
  xml/export/APPLI_CLUSTER.xml
```

You can choose to work in Planning mode, but the Planning mode works only with the file in `xml/planning`, not with the files in `xml/export`. Therefore, you must complete the following manual steps:

1. Change to the directory:

```
cd
/opt/ibm/director/PowerHASystemMirror/eclipse/plugins/com.ibm.director.power.ha
.systemmirror.common_7.1.3.1/bin
```

2. Then copy the XML file:

```
cp xml/export/APPLI_CLUSTER.xml xml/planning
```

3. From the PowerHA SystemMirror menu, select a mode for this session, and then switch to **Planning** mode, as shown in Figure 5-6.

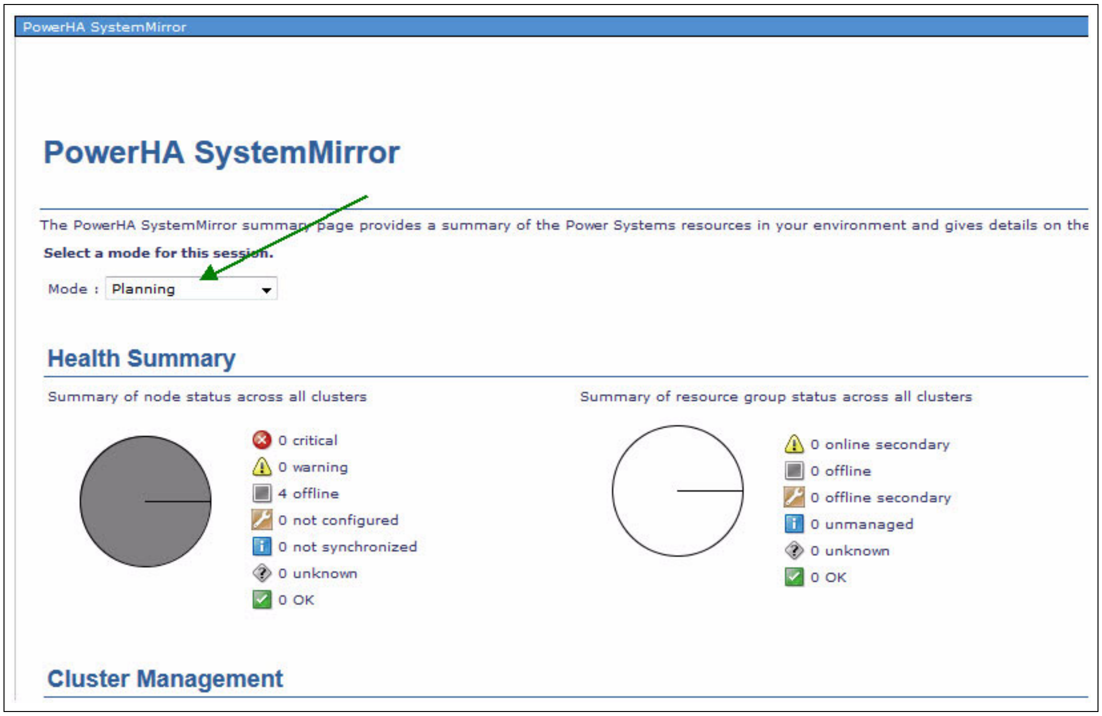


Figure 5-6 Switching to Planning mode

Then, the “Planning mode” pane shown in Figure 5-7 opens.

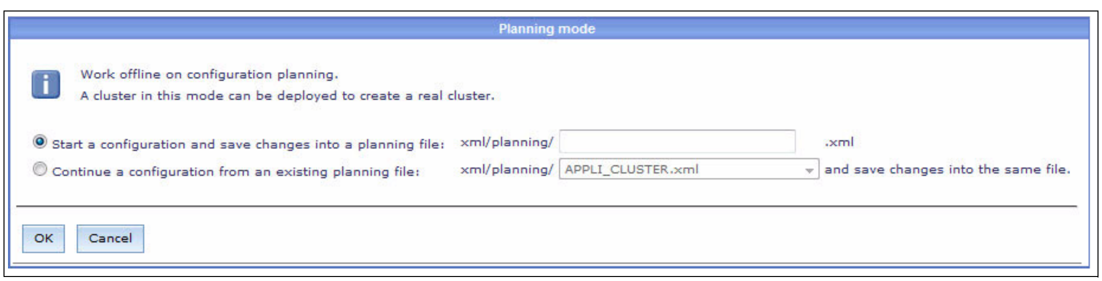


Figure 5-7 “Planning mode” configuration options panel

4. Select the radio button to **Continue a configuration from an existing planning file**, as shown in Figure 5-8 on page 87.

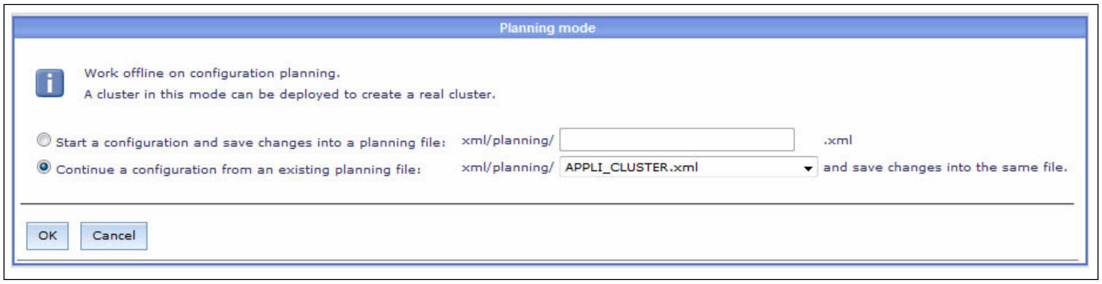


Figure 5-8 Planning mode option panel

You are then in Planning mode with the file that you specified as input, as shown in Figure 5-9.

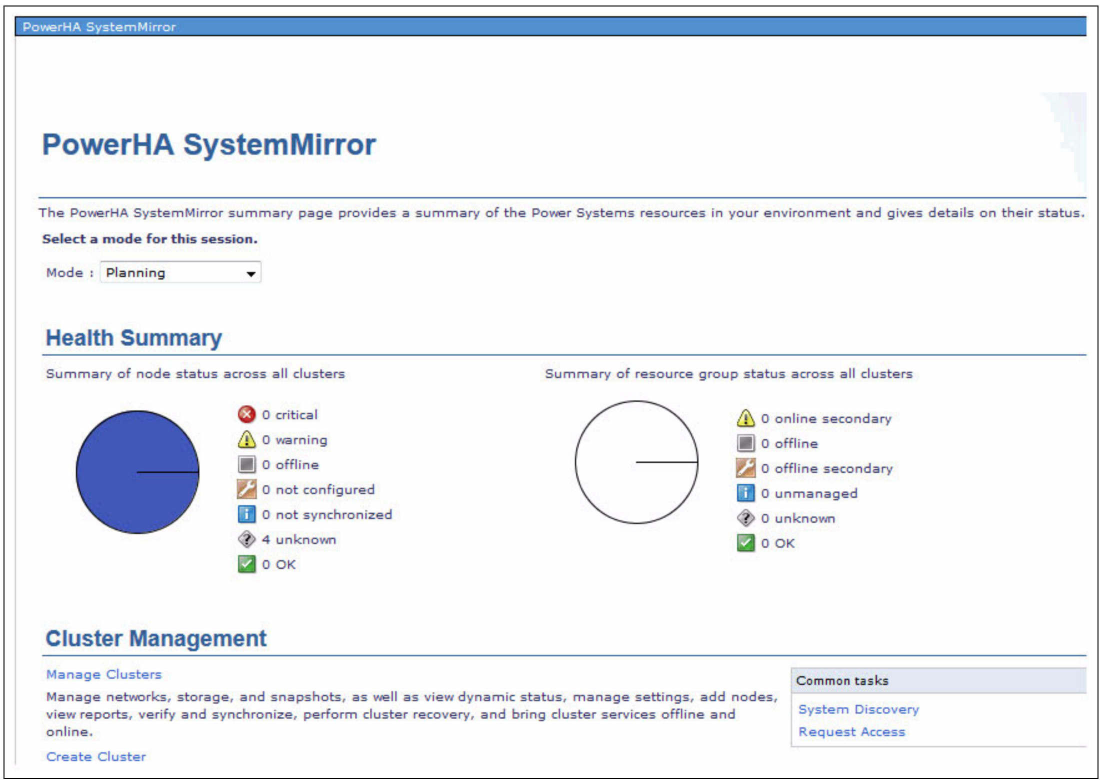


Figure 5-9 Planning mode using the selected XML input file

Figure 5-10 shows another view of Planning mode with the same XML file.

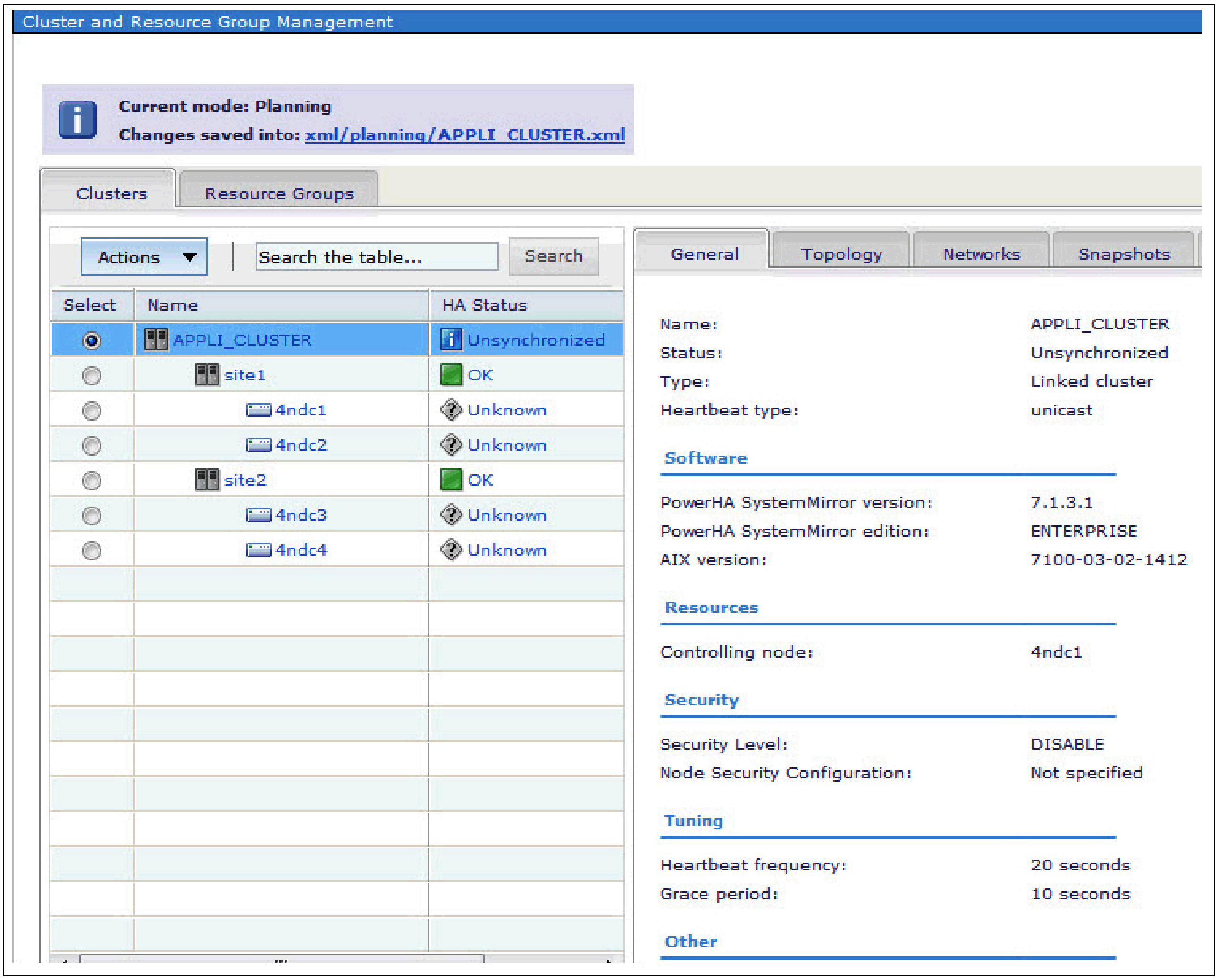


Figure 5-10 Cluster and resource group management menu

Option B: Export to XML by using the command line

Example 5-3 shows the command-line syntax to export to XML.

Example 5-3 Command line showing how to export to XML

```
# smcli exportcluster -h -v

smcli sysmirror/exportcluster [-h|-?|--help] [-v|--verbose]
smcli sysmirror/exportcluster [{-a|--data} <xml_data_file>] [<CLUSTER>]

Command Alias: excl
```

Before running the command shown in Example 5-3, check to make sure that you have the most recent version of the XML env file. If necessary, run the command **`smcli discoverenv`** with the following flags:

`-h|-?|--help`	Requests help for this command.
`-v|--verbose`	Requests maximum details in the displayed information.

-a|--data <xml_data_file> Sets the XML data file that contains the cluster to be exported. The file name is relative to the PowerHA SystemMirror root directory for XML files. For example:

`xml/planning/export_clusterxxx_data.xml`

If it is not set, a name is automatically generated.

<CLUSTER> The label of cluster to perform this operation on. If not specified and if there is only one cluster in the XML data file, this one is taken by default:

```
smcli sysmirror/exportcluster [-h|-?|--help]
[-v|--verbose]

smcli sysmirror/exportcluster [{-a|--data}
<xml_data_file>] [<CLUSTER>]

Command Alias: excl
```

Example 5-4 shows the **smcli exportcluster** command and the output messages.

Example 5-4 smcli exportcluster command

```
# smcli exportcluster
Using         :
 Xml files dir :
/opt/ibm/director/PowerHASystemMirror/eclipse/plugins/com.ibm.director.power.ha.sy
stemmirror.common_7.1.3.1/bin
 Xml data file : xml/planning/expclu_data_20140418_054125.xml
 Xml env file  : xml/planning/discenv_20140418_044421.xml
 Cluster       : APPLI_CLUSTER (1558101948)


Trying to export "APPLI_CLUSTER (1558101948)" cluster into
"xml/planning/expclu_data_20140418_054125.xml" file.

smcli exportcluster is successfull into
"xml/planning/expclu_data_20140418_054125.xml" file.
```

Notes:
You can work with the console in Planning mode using this file, but this time the exported file is already in `xml/planning` (`xml/planning/expclu_data_20140418_054125.xml`). Therefore, there is no need for manual copy from *xml/export* to `xml/planning`.

PowerHA console mode management can be done by using the command line. To get help, use the **smcli modemngt –h –v** command.

Create the discovered environment XML file

The XML env file is needed when you export your real configuration to an XML file (bold text, as shown in Example 5-5).

Example 5-5 Mandatory steps for creating the discovered environment XML file

```
# smcli discoverenv
 xml files dir
/opt/ibm/director/PowerHASystemMirror/eclipse/plugins/com.ibm.director.power.ha.sy
stemmirror.common_7.1.3.1/bin
Starting discovery ...
```

```
Discovering nodes ...
Discovering physical volumes ...
Discovering volume groups ...
Discovering /etc/hosts ips ...
Discovering users ...
Discovering groups ...
Discovering roles ...
Discovering interfaces ...
Discovering networks ...
Discovering cluster configurations ...
Processing cluster configurations ...

Environment discovery status:
  Discovered nodes : 4
  Discovered physical volumes : 48
  Discovered volumes group: 0
  Discovered interfaces : 0
Generated xml environment file : xml/planning/discenv_20140418_044421.xml

Data discovery status:
  Discovered users : 0
  Discovered groups : 0
  Discovered roles : 0
  Discovered cluster configurations : 0
Generated xml data file : xml/planning/discdata_20140418_044421.xml

(0) root @ scratchy09: : /home/desgrang
```

Option C: Export to XML agent side and deploy from XML agent side

This section shows how to use the command line (Example 5-6) on the agent side to export a real configuration to XML files and then use the generated XML files to deploy the configuration. The example is a deployment of the XML files that are generated from the agent side, but you can deploy agent-side XML files, which would have been exported from the server side in Planning mode.

Example 5-6 Export and deploy XML from the agent side

```
cd /tmp
mkdir xml
mkdir xml/planning
export PATH=/usr/java6/jre/bin:/usr/java6/bin:$PATH

ClmgrExport
ClmgrExport : Help Verbose
java -DCAS_AGENT=/var/opt/tivoli/ep -cp
/usr/es/sbin/cluster/utilities/clmgrutility.jar
com.ibm.director.power.ha.systemmirror.agent.impl.ClmgrExport --help --verbose

Currently running "ClmgrExport -h -v"

Usage :

ClmgrExport -h|--help [-v|--verbose]
```

```
ClmgrExport -x|--export [-i|--isd]  [-D|--Debug {0|1|2|3}]
[-L|--Level {SEVERE|WARNING|INFO|CONFIG|FINE|FINER|FINEST} ]  [-d|--dir
<xmlFilesDir> ]
-a|--data <xmlDataFile> -e|--env <xmlEnvFile>

Verbose usage :

ClmgrExport -h|--help [-v|--verbose ]
  -h|--help : to display help.
  [-v|--verbose] : with or without verbose.

ClmgrExport -x|--export [-i|--isd]  [-D|--Debug {0|1|2|3}]
[-L|--Level {SEVERE|WARNING|INFO|CONFIG|FINE|FINER|FINEST} ]  [-d|--dir
<xmlFilesDir> ]
-a|--data <xmlDataFile> -e|--env <xmlEnvFile>
  -x|--export : to export configuration to xml files.
  [-i|--isd] : to indicate the command is launched from ISD.
  [-D|--Debug {0|1|2|3} ]
          0 for no trace info,
          1 for trace to Console,
          2 for trace to file /tmp/export_output.txt,
          3 for both.
   [ -L|--Level : {SEVERE|WARNING|INFO|CONFIG|FINE|FINER|FINEST} ]
          logger level
   [ -d|--dir <xmlFilesDir>] : xml files dir, default is /tmp ]
  -a|--data <xmlDataFile> : xml file containing the data.
  -e|--env <xmlEnvFile> : xml file containing the environment.
```

ClmgrExport : Export configuration to xml files

```
java -DCAS_AGENT=/var/opt/tivoli/ep -cp
/usr/es/sbin/cluster/utilities/clmgrutility.jar
com.ibm.director.power.ha.systemmirror.agent.impl.ClmgrExport
-x -e myenv.xml -a mydata.xml
Running currently "ClmgrExport -x -d /tmp -a mydata.xml -e myenv.xml"
Successfully exported to xmlEnvFile  /tmp/myenv.xml
Successfully exported to xmlDataFile /tmp/mydata.xml
```

ClmgrExport : Export configuration to xml files

```
java -DCAS_AGENT=/var/opt/tivoli/ep -cp
/usr/es/sbin/cluster/utilities/clmgrutility.jar
com.ibm.director.power.ha.systemmirror.agent.impl.ClmgrExport
-x -d /tmp/xml/planning/ -e myenv.xml -a mydata.xml
Running currently "ClmgrExport -x -d /tmp/xml/planning/ -a mydata.xml -e
myenv.xml"
Successfully exported to xmlEnvFile  /tmp/xml/planning//myenv.xml
Successfully exported to xmlDataFile /tmp/xml/planning//mydata.xml
```

ClmgrDeploy

ClmgrDeploy Help

```
java -DCAS_AGENT=/var/opt/tivoli/ep -cp
/usr/es/sbin/cluster/utilities/clmgrutility.jar
com.ibm.director.power.ha.systemmirror.agent.impl.ClmgrDeploy --help --verbose

ClmgrDeploy -h -v
```

```
Usage :

ClmgrDeploy -h|--help [-v|--verbose]

ClmgrDeploy -x|--xml    [-i|--isd]  [-D|--Debug {0|1|2|3}]
[-L|--Level {SEVERE|WARNING|INFO|CONFIG|FINE|FINER|FINEST} ]
[-d|--dir <xmlFilesDir> ] -a|--data <xmlDataFile> -e|--env <xmlEnvFile>

ClmgrDeploy -c|--create [-i|--isd]  [-D|--Debug {0|1|2|3}]
[-L|--Level {SEVERE|WARNING|INFO|CONFIG|FINE|FINER|FINEST} ]
[ -r|--restrict {0|1|2|3|12|13|123} ]  [-d|--dir <xmlFilesDir> ]
-a|--data <xmlDataFile> -e|--env <xmlEnvFile>

Verbose usage :

ClmgrDeploy -h|--help [-v|--verbose ]
  -h|--help : to display help.
  [-v|--verbose] : with or without verbose.

ClmgrDeploy -x|--xml [-i|--isd]  [-D|--Debug {0|1|2|3}]
[-L|--Level {SEVERE|WARNING|INFO|CONFIG|FINE|FINER|FINEST} ]
[-d|--dir <xmlFilesDir> ] -a|--data <xmlDataFile> -e|--env <xmlEnvFile>

  -x|--xml : to display contents of xml files, and check them, without deploying
them.
  [-i|--isd] : to indicate the command is launched from ISD.
  [-D|--Debug {0|1|2|3} ]
      0 for no trace info,
      1 for trace to Console,
      2 for trace to file /tmp/deploy_output.txt,
      3 for both.
  [ -L|--Level : {SEVERE|WARNING|INFO|CONFIG|FINE|FINER|FINEST} ]
         logger level
  [ -d|--dir <xmlFilesDir>] : xml files dir, default is /tmp ]
  -a|--data <xmlDataFile] : xml file containing the data.
  -e|--env <xmlEnvFile] : xml file containing the environment.

ClmgrDeploy -c|--create [-i|--isd]  [-D|--Debug 0 | 1 | 2 | 3 ]
   [-L|--Level SEVERE | WARNING | INFO | CONFIG | FINE | FINER | FINEST ]
   [ -r|--restrict {0|1|2|3|12|13|123} ]  [-d|--dir <xmlFilesDir> ]
   -a|--data <xmlDataFile> -e|--env <xmlEnvFile>
   -c|--create : to create cluster configuration by deploying contents of xml
files.
  [ -r|--restrict {0|1|2|3|12|13|123} ]  : to restrict creation to one scope.
         0 to create nothing.
         1 to restrict creation to cluster object.
         2 to restrict creation to resource group objects (cluster is supposed to
already exist).
        3 to restrict creation to storage objects (cluster is supposed to already
exist).
         12 to restrict creation to cluster object and resource group objects.
         13 to restrict creation to cluster object and storage objects.
         123 to perfom full creation : cluster object, resource group objects,
storage objects.
  [-i|--isd] : to indicate the command is launched from ISD.
```

```
  [-D|--Debug {0|1|2|3} ]
         0 for no trace info, 1 for trace to Console, 2 for trace to file
/tmp/deploy_output.txt, 3 for both.
  [ -L|--Level : {SEVERE|WARNING|INFO|CONFIG|FINE|FINER|FINEST} ]
         logger level
  [ -d|--dir <xmlFilesDir>] : xml files dir, default is /tmp ]
  -a|--data <xmlDataFile] : xml file containing the data.
  -e|--env <xmlEnvFile] : xml file containing the environment.

ClmgrDeploy: Display contents of xml files
java -DCAS_AGENT=/var/opt/tivoli/ep -cp
/usr/es/sbin/cluster/utilities/clmgrutility.jar
com.ibm.director.power.ha.systemmirror.agent.impl.ClmgrDeploy -x -d /tmp -a
mydata.xml -e myenv.xml

ClmgrDeploy : Deploy contents of xml files
java -DCAS_AGENT=/var/opt/tivoli/ep -cp
/usr/es/sbin/cluster/utilities/clmgrutility.jar
com.ibm.director.power.ha.systemmirror.agent.impl.ClmgrDeploy -c -d /tmp -a
mydata.xml -e myenv.xml
Running currently "ClmgrDeploy -d /tmp -a mydata.xml -e myenv.xml"
 Agent xml api init OK on "4ndc1.aus.stglabs.ibm.com".
 Check consistency OK on "4ndc1.aus.stglabs.ibm.com".
 Cluster "APPLI_CLUSTER" successfully deployed on "4ndc1.aus.stglabs.ibm.com".
Successful deployment on "4ndc1.aus.stglabs.ibm.com".
```

Note: PowerHA console XML deployment can be done using command line. To get help, use this command:

`smcli deploycluster –h –v`

Simulated mode: Offline mode

This section explains the various files that are delivered ready to use, including where they are, what they contain.

Several XML configurations are included:

- Linked cluster with two sites, four nodes
- Stretched cluster with two sites, four nodes
- No site cluster with four 7.1.3 nodes
- No site cluster with four legacy 7.1.2.2 nodes

In this mode, there are two kinds of files:

- Non-modifiable samples into `xml/simu` directory.
- Customized into `xml/custom` directory.

The syntax of the XML file (`powerha_systemmirror.xsd`) contains a full PowerHA data model (58 enumerative types, 70 entity object types, 30 ref entity object types). The naming in the XSD file matches the naming of the **clmgr** command line, and it is very legible, as shown in Example 5-7 on page 94 and Example 5-8 on page 95.

These rules apply to the XML files:

- Contains the same XSD name for the XML env and XML data files
- Shows XSD enumerative types whenever possible
- Uses XSD pattern for IPv4 or IPv6 addresses
- Uses XSD type for each PowerHA object
- Use references as there is no duplication of objects in the XML files

There are two types of XML files for persistence and separation of logic:

- XML env file
- XML data file

XML env file

This file describes the physical environment in which the console runs. The XML env file also contains PowerHA nodes, physical volumes on the nodes, networks, interfaces, users, groups, and so on, as shown in Example 5-7.

Example 5-7 XML sample env file

```
<PowerhaConfig xsi:schemaLocation="http://www.ibm.com/powerhasystemmirror
powerha_systemmirror.xsd">
<PowerhaEnvConfig EnvDiscoveryDate="2013-10-18 19:17:54" XMLFileType="planning"
DataFile="xml/planning/MyPlanning.xml">
<DiscoveredConfig>
   <DiscoveredNodes>
      <Node Name="clio1" Hostname="clio1.coopibm.frec.bull.fr" Oid="136306"
Available="true" Aixlevel="7100-00-03-1115" Edition="ENTERPRISE" Version="7.1.3.0"
/>
      <Node Name="clio2" Hostname="clio2.coopibm.frec.bull.fr" Oid="136745"
Available="true" Aixlevel="7100-00-03-1115" Edition="ENTERPRISE" Version="7.1.3.0"
/>… </DiscoveredNodes>
   <DiscoveredNetworks>… </DiscoveredNetworks>
   <DiscoveredPhysicalVolumes>
      <PhysicalVolume Nodes="clio1" Status="Available" Reference="clio2"
EnhancedConcurrentMode="false" Concurrent="false" Available="2048" Size="2048"
Description="Virtual SCSI Disk Drive" Uuid="27f707cd-f74b-0abf-f2ad-89b47da07f32"
Pvid="00cab572b82a59b5" Type="vdisk" Name="hdisk4"/>
      <PhysicalVolume Nodes="clio2" Status="Available" Reference="clio2"
EnhancedConcurrentMode="false" Concurrent="false" Available="2048" Size="2048"
Description="Virtual SCSI Disk Drive" Uuid="dad5faec-e3ad-cf10-25fd-e871f1a123ee"
Pvid="00cab572b82a56f5" Type="vdisk" Name="hdisk3"/> …</DiscoveredPhysicalVolumes>
…
</DiscoveredConfig></PowerhaEnvConfig></PowerhaConfig>
```

The data is used by the XML mode to create PowerHA configurations.

XML data file

This contains the PowerHA configuration created by the console. The XML data file contains clusters, resource groups, storage, replicated mirror groups, and more, as shown in Example 5-8 on page 95.

Example 5-8 XML sample data file


```
<PowerhaConfig xsi:schemaLocation="http://www.ibm.com/powerhasystemmirror
powerha_systemmirror.xsd">
<PowerhaDataConfig XMLFileType="planning"
EnvFile="xml/planning/discenv_20131018_191754.xml">

<Cluster Name= "MyCluster" Id="1" Type="LC" ControllingNode="clio1"
HeartbeatType="unicast" Version="7.1.3.0" Edition="ENTERPRISE" VersionNumber="15"
State="UNSYNCED"  UnsyncedChanges="true" DeployState="UNDEPLOYED"
SitePolicyFailureAction="FALLOVER" ChangedDate="Thu Oct 24 14:01:44 2013"
CreatedDate="Thu Oct 24 14:01:44 2013" >
   <Site Name="blank_site_of_cluster_THURSDAY" Gid="1" State="STABLE" />
   <Site Name="SITE1" Gid="2"  State="STABLE" >
      <SiteNode NodeName="clio1" Oid="136306"/>
      <RepositoryRef Backup="false" Pvid="00cab572b82a56f5"/>
   </Site>
   <Site Name="SITE2" Gid="3" State="STABLE">
      <SiteNode NodeName="clio2" Oid="136745"/>
      <RepositoryRef Backup="false" Pvid="00cab572b82a59b5"/>
   </Site>
   <ClusterSecurity GracePeriod="21600" PeriodicRefreshRate="86400"
AutoDistribCert="false" SecurityLevel="DISABLE"/>
…
</Cluster></PowerhaDataConfig></PowerhaConfig>
```


> **Note:** One XML data file is linked with one XML env file, and one XML env file can be shared by several XML data files.

Simulated mode: Planning mode

In this mode, real data is used (node names, networks, interfaces, available disks) as discovered from the real environment in an initial, manual step using the **`smcli discoverenv`** command. After the initial discovery, the agents are not contacted again unless a deployment is requested.

A configuration can be prepared by using the console, where it can be adjusted and reviewed to get it ready for later exporting and deployment. The configurations go into the `xml/planning` directory.

A configuration created in Planning mode is deployable on a PowerHA SystemMirror for AIX node in two ways:

- Via the Systems Director console:
  a. Select the cluster.
  b. Click the Deploy menu item.
  c. Choose the node on which to deploy.
- Via the command line:

  ```
  smcli deploycluster –h –v
  ```

5.3 Using SUSE Linux as a KVM guest system

For this section, the IBM Systems Director was installed in a SUSE Linux Enterprise server operating system. The system is a kernel-based virtual machine (KVM) guest on a laptop computer that is running RedHat Enterprise Linux 6.4.

The SUSE Linux KVM guest has the following specifications:

- 2 virtual CPUs
- 4 GB memory
- 40 GB disk space (Virtio driver)
- At least 1 DVD ROM drive (use 2 to have IBM Systems Director DVD and SUSE Linux installation media at the same time)
- 1 network interface, type NAT (Virtio driver)

The operating system for the test installation was SUSE Linux Enterprise Server 11 SP3 i586.

The 32-bit installation usually includes all libraries that are required for the Systems Director installation. When using the 64-bit SUSE Linux Enterprise server, some 32-bit libraries are requested during the installation process, such as this example:

```
/usr/lib/libstdc++.so.5
```

For general information regarding the Systems Director installation, see the *IBM Systems Director 6.3 Best Practices: Installation and Configuration*, REDP-4932.

When the base Systems Director is installed and the latest updates have been applied, install the PowerHA plug-in as described in the “PowerHA SystemMirror for IBM Systems Director” of the IBM Knowledge Center:

http://ibm.co/1kE9i2y

5.4 Importing configurations from stand-alone systems

It is now possible to export a PowerHA SystemMirror configuration from a cluster and import it into Systems Director. There is no longer a need to have a direct connection between the systems. In Planning mode and in Offline mode, you can work with the imported configuration, make changes, and export it for deployment at the cluster.

5.4.1 Minimum versions and overview

These are the minimum versions to use this feature:

- IBM PowerHA SystemMirror 7.1.3
- IBM Systems Director 6.3.3 with the PowerHA_SystemMirror 7.1.3 plug-in
- Java 1.6

Figure 5-11 shows working with configurations of systems that are not connected.

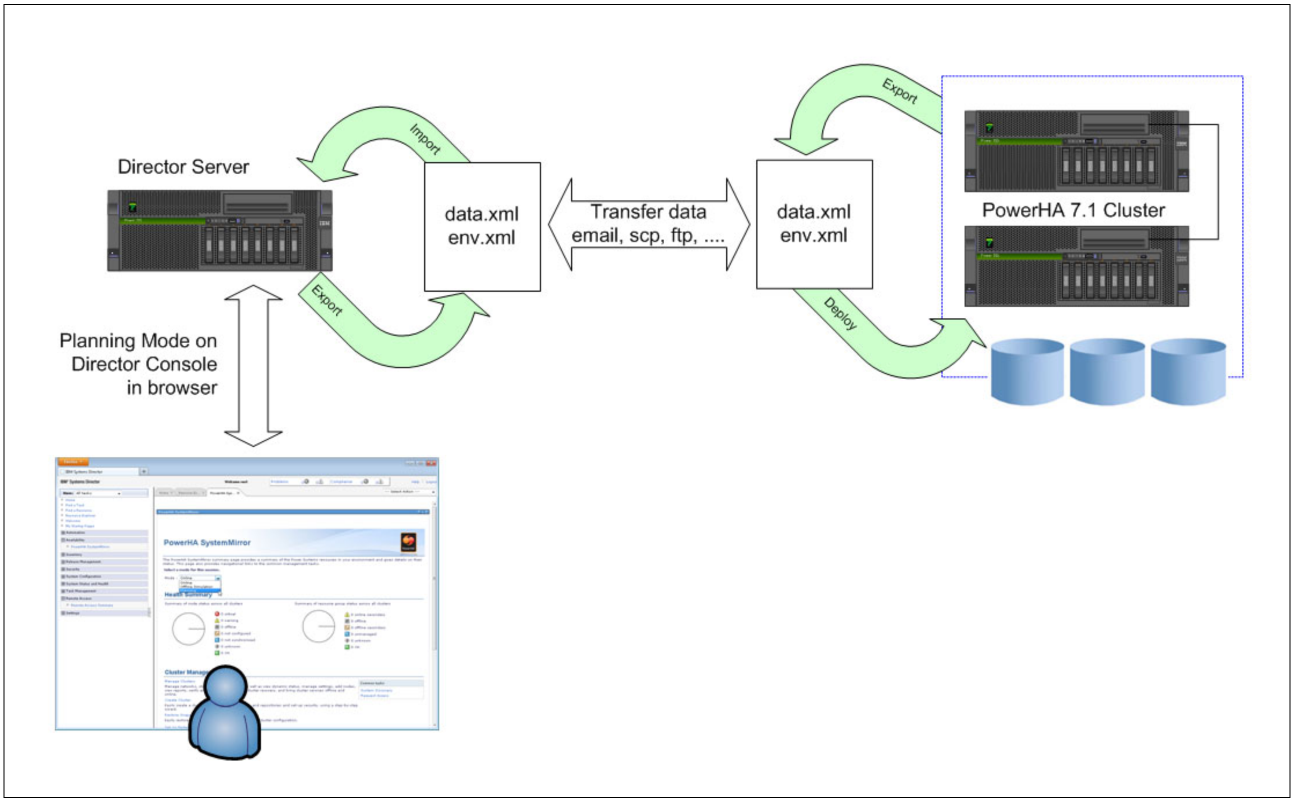


Figure 5-11 Overview diagram

5.4.2 Export and import process steps

You can export the XML configuration by using either the **clmgrutility.jar** utility (Example 5-9) orthe cluster manager **clmgr** command (Example 5-10 on page 98). The **clmgr** command is handy, but you need to specify the `SNAPSHOTPATH` environment variable before executing it. Otherwise, it uses this as the as output path:

```
/usr/es/sbin/cluster/snapshots/
```

Example 5-9 Export using Java

```
root@a2:/> which java
/usr/java6/jre/bin/java
root@a2:/> java -version
java version "1.6.0"
Java(TM) SE runtime Environment (build pap3260sr14-20130705_01(SR14))
IBM J9 VM (build 2.4, JRE 1.6.0 IBM J9 2.4 AIX ppc-32
jvmap3260sr14-20130704_155156 (JIT enabled, AOT enabled)
J9VM - 20130704_155156
JIT  - r9_20130517_38390
GC   - GA24_Java6_SR14_20130704_1138_B155156)
JCL  - 20130618_01
```

```
root@a2:/> mkdir -p /tmp/xml/planning
root@a2:/> cd /tmp
root@a2:/> java -DCAS_AGENT=/var/opt/tivoli/ep -cp \
/usr/es/sbin/cluster/utilities/clmgrutility.jar \
com.ibm.director.power.ha.systemmirror.agent.impl.ClmgrExport \
--export -d /tmp -a xml/planning/mysnap.xml -e xml/planning/mysnap_env.xml
```

Example 5-10 Export using clmgr

```
root@a2:/tmp> SNAPSHOTPATH=/tmp clmgr create snapshot mysnap TYPE=xml

clsnapshot: Creating file /tmp/mysnap.xml.

clsnapshot: Creating file /tmp/mysnap.info.
Running currently "ClmgrExport -x -d /tmp -a mysnap.xml -e mysnap_env.xml"
Successfully exported to xmlEnvFile  /tmp/mysnap_env.xml
Successfully exported to xmlDataFile /tmp/mysnap.xml

clsnapshot: Executing clsnapshotinfo command on node: a2...

clsnapshot: Executing clsnapshotinfo command on node: b2...

clsnapshot: Succeeded creating Cluster Snapshot: mysnap
```

Both XML files must be placed in the following path:

/opt/ibm/director/PowerHASystemMirror/eclipse/plugins/com.ibm.director.power.ha.systemmirror.common_**<VERSION>**/bin/xml/planning

Example 5-11 shows the import of the XML file into the Systems Director database to make it available in the web GUI.

Example 5-11 Importing the XML configuration file

```
SysDir63:~ # smcli modemngt -s -x -p -a xml/planning/mysnap.xml
Trying to set console to xml planning mode, and continue an existing planning
session.

Successful
PowerHA SystemMirror Console settings :
  - Configuration from /opt/ibm/director/data/powerhasystemmirror.{dat|idx} files.
  - Console mode : xml planning mode.
  - xml files dir :
/opt/ibm/director/PowerHASystemMirror/eclipse/plugins/com.ibm.director.power.ha.sy
stemmirror.common_7.1.3.0/bin
  - With env  xml file : xml/planning/mysnap_env.xml
  - With data xml file : xml/planning/mysnap.xml
```

Now, the configuration is available in the Systems Director management GUI. Figure 5-12 on page 99 shows the screen after logging in to the web GUI by clicking **Availability** → **PowerHA SystemMirror** from the left menu.

Change the mode to **Planning**.

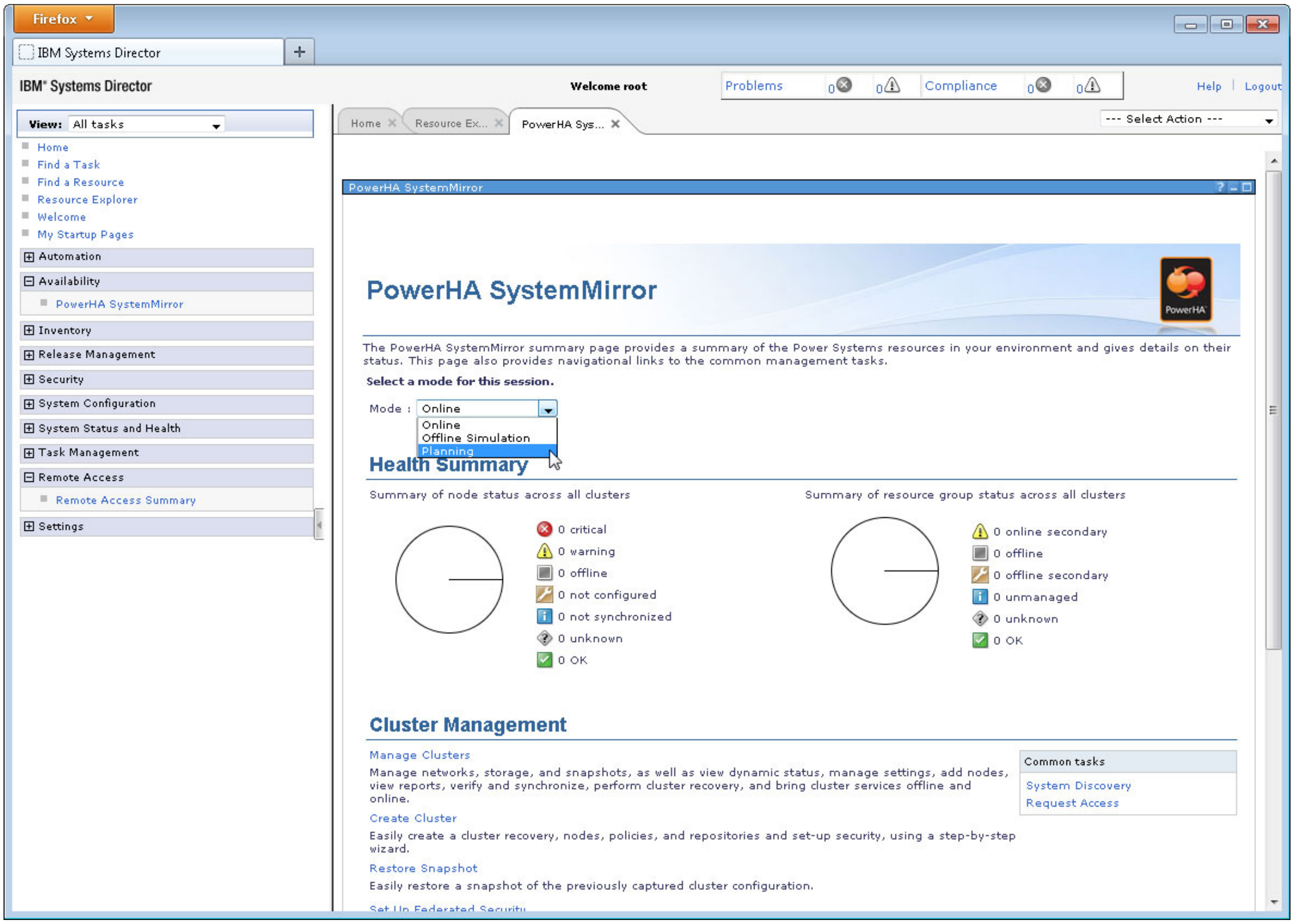


Figure 5-12 Switching to Planning mode

From the drop-down menu, choose your configuration file, as shown in Figure 5-13.

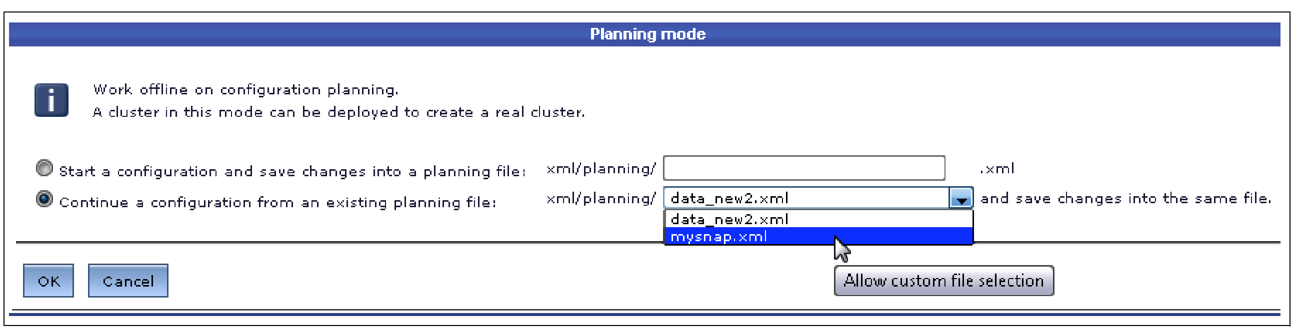


Figure 5-13 Choosing the configuration file

The Planning mode offers several possibilities to display and manage the cluster and resource groups. Figure 5-14 on page 100 shows the displayed configuration in Planning mode after the configuration has been loaded.

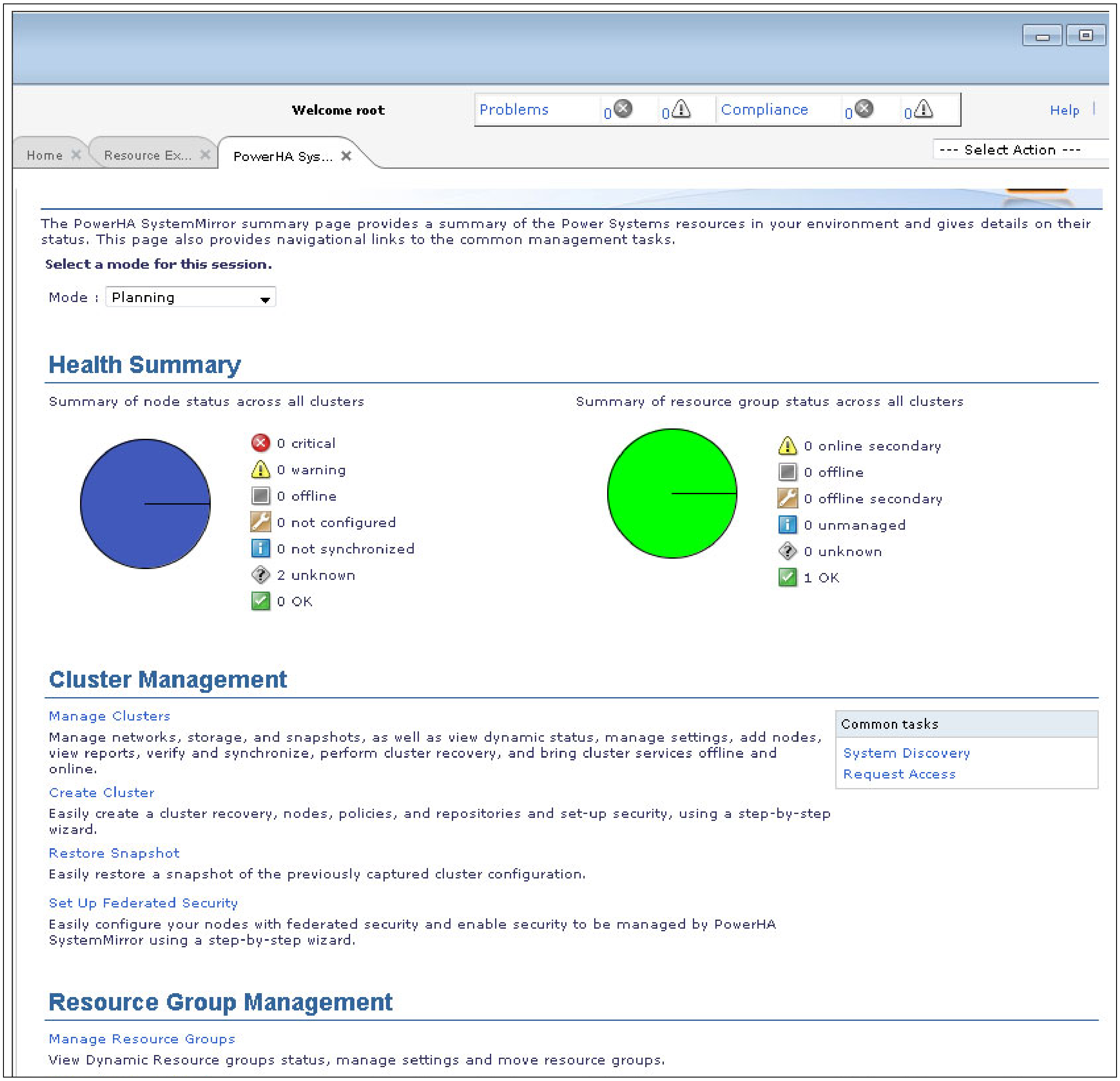


Figure 5-14 Planning mode after configuration is loaded

There is no need to manually copy the XML files to the cluster node. When running in Planning mode, the context menu on the selected cluster has a Deploy action that copies the XML files to the agent node and deploys the cluster.

On the cluster, the new configuration can be deployed as shown in the Example 5-12.

Example 5-12 Deploying the new configuration

```
root@a2:/tmp> rm /tmp/mysnap.infols -ltr /tmp/mys*
-rw-------    1 root     system        3554 Dec 17 07:43 /tmp/mysnap_env.xml
-rw-------    1 root     system        4439 Dec 17 07:43 /tmp/mysnap.xml

root@a2:/tmp> /usr/es/sbin/cluster/utilities/cldare -trjava -DCAS
_AGENT=/var/opt/tivoli/ep -cp  /tmp/clmgrutility.jar
com.ibm.director.power.ha.systemmirror.agent.impl.ClmgrDeploy --create -d /tmp -a
mysnap.xml -e mysnap _env.xml
Running currently "ClmgrDeploy -d /tmp -a mysnap.xml -e mysnap_env.xml"
  Agent xml api init OK on "a2".
  Check consistency OK on "a2".
  Something wrong while deploying resource group(s) and/or storage(s), what has
been deployed is going to be un-deployed.

An exception while creating cluster
     ERROR: the specified object already exists: "sas_itso_cl"
```

6

Implementing DB2 with PowerHA

This chapter describes implementing IBM DB2 clusters with IBM PowerHA SystemMirror. It provides guidance specific to implementing the latest version of PowerHA and covers the following topics:

- Introduction to the example scenario
- Prepare for DB2 v10.5 installation
- Install DB2 v10.5 on AIX 7.1 TL3 SP1
- Prepare the cluster infrastructure
- Create a PowerHA DB2 cluster
- Test DB2 cluster functions

6.1 Introduction to the example scenario

In this section, we describe best practices for IBM DB2 v10.5 high availability configuration using IBM PowerHA SystemMirror 7.1.3, with sample configuration details. All scenarios in this section are built with IBM AIX 7.1 TL3 SP1. It is strongly recommended that you have all latest AIX Technology Levels (TLs) and Service Packs (SPs) before proceeding with a DB2 server installation.

Although this book focuses on PowerHA mechanisms and techniques, DB2 high availability is shown here without the *DB2 Disaster Recovery* support module.

> **Note:** For more information about IBM DB2 high availability and disaster recovery options, see the IBM Redbooks publication titled *High Availability and Disaster Recovery Options for DB2 for Linux, UNIX, and Windows*, SG24-7363:
>
> http://publib-b.boulder.ibm.com/abstracts/sg247363.html?Open

DB2 installation in a cluster requires many prerequisites for proper design and operation. For this scenario, a basic two-node cluster was built to perform all DB2 drills and high availability testing.

As Figure 6-1 shows, DB2 services are designed to work on *DB2 Server01* server while *DB2 Server02* is in standby with no services running. In case of a planned outage, such as maintenance on DB2 Server01, or an unplanned one, such as a hardware failure on DB2 Server01, PowerHA mechanisms automatically switch the DB2 services to DB2 Server02 to reduce the service outage duration for users.

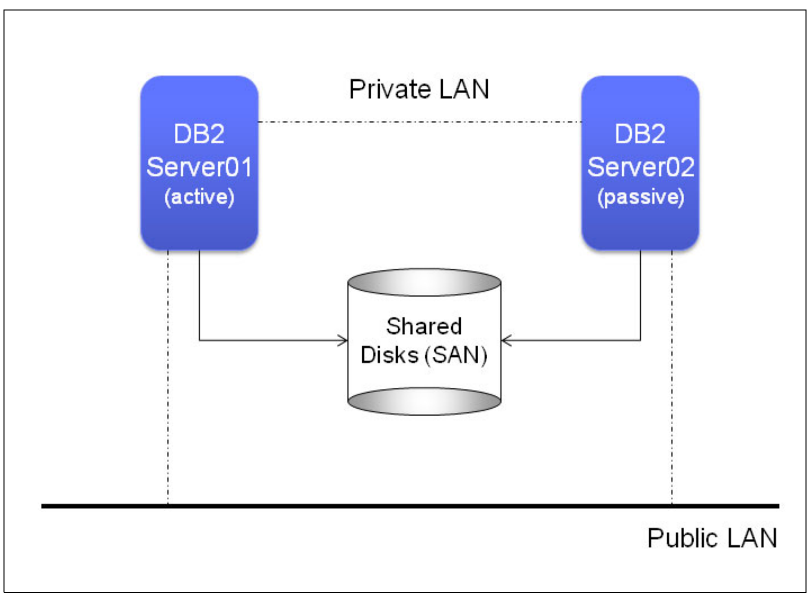


Figure 6-1 Basic two-node DB2 cluster

> **Important:** Using DB2 high availability with PowerHA does not cover an outage caused by a total data loss on the shared storage end. If the solution design requires resilience for data loss, a storage synchronization solution or DB2 High Availability and Disaster Recovery (DB2 HADR) must be considered.

6.2 Prepare for DB2 v10.5 installation

Preparation for DB2 installation depends directly of the environment design being deployed. Anyway, some standard custom configuration must be considered in all landscapes, specially those related to memory use, network tuning, and file system sizes that are normally required to open a support ticket with IBM technical support.

Note: Remember that all changes must be applied on all cluster nodes.

6.2.1 Memory parameters

When DB2 keeps its own data cache control, it is important to keep AIX Virtual Machine Manager (VMM) in a good shape. To reach that goal, some suggested parameters are defined for systems that are hosting DB2 services. The parameters listed in this section are changed by using the **vmo** command. For more information about its use, see the **man vmo** command help pages.

The first recommended memory-related parameter is **min_free**. In general, the **min_free** parameter defines the number of free memory frames in the VMM list when VMM starts stealing pages from memory. It keeps a free list in a health state.

A standard DB2 recommendation for the min_free value is 4096 for systems with less than 8 GB of RAM memory and 8192 for systems with more than 8 GB of RAM memory.

The second memory-related parameter is **max_free**. This parameter defines the number of free pages on the VMM free list, where VMM stops stealing pages from memory. A generic recommendation for this parameter is **min_free** + 512, which means if **min_free** was defined as 4096, for example, **max_free** should be 40608.

With the standard values for **maxperm%** (90), **maxclient%** (90), **strict_maxclient** (1), **minperm%** (3), and **lru_file_repage** (0), no changes are required unless any specific performance behavior is detected that requires customization of these parameters.

6.2.2 Network parameters

Even with the AIX operating system having general parameters appropriate to most production environments, some changes are recommended related to bandwidth use enhancements and security. Some of the recommended factors are described in the following sections. The parameters that follow are changed with the **no** command. For more about its use, see the **man no** command help pages.

- **tcp_sendspace**: Defines the kernel buffer size for sending applications before applications are blocked by a send call. The recommended standard value for DB2 is 262144.
- **tcp_recvspace**: Specifies the number of receiving bytes allowed to be buffered in the kernel. The recommended standard value is 262144.
- **ipqmaxlen**: Defines the number of packages that can be queued in the IP input queue. The recommended standard value is 250.
- **tcp_nagle_limit**: The AIX operating system, by default, tries to consolidate many packages before sending. Changing this parameter enhances the real-time data transfer rate but causes a higher overhead for network operations. The recommended standard value is 1 (disabling packages grouping or consolidating before sending).

- **rfc1323**: This parameter enables the communication enhancement proposed by RFC 1323, increasing the TCP window scale from the default 64 KB to 1 GB. The recommended value for this parameter is 1 (enable).
- **tcp_nodelayack**: This parameter also increases system overhead, but it enhances network communication, sending immediate ACK packets for requisitions. The recommended value for this parameter is 1 (enable).
- **clean_partial_connection**: This is a parameter for security enhancement. Partial connections are used to fulfill the queue backlog, causing a denial of service (SYN ACK attack). By avoiding partial connections, possible SYN ACK attack packets can be blocked. The recommended value for this parameter is 1 (enable).
- **tcp_tcpsecure**: This is intended to protect systems from some network vulnerabilities. For example:
  - Using a numeral 1 as parameter value protects the systems from a vulnerability where a fake SYN package is sent by an attacker, aborting an established connection on the DB2 server.
  - Using a 2 mitigates a vulnerability where a fake RST (reset) package is sent by an attacker.
  - Using a 4 protects from injecting fake DATA information on an established connection.

  The **tcp_tcpsecure** parameter allows a combination of options, so possible options are 1, 2, 3, 4, 5, 6, and 7. The recommended standard value is 5.
- **ipignoreredirects**: Internet Control Message Protocol (ICMP) redirect packages and notifies a sender to resend its packets from alternative routes. When this option is enabled, the AIX operating system ignores redirection of ICMP packages to protect the systems from malicious ICMP packages that are intended to create manipulated routes. The recommended value is 1 (enable).

6.2.3 Asynchronous I/O operations on AIX

Until AIX 5.3 TL5, asynchronous operations were performed by AIX asynchronous I/O devices (aio), working at thread level. AIX 5.3 TL5 introduced a deeper level that uses the I/O Completion Port (IOCP) API. So, for DB2 v10.5 installation and proper operation, IOCP devices must be configured and enabled.

To configure IOCP on AIX 7.1, first check the actual device state, as shown in Example 6-1.

Example 6-1 Checking IOCP device state

```
root@jazz(/)# lsdev -Cc iocp
iocp0 Defined  I/O Completion Ports
root@jazz(/)#
```

As can be seen on Example 6-1, the initial state of the IOCP device on AIX 7.1 is as defined. Therefore, before start installing DB2, IOCP must be configured. Type **smitty iocp** → **Change / Show Characteristics of I/O Completion Ports** to configure it. For the “STATE to be configured at system restart” field, select the *available* option, as shown in Example 6-2 on page 107.

Example 6-2 Configuring the IOCP device

```
Change / Show Characteristics of I/O Completion Ports

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

  STATE to be configured at system restart           available
```

> **Note:** In case the iocp0 device stays as *defined* even after the procedure described above, enter the **`mkdev -l iocp0`** command as root and check it again. If it remains as defined, reboot the system.

6.2.4 Paging space area

For DB2 systems, the general recommendation related to the paging space area is to have a paging area equal to twice the RAM size. Even if it sounds large for larger systems (with 2 TB of RAM, for example), it is a basic recommendation for DB2 systems. But the exact paging space area must be calculated by the DB2 administrator. Some DB2 environments do not use paging areas.

As a best practice in a DB2 system, the same paging space areas organization guidelines can be followed:

- The default paging space (hd6) stored on rootvg with at least 512 MB size.
- Multiple paging spaces stored across all available disks with a size of up to 64 GB each.
- It is highly recommended to have only 1 (one) paging space area per disk.

6.2.5 DB2 groups and users

Before starting the installation, all required users and groups must be created on all cluster nodes. These are the main users to be created:

- An instance owner (for the scenario in this book, `db2inst1`)
- A fenced user (`db2fenc1`)
- The DB2 administration server user, DAS (`dasusr1`)

It is recommended that each of these users, due to their special rights, be in a dedicated group to make administration of permissions easy. For this example, we created the following groups:

- `db2iadm1` for the instance owner
- `db2fadm1` for the forced user
- `dasadm1` for the DAS user

Because the servers being deployed are intended to be part of a cluster sharing configuration, it is important to create all users' home directories on the shared disks to make sure that any modifications for users' data are reflected and accessible for all cluster nodes.

DB2 users limits

To make sure that all DB2 operations work properly, all resource limitations (*user limits*) must be properly configured as shown in Example 6-3 on page 108. The configuration can be set manually by the root user, using the **`ulimit`** command, but it is strongly recommended to include it in the `/etc/security/limits` operating system file.

For scenario purposes, all DB2-related users (`db2inst1`, `db2fenc1`, and `dasusr1`) have their limits set to *unlimited*. But this must be carefully designed for each production environment.

Example 6-3 Changing user limits for DB2-related users

```
chuser fsize=-1 fsize_hard=-1 data=-1 data_hard=-1 stack=-1 stack_hard=-1 rss=-1
rss_hard=-1 nofiles=-1 nofiles_hard=-1 db2inst1
chuser fsize=-1 fsize_hard=-1 data=-1 data_hard=-1 stack=-1 stack_hard=-1 rss=-1
rss_hard=-1 nofiles=-1 nofiles_hard=-1 db2fenc1
chuser fsize=-1 fsize_hard=-1 data=-1 data_hard=-1 stack=-1 stack_hard=-1 rss=-1
rss_hard=-1 nofiles=-1 nofiles_hard=-1 dasusr1
```

6.2.6 Cluster IP addresses

To avoid incongruence in the cluster configuration, it is mandatory to have all cluster IP addresses properly defined previously and to copy them to the `/etc/hosts file` on all cluster nodes.

For this scenario, the IP addresses shown in Example 6-4 were defined as to be used in the cluster as cluster IP addresses.

Example 6-4 P addresses defined to be used as cluster IP addresses

```
root@blues(/)# cat /etc/hosts
27.0.0.1               loopback localhost      # loopback (lo0) name/address
::1                     loopback localhost      # IPv6 loopback (lo0) name/address

# Cluster addresses

129.40.119.203  blues db2host
129.40.119.225 jazz
172.10.10.203 bluespriv
172.10.10.225 jazzpriv
172.10.10.237 cluster01priv
129.40.119.237 cluster1
```

6.2.7 Cluster disks, volume groups, and file systems

All DB2- and PowerHA-related data must rely on external disks to be accessible to all cluster nodes. For this scenario, five SAN disks were attached to both cluster nodes: one 10 GB disks for CAA use and four 30 GB disks for DB2 data, as shown in Example 6-5.

Example 6-5 Disks assignment to DB2 cluster

```
root@blues(/)# lspv
hdisk0          00f623c5527a212a                    rootvg          active
hdisk1          00f623c591941681                    None
hdisk2          00f623c58fab0ef6                    db2vg
hdisk3          00f623c58fab0fb5                    db2vg
hdisk4          00f623c5919415d4                    None
hdisk5          00f623c59194147c                    None

root@blues(/)# for i in 1 2 3 4 5^Jdo echo "hdisk$i - size in MB: `bootinfo -s
hdisk$i`"^Jdone
hdisk1 - size in MB: 10240
```

```
hdisk2 - size in MB: 30720
hdisk3 - size in MB: 30720
hdisk4 - size in MB: 30720
hdisk5 - size in MB: 30720
```

After the disks are assigned and recognized, an enhanced capable volume group and a JFS2 file systems are created by using these shared disks. All file systems are created with no auto mount, and the volume group is defined with auto varyon as Off, as shown in Example 6-6. The 10 GB disk became untouched and is used by CAA during cluster creation.

Example 6-6 Creating DB2 volume group and file systems

```
root@blues(/)# mkvg -S -s 512 -V 65 -y db2vg hdisk2 hdisk3 hdisk4 hdisk5
db2vg
root@blues(/)# varyoffvg db2vg
root@blues(/)# chvg -an db2vg
root@blues(/)# varyonvg db2vg

root@blues(/)# mklv -t jfs2 -y db2inst1lv db2vg 40
root@blues(/)# mklv -t jfs2 -y db2fenc1lv db2vg 20
root@blues(/)# mklv -t jfs2 -y dasusr1lv db2vg 20

root@blues(/)# crfs -v jfs2 -d /dev/db2inst1lv -m /db2/db2inst1 -A no -p rw
root@blues(/)# crfs -v jfs2 -d /dev/db2fenc1lv -m /db2/db2fenc1 -A no -p rw
root@blues(/)# crfs -v jfs2 -d /dev/dasusr1lv -m /db2/dasusr1 -A no -p rw
```

Note: To avoid SCSI locking issues, it is important to define cluster shared disks with no reservation policy:

```
root@blues(/)# chdev -l hdisk4 -a reserve_policy=no_reserve
hdisk4 changed
```

6.3 Install DB2 v10.5 on AIX 7.1 TL3 SP1

In this section, we describe the DB2 v10.5 installation as performed for this scenario. Remember that this book focuses on PowerHA SystemMirror 7.1.3, so no deep configurations for DB2 were performed (only the minimal required to have a functional DB2 instance).

First, the DB2 installation images must be downloaded from the DB2 for Linux, UNIX and Windows web page:

http://www.ibm.com/software/data/db2/linux-unix-windows/downloads.html

The **Download** and **DB2 for 90 Days** options must be selected.

Note: In case you do not have a valid DB2 license to apply, the trial license is automatically activated. The software then works for only 90 days.

During this installation scenario, the use of the *instance owner* user during the installation process is chosen to make the process simpler. But keep the considerations in Table 6-1 on page 110 in mind when choosing the user ID to do the installation (root or non-root ID).

Table 6-1 Differences between root and non-root user installation

Criteria	installation using ROOT user	installation using non-ROOT
Select installation directory	Yes	No
Number of instances	Multiple	Only one
Files deployed	Binaries only	Binaries and instance files
Upgrade version and instance	No	Version and instance together

After downloading and uncompressing the installation images, as instance owner user (db2inst1), change to the directory where the images are copied and run the **db2_install** installation command, as shown in Figure 6-2.

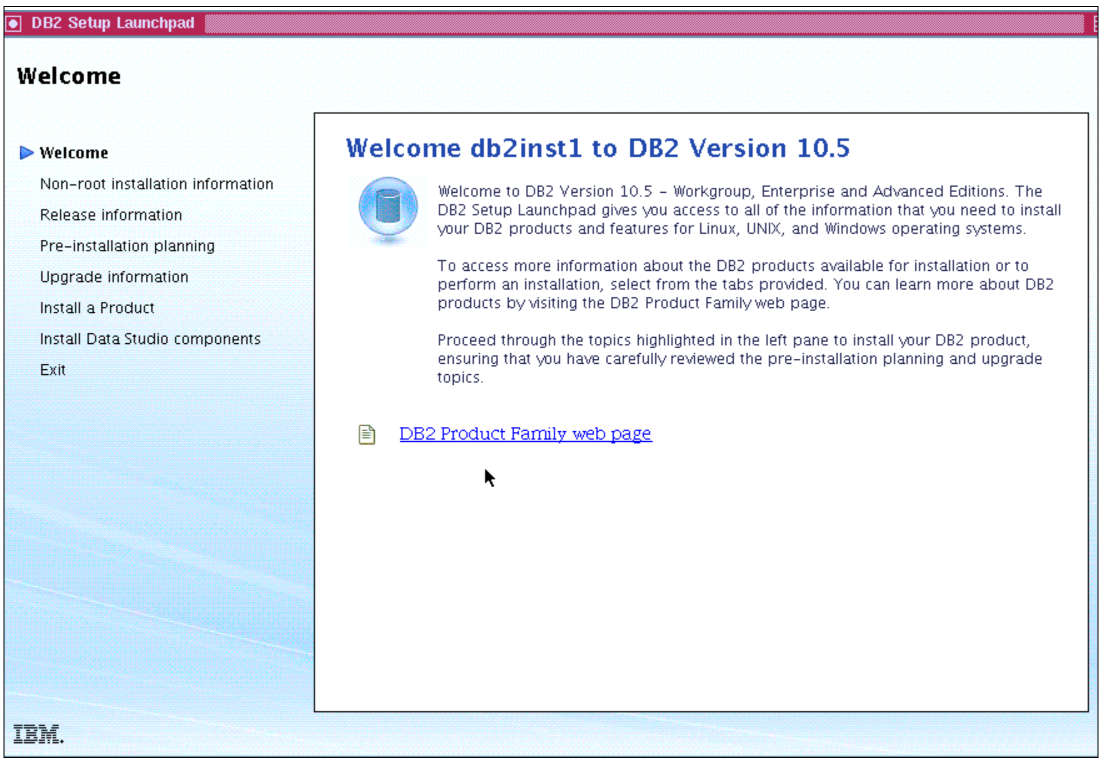


Figure 6-2 Starting the DB2 installation

Click the **Install a Product** option on the left menu.

Then, from the list, click the **Install New** button, which is just after the “DB2 Version 10.5 Fix Pack 2 Workgroup, Enterprise, and Advanced Editions” text section.

Figure 6-3 on page 111 shows the initial installation screen for the DB2 Enterprise Server installation.

Click **Next**.

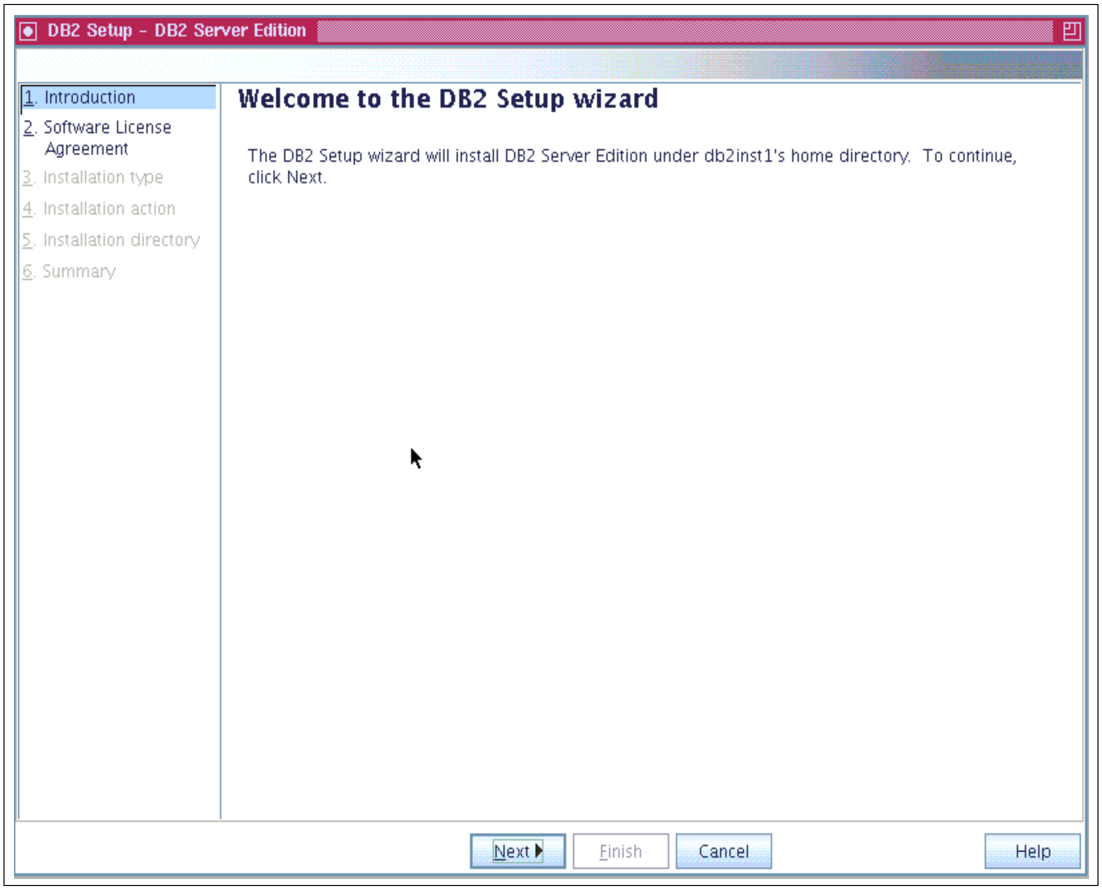


Figure 6-3 Initial DB2 Enterprise Server installation window

In the License Agreement window, mark the option to accept it, and then click **Next** again. as shown in Figure 6-4.

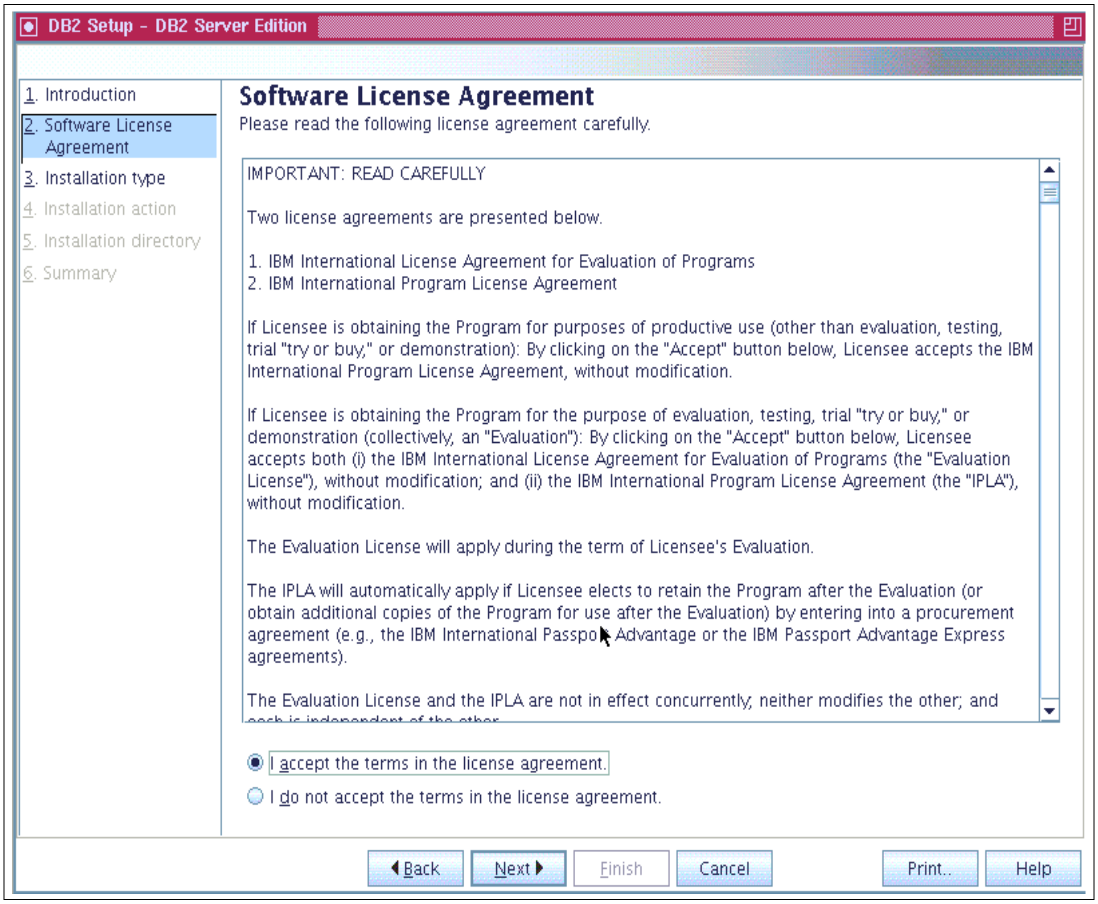


Figure 6-4 DB2 installation license agreement

In the next window, the installation type must be selected. For this scenario, we chose **Typical**, as shown in Figure 6-5 on page 113.

Click **Next**.

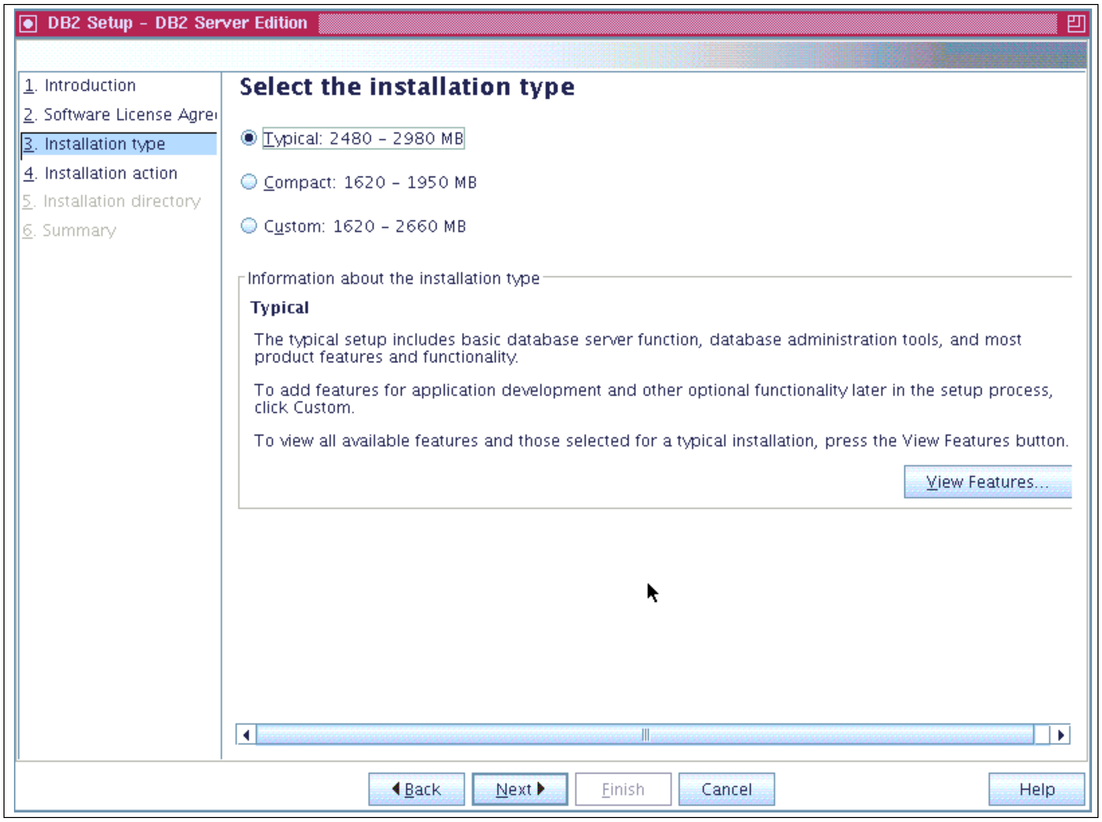


Figure 6-5 DB2 installation type options

In the next window, you can choose either to install the server or to create a response file. For this scenario, we chose **Install DB2 Server Edition on this computer**, as shown in Figure 6-6 on page 114.

Then, click **Next.**

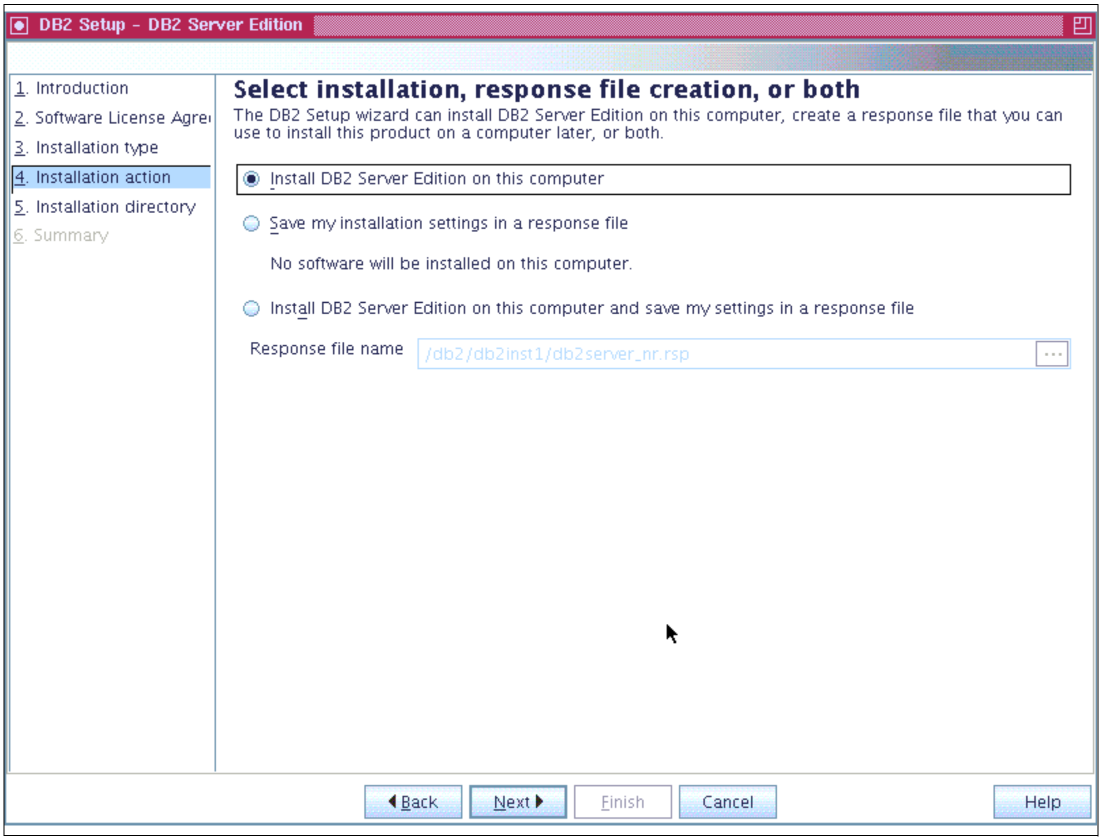


Figure 6-6 DB2 installation and response file creation

In the next window, the installation directory is chosen, as shown in Figure 6-7 on page 115. This installation scenario was performed with a non-root user, so the installation directory is automatically defined as the installation user's home directory (/db2/db2inst1).

Click **Next.**

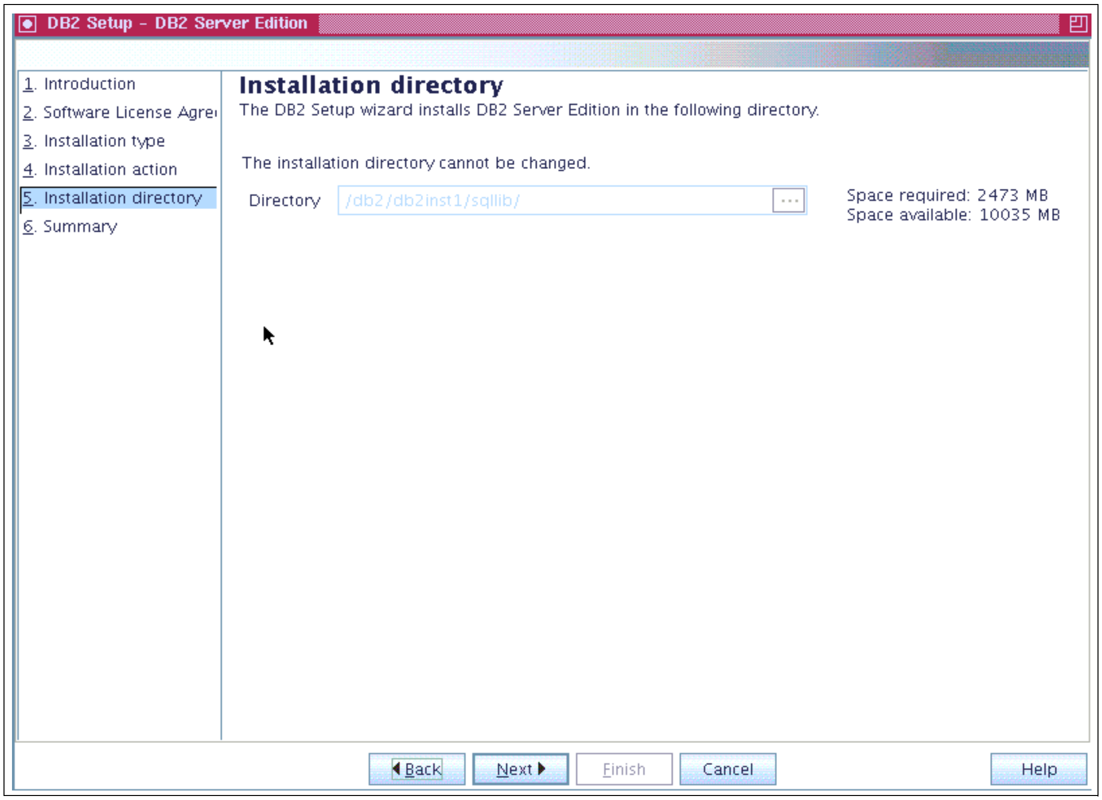


Figure 6-7 Choosing the DB2 installation directory

The next window shows an Installation Summary (Figure 6-8). If incorrect information appears, click **Back** and correct it. If everything is fine, click **Finish** to begin the installation process.

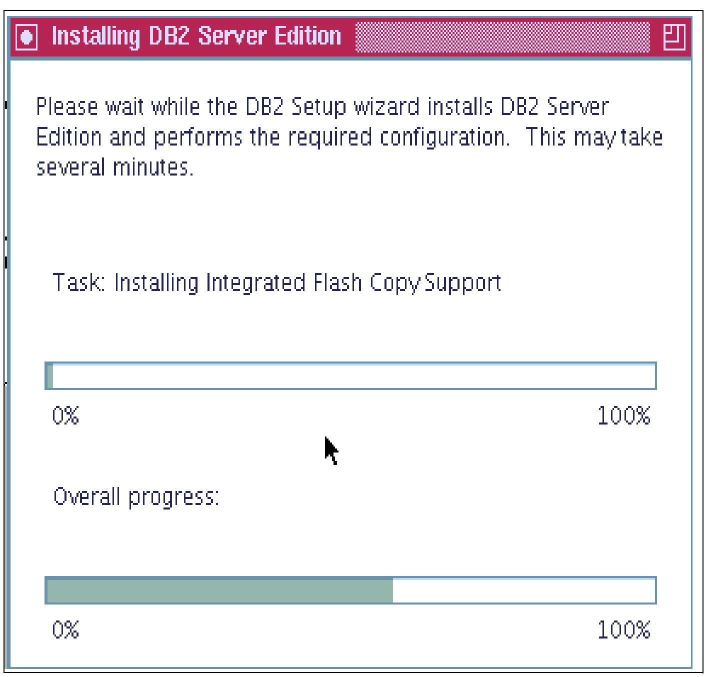


Figure 6-8 DB2 installation in progress

After the installation is finished, just click **Finish** to close the window.

6.3.1 Create a sample database for scenario validation

When the installation is done with the graphical user interface (GUI), it automatically opens the First Steps panel, as shown in Figure 6-9. Inside it, click **Create Sample Database** to start the process. A new window opens showing that new database is created under the installed instance. Click **OK**.

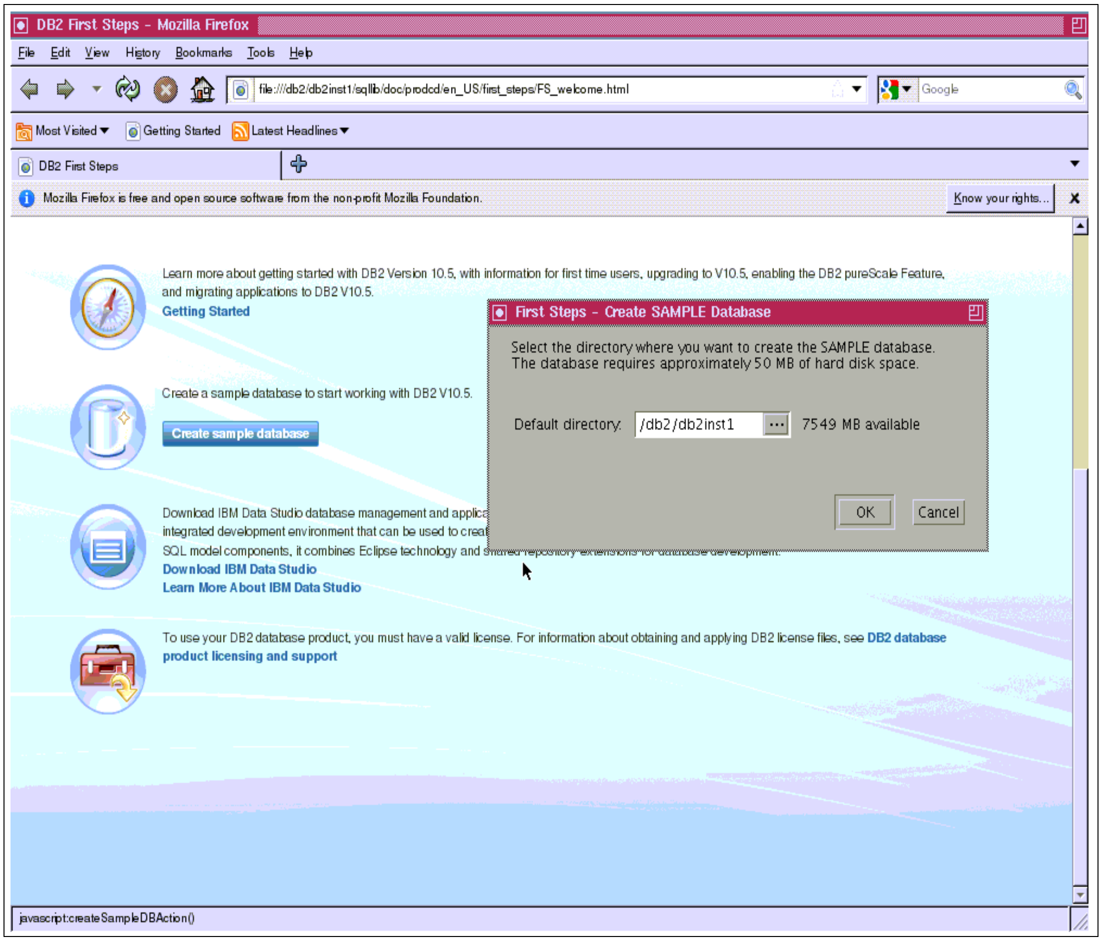


Figure 6-9 DB2 First Steps screen

After the process is finished, the new sample database can be checked by looking at the output of **`db2 list database directory`** command, run as the `db2inst1` user, as shown in Example 6-7.

Example 6-7 Checking sample database creation

```
db2inst1@blues(/usr/bin)$ db2 list database directory

System Database Directory
Number of entries in the directory    = 1
Database 1 entry:
Database alias                        = SAMPLE
Database name                         = SAMPLE
Local database directory              = /db2/db2inst1
Database release level                = 10.00
```

```
Comment                                =
Directory entry type                   = Indirect
Catalog database partition number      = 0
Alternate server hostname              =
Alternate server port number           =
```

6.3.2 Validate DB2 accessibility

To verify that the DB2 services are running properly, try to perform a query on the sample database, as shown in Example 6-8.

Example 6-8 Testing SAMPLE DB2 database

```
root@blues(/)# su - db2inst1
db2inst1@blues(/db2/db2inst1)$ db2 connect to sample

   Database Connection Information

 Database server        = DB2/AIX64 10.5.2
 SQL authorization ID   = DB2INST1
 Local database alias   = SAMPLE

db2inst1@blues(/db2/db2inst1)$ db2 select PID,NAME from product

PID     NAME
-----------------------------------
100-100-01 Snow Shovel, Basic 22 inch
100-101-01 Snow Shovel, Deluxe 24 inch
100-103-01 Snow Shovel, Super Deluxe 26 inch
100-201-01 Ice Scraper, Windshield 4 inch

  4 record(s) selected.
```

6.4 Prepare the cluster infrastructure

In the scenario built for this book, all resources (CPU, memory, I/O, and network) are virtual, using Virtual I/O Servers, as shown in Figure 6-10 on page 118. But to be specifically for PowerHA, it is recommended to keep at least two virtual adapters to make sure that PowerHA does not miss any network events.

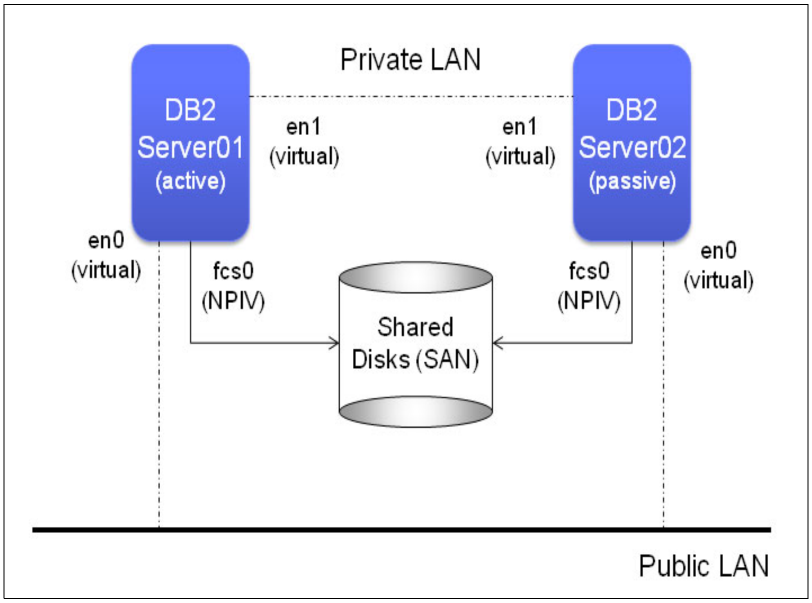


Figure 6-10 Virtual adapters on DB2 cluster scenario

6.4.1 Service IP address on the DB2 PowerHA cluster

One common issue during DB2 deployment on a cluster environment is how to make DB2 run properly on all cluster nodes. DB2 relies on the `db2nodes.cfg` file configuration and requires that the hostname in this file be a valid hostname for a node starting DB2 services. This step must be carefully planned.

After the DB2 installation, the `db2nodes.cfg` file has the local hostname information, as shown in Example 6-9.

Example 6-9 db2nodes.cfg file after DB2 installation

```
db2inst1@blues(/db2/db2inst1)$ cat sqllib/db2nodes.cfg
0 blues 0
db2inst1@blues(/db2/db2inst1)$
```

There are many ways to accomplish the configuration through DB2, PowerHA, and AIX mechanisms. In this section, we explain several of the options.

Option 1: Use a host alias in the /etc/hosts file

The simplest way to accomplish DB2 services running on all cluster nodes is using a host alias as DB2 service hostname. This option is extremely useful if the production environment where DB2 high availability is being configured does not allow the use of SSH or RSH.

Basically, an alias must be added after the local hostname in the hosts file, and then the same alias is added to the `db2nodes.cfg` configuration file as shown in Example 6-10 on page 119.

Example 6-10 Host alias for DB2 services

```
root@blues(/)# cat /etc/hosts | grep db2service
129.40.119.203  blues db2service
root@blues(/)#

db2inst1@blues(/db2/db2inst1/sqllib)$ cat db2nodes.cfg
0 db2service 0
```

Now DB2 can be correctly started on all cluster nodes with no major changes as shown in Example 6-11.

Example 6-11 Starting DB2 services on the cluster nodes

```
First cluster node - blues
db2inst1@blues(/db2/db2inst1/sqllib)$ cat db2nodes.cfg
0 db2service 0

db2inst1@blues(/db2/db2inst1/sqllib)$ db2start
12/12/2013 09:28:45     0   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.

db2inst1@blues(/db2/db2inst1/sqllib)$ ps -fe | grep db2
db2inst1 10354906 18153532   0 09:28:44      -  0:00 db2vend
db2inst1 12714234 19660972   0 09:28:44      -  0:00 db2ckpwd 0
db2inst1 14221450 19660972   0 09:28:44      -  0:00 db2ckpwd 0
db2inst1 14680294 19923128   0 09:28:49  pts/0  0:00 grep db2
db2inst1 17367170 19660972   0 09:28:44      -  0:00 db2ckpwd 0
db2inst1 18153532        1   0 09:28:44      -  0:00 db2wdog 0 [db2inst1]
db2inst1 19660972 18153532   1 09:28:44      -  0:00 db2sysc 0
db2inst1 21561550 18153532 120 09:28:45      -  0:02 db2acd
Second node - jazz
db2inst1@jazz(/db2/db2inst1/sqllib)$ cat db2nodes.cfg
0 db2service 0

db2inst1@jazz(/db2/db2inst1/sqllib)$ db2start
12/12/2013 09:37:08     0   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.

db2inst1@jazz(/db2/db2inst1/sqllib)$ ps -fe | grep db2
db2inst1  8650898 12255338   0 09:37:06      -  0:00 db2vend
db2inst1 11010052 11993222   0 09:37:06      -  0:00 db2ckpwd 0
db2inst1 11075620 11993222   0 09:37:06      -  0:00 db2ckpwd 0
db2inst1 11272304 11993222   0 09:37:06      -  0:00 db2ckpwd 0
b2inst1 11927748 12255338   0 09:37:08      -  0:03 db2acd
db2inst1 11993222 12255338   0 09:37:06      -  0:00 db2sysc 0
db2inst1 12255338        1   0 09:37:06      -  0:00 db2wdog 0 [db2inst1]
```

Option 2: Use PowerHA scripts to modify the db2nodes.cfg file

One disadvantage of **/etc/hosts** file editing is the need of keep checking if the hosts file, outside cluster environment, is still healthy and properly configured. Any configuration loss may cause larger disruptions during cluster resources movement.

An alternative for this, is to keep the file management inside the cluster scripts coding. So, instead of manually adding a hostname alias inside **/etc/hosts**, the **db2nodes.cfg** itself can be directly updated on all startup operations performed by the PowerHA cluster.

Basically, the **db2nodes.cfg** file has a simple standard as shown in Example 6-12.

Example 6-12 basic db2nods.cfg file format

```
<nodenumber> <hostname> <logical port>
```

Where **nodenumber** represents the unique ID for a database server (default is 0), hostname represents the server (according to the **/etc/hosts** file), and the logical port represents the database partition (default is 0).

Considering a two node cluster composed by **hosts** blues and **jazz**, the PowerHA scripts must dynamically generated these two **db2nodes.cfg** files variations as shown in Example 6-13.

Example 6-13 db2nodes.cfg file versions for specific cluster nodes

```
when cluster is starting at blues node:
0 blues 0

when cluster is starting at jazz node:
0 jazz 0
```

Option 3: Use the db2gcf command

Rather than changing the `/etc/hosts` file, you can use the **db2gcf** internal command. This dynamically changes the DB2 service host name each time that DB2 services are started by PowerHA scripts, even when each cluster node is performing the startup as shown in Example 6-14.

Example 6-14 Changing DB2 service hostname by using the db2gcf command

```
db2inst1@jazz(/db2/db2inst1)$ cat sqllib/db2nodes.cfg
0 blues 0

db2inst1@jazz(/db2/db2inst1)$ db2gcf -u -p 0 -i db2inst1

Instance : db2inst1
DB2 Start : Success
Partition 0 : Success

db2inst1@jazz(/db2/db2inst1)$ cat sqllib/db2nodes.cfg
0 jazz 0
```

Note: Considering the PowerHA environment, the **db2gcf -u -p 0 -i <instance name>** command line must be included in the application start script for the DB2 resource group.

Option 4: Use the db2start command to refresh the db2nodes.cfg configuration

Another option to guarantee that the DB2 services are starting properly on all cluster nodes is by forcing a `db2nodes.cfg` file refresh every time these services are initiated.

A requirement for this option is to establish a remote shell (RSH or SSH) connection that belongs to all cluster nodes.

First, you must insert in the DB2 registers the remote shell command to be used. In Example 6-15, SSH is chosen.

Example 6-15 DB2 register parameters for the remote shell command

```
db2set DB2RSHCMD=/usr/bin/ssh

db2inst1@jazz(/db2/db2inst1/sqllib)$ db2set | grep ssh
DB2RSHCMD=/usr/bin/ssh
```

After SSH is chosen, an SSH connection between all cluster nodes working without password and the db2inst1 user are required. This is because some security policies in certain environments deny the use of the SSH connection with no passwords.

When all requirements are met, the only changes in the PowerHA application startup scripts include extra parameters for the **db2start** command, as shown in Example 6-16.

Example 6-16 Starting DB2 services by using hostname restart

```
db2inst1@blues(/db2/db2inst1)$ hostname
blues

db2inst1@blues(/db2/db2inst1)$ cat sqllib/db2nodes.cfg
0 jazz 0

db2inst1@blues(/db2/db2inst1)$ db2start dbpartitionnum 0 restart hostname blues
12/12/2013 12:29:04     0   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.
db2inst1@blues(/db2/db2inst1)$ ps -fe | grep db2
db2inst1  8454176 16646242   0 12:29:04      -  0:04 db2acd
db2inst1 10354842 16646242   0 12:29:02      -  0:00 db2sysc 0
db2inst1 12976302 10354842   0 12:29:02      -  0:00 db2ckpwd 0
db2inst1 13172754 16646242   0 12:29:03      -  0:00 db2vend
db2inst1 13959416 10354842   0 12:29:02      -  0:00 db2ckpwd 0
db2inst1 16646242        1   0 12:29:02      -  0:00 db2wdog 0 [db2inst1]
db2inst1 19791982 10354842   0 12:29:02      -  0:00 db2ckpwd 0
```

6.4.2 Configure DB2 to work on all cluster nodes

After DB2 is installed and is running on the first cluster node, it is time to start configuring DB2 to work on the second node.

Import DB2 volume group and file systems

First, all DB2 services must be stopped on the first cluster node, as shown in Example 6-17 on page 122.

Example 6-17 Stopping DB2 services on first cluster node

```
db2inst1@blues(/db2/db2inst1)$ db2stop force
12/12/2013 07:23:19     0   0   SQL1064N  DB2STOP processing was successful.
SQL1064N  DB2STOP processing was successful.

db2inst1@blues(/db2/db2inst1)$ db2 terminate
DB20000I  The TERMINATE command completed successfully.
db2inst1@blues(/db2/db2inst1)$

db2inst1@blues(/db2/db2inst1)$ ps -fe | grep db2
db2inst1 13762734 11206704   0 07:24:34  pts/0  0:00 ps -fe
db2inst1@blues(/db2/db2inst1)$
```

With all services stopped in the first node, **umount** all DB2 file systems and **varyoff** the DB2 volume group, as shown in Example 6-18.

Example 6-18 Umounting DB2 file systems and volume group

```
root@blues(/)# umount /db2/dasusr1
root@blues(/)# umount /db2/db2fenc1
root@blues(/)# umount /db2/db2inst1
root@blues(/)#

root@blues(/)# lsvg -l db2vg
db2vg:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
db2inst1lv          jfs2       320     320     1    closed/syncd  /db2/db2inst1
db2fenc1lv          jfs2       320     320     2    closed/syncd  /db2/db2fenc1
dasusr1lv           jfs2       320     320     1    closed/syncd  /db2/dasusr1
loglv00             jfs2log    1       1       1    closed/syncd  N/A

root@blues(/)# varyoffvg db2vg
root@blues(/)#
```

Then, all Logical Volume Manager (LVM) information must be imported on the second cluster node, as shown in Example 6-19.

Example 6-19 Importing DB2 LVM information on the second cluster node

```
root@jazz(/)# importvg -V 65 -y db2vg hdisk2
db2vg
0516-783 importvg: This imported volume group is concurrent capable.
        Therefore, the volume group must be varied on manually.
root@jazz(/)# chvg -an db2vg

root@jazz(/)# varyonvg db2vg

root@jazz(/)# lsvg -l db2vg
db2vg:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
db2inst1lv          jfs2       320     320     1    closed/syncd  /db2/db2inst1
db2fenc1lv          jfs2       320     320     2    closed/syncd  /db2/db2fenc1
dasusr1lv           jfs2       320     320     1    closed/syncd  /db2/dasusr1
loglv00             jfs2log    1       1       1    closed/syncd  N/A
root@jazz(/)#
```

Start DB2 services manually on the second cluster node

With the volume group and the file systems properly mounted, it is time to check whether the DB2 services run properly on the second cluster node when using shared disks. By using a couple of DB2 commands (**`cat sqllib/db2nodes.cfg`** and **`db2start`**), you can validate the DB2 services by running queries after the start, as shown in Example 6-20.

Example 6-20 Starting DB2 services on second cluster node

```
db2inst1@jazz(/db2/db2inst1)$ cat sqllib/db2nodes.cfg
0 jazz 0

db2inst1@jazz(/db2/db2inst1)$ db2start
12/12/2013 13:02:00     0   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.

db2inst1@jazz(/db2/db2inst1)$ db2 connect to sample

   Database Connection Information

 Database server        = DB2/AIX64 10.5.2
 SQL authorization ID   = DB2INST1
 Local database alias   = SAMPLE

db2inst1@jazz(/db2/db2inst1)$ db2 select NAME from product

NAME
--------------------------
Snow Shovel, Basic 22 inch
Snow Shovel, Deluxe 24 inch
Snow Shovel, Super Deluxe 26 inch
Ice Scraper, Windshield 4 inch

  4 record(s) selected.
```

6.5 Create a PowerHA DB2 cluster

With all prerequisites applied and all cluster components tested manually, the next step is to configure the PowerHA infrastructure.

The packages installed on both cluster nodes, *blues* and *jazz*, are shown in Example 6-21.

Example 6-21 PowerHA packages installed on cluster nodes

```
root@blues(/)# lslpp -l | grep cluster
  bos.cluster.rte           7.1.3.1  COMMITTED  Cluster Aware AIX
  bos.cluster.solid         7.1.1.15  COMMITTED  POWER HA Business Resiliency
  cluster.adt.es.client.include
  cluster.adt.es.client.samples.clinfo
  cluster.adt.es.client.samples.clstat
  cluster.adt.es.client.samples.libcl
  cluster.doc.en_US.es.pdf   7.1.3.0  COMMITTED  PowerHA SystemMirror PDF
  cluster.doc.en_US.glvm.pdf
  cluster.es.assist.common   7.1.3.0  COMMITTED  PowerHA SystemMirror Smart
  cluster.es.assist.db2      7.1.3.0  COMMITTED  PowerHA SystemMirror Smart
```

```
  cluster.es.client.clcomd   7.1.3.0  COMMITTED  Cluster Communication
  cluster.es.client.lib      7.1.3.0  COMMITTED  PowerHA SystemMirror Client
  cluster.es.client.rte      7.1.3.0  COMMITTED  PowerHA SystemMirror Client
  cluster.es.client.utils    7.1.3.0  COMMITTED  PowerHA SystemMirror Client
  cluster.es.cspoc.cmds      7.1.3.0  COMMITTED  CSPOC Commands
  cluster.es.cspoc.rte       7.1.3.0  COMMITTED  CSPOC Runtime Commands
  cluster.es.migcheck        7.1.3.0  COMMITTED  PowerHA SystemMirror Migration
  cluster.es.nfs.rte         7.1.3.0  COMMITTED  NFS Support
  cluster.es.server.diag     7.1.3.0  COMMITTED  Server Diags
  cluster.es.server.events   7.1.3.0  COMMITTED  Server Events
  cluster.es.server.rte      7.1.3.0  COMMITTED  Base Server Runtime
  cluster.es.server.testtool
  cluster.es.server.utils    7.1.3.0  COMMITTED  Server Utilities
  cluster.license            7.1.3.0  COMMITTED  PowerHA SystemMirror
  cluster.msg.en_US.assist   7.1.3.0  COMMITTED  PowerHA SystemMirror Smart
  cluster.msg.en_US.es.client
  cluster.msg.en_US.es.server
  mcr.rte                    7.1.3.1  COMMITTED  Metacluster Checkpoint and
  bos.cluster.rte            7.1.3.1  COMMITTED  Cluster Aware AIX
  bos.cluster.solid         7.1.1.15  COMMITTED  POWER HA Business Resiliency
  cluster.es.assist.db2      7.1.3.0  COMMITTED  PowerHA SystemMirror Smart
  cluster.es.client.clcomd   7.1.3.0  COMMITTED  Cluster Communication
  cluster.es.client.lib      7.1.3.0  COMMITTED  PowerHA SystemMirror Client
  cluster.es.client.rte      7.1.3.0  COMMITTED  PowerHA SystemMirror Client
  cluster.es.cspoc.rte       7.1.3.0  COMMITTED  CSPOC Runtime Commands
  cluster.es.migcheck        7.1.3.0  COMMITTED  PowerHA SystemMirror Migration
  cluster.es.nfs.rte         7.1.3.0  COMMITTED  NFS Support
  cluster.es.server.diag     7.1.3.0  COMMITTED  Server Diags
  cluster.es.server.events   7.1.3.0  COMMITTED  Server Events
  cluster.es.server.rte      7.1.3.0  COMMITTED  Base Server Runtime
  cluster.es.server.utils    7.1.3.0  COMMITTED  Server Utilities
  mcr.rte                    7.1.3.1  COMMITTED  Metacluster Checkpoint and
  cluster.man.en_US.es.data  7.1.3.0  COMMITTED  Man Pages - U.S. English
```

To create an initial cluster configuration, verify that there are no file systems mounted on any cluster node and that `db2vg` is set to `varyoff`.

6.5.1 Create the cluster topology

Run the **`smitty sysmirror`** command, and then select **`Cluster Nodes and Networks`** → **`Standard Cluster Deployment`** → **`Set up a Cluster, Nodes and Networks`**.

This opens the panel that is shown in Figure 6-11 on page 125.

```
 Set up a Cluster, Nodes and Networks
Move cursor to the item that you want, and press Enter.
Type or select values in entry fields.
Press Enter AFTER making all changes.

                                                        [Entry Fields]
* Cluster Namebout repository disk and cluster IP add   [db2cluster]
  New Nodes (via selected communication paths)          [jazz]          +
  Currently Configured Node(s)                          blues
```

Figure 6-11 Creating DB2 cluster topology

After the cluster is created, define the repository disk for Cluster Aware AIX (CAA) by running **smitty sysmirror** → **Define Repository Disk and Cluster IP Address** and choosing the disk to be used, as shown in Figure 6-12.

```
 [Entry Fields]
* Cluster Name                                        db2cluster
* Heartbeat Mechanism                                 Unicast +
* Repository Disk                                    [(00f623c591941681)] +
  Cluster Multicast Address                          []
      (Used only for multicast heartbeat)
```

Figure 6-12 Defining the repository disk for the DB2 cluster

Before proceeding, run the **/usr/es/sbin/cluster/utilities/cltopinfo** command to check that all topology configuration that you just performed is correct. The result will look similar to Example 6-22.

Example 6-22 Output from the cltopinfo cluster command

```
root@blues(/etc/cluster)# /usr/es/sbin/cluster/utilities/cltopinfo
Cluster Name:    db2cluster
Cluster Type:    Standard
Heartbeat Type:  Unicast
Repository Disk: hdisk1 (00f623c591941681)

There are 2 node(s) and 2 network(s) defined

NODE blues:
        Network net_ether_01
                bluespriv       172.10.10.203
        Network net_ether_010
                blues   129.40.119.203

NODE jazz:
        Network net_ether_01
                jazzpriv        172.10.10.225
        Network net_ether_010
                jazz    129.40.119.225
```

Note: With the topology configuration finished, run the cluster verification with **`smitty sysmirror`** → **`Cluster Nodes and Networks`** → **`Verify and Synchronize Cluster Configuration`**. This procedure automatically replicates the cluster settings to all cluster nodes.

6.5.2 Create a DB2 resource group

After the cluster topology is created and replicated to all cluster nodes, the DB2 resource group can be created. Basically, this resource is composed of the *db2vg* volume group, including all of its file systems, DB2 service IP addresses, and application start and stop scripts that manage the DB2 services within clusters.

To create the service IP address: **`smitty sysmirror`** → **`Cluster Applications and Resources`** → **`Resources`** → **`Configure Service IP Labels/Addresses`** → **`Add a Service IP Label/Address`**. Then, define all service IP addresses related to all networks that are available within the cluster's topology, as shown in Example 6-23.

Example 6-23 Adding a service IP address to the cluster

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

                                                       [Entry Fields]
* IP Label/Address                                   cluster1 +
  Netmask(IPv4)/Prefix Length(IPv6)                 []
* Network Name                                       net_ether_01
```

Next, create the application, which is basically the DB2 services and their start and stop scripts. Enter or type **`smitty sysmirror`** → **`Cluster Applications and Resources`** → **`Resources`** → **`Configure User Applications (Scripts and Monitors)`** → **`Application Controller Scripts`** → **`Add Application Controller Scripts`**, as shown in Example 6-24.

Example 6-24 Creating application

```
[Entry Fields]
* Application Controller Name          [db2services] * Start Script
[/usr/es/sbin/cluster/scripts/db2start.ksh] * Stop Script
[/usr/es/sbin/cluster/scripts/db2stop.ksh]
 Application Monitor Name(s)
+
  Application startup mode                          [foreground]
```

After creating all of the resources, type **`smitty sysmirror`** → **`Cluster Applications and Resources`** → **`Resource Groups`** → **`Add a Resource Group`** to create the DB2 resource group shown in Example 6-25 on page 127.

Example 6-25 Creating DB2 resource group

```
[Entry Fields]
* Resource Group Name                                [db2rg]
* Participating Nodes (Default Node Priority)        [blues jazz] +

Startup Policy                                      Online On Home Node Only +
Fallover Policy                                    Fallover To Next Priority Node
In The List +
Fallback Policy                                     Never Fallback Never Fallback
```

Then, type **smitty sysmirror** → **Cluster Applications and Resources** → **Resource Groups** → **Change/Show Resources and Attributes for a Resource Group** to assign all DB2-related resources to this resource group, as shown in Example 6-26.

Example 6-26 Adding resources to a resource group

```
[TOP]                                                  [Entry Fields]
  Resource Group Name                                db2rg
  Participating Nodes (Default Node Priority)        blues jazz

  Startup Policy                                     Online On Home Node Only
  Fallover Policy                                    Fallover To Next Priority
Node In The List
  Fallback Policy                                    Never Fallback

  Service IP Labels/Addresses                       [cluster1 cluster01priv] +
  Application Controllers                           [db2services] +

  Volume Groups                                     [db2vg]
```

With all configurations complete, do another cluster verification and synchronization with **smitty sysmirror** → **Cluster Nodes and Networks** → **Verify and Synchronize Cluster Configuration**.

After this synchronization operation, the DB2 cluster is fully configured and ready to be tested and validated.

6.6 Test DB2 cluster functions

The first step to test and validate DB2 services inside a PowerHA cluster is to start all cluster services. To do this, run the **smitty clstart** command and choose all cluster nodes in the Start Cluster Services pane. This starts DB2 services in the order defined for the DB2 resource group.

After several seconds, the output of the **clRGinfo** cluster command shows that the cluster is active and the db2rg resource group is enabled on the blues cluster node, as shown in Example 6-27 on page 128.

Example 6-27 clRGinfo command output after services startup

```
root@blues(/var/hacmp/log)# clRGinfo
-----------------------------------------------------------------------------
Group Name     State                        Node
-----------------------------------------------------------------------------
db2rg          ONLINE                       blues
               OFFLINE                      jazz
```

6.6.1 Test database connectivity on the primary node

For a network node that is defined as a DB2 client, test the DB2 services by using the cluster IP address. The db2inst1 instance and a SAMPLE database were defined, as shown in Example 6-28.

Example 6-28 DB2 client definitions

```
db2 => list node directory

 Node Directory

 Number of entries in the directory = 1

Node 1 entry:

 Node name                      = CLUSTER1
 Comment                        =
 Directory entry type           = LOCAL
 Protocol                       = TCPIP
 Hostname                       = cluster1
 Service name                   = 50000

db2 =>

db2 => list database directory

 System Database Directory

 Number of entries in the directory = 1

Database 1 entry:

 Database alias                       = R_SAMPLE
 Database name                        = SAMPLE
 Node name                            = CLUSTER1
 Database release level               = 10.00
 Comment                              =
 Directory entry type                 = Remote
 Catalog database partition number    = -1
 Alternate server hostname            =
 Alternate server port number         =
```

With the DB2 client properly configured, the connection to the SAMPLE database as R_SAMPLE can be validated. Example 6-29 on page 129 shows that DB2 is working in the cluster and answering properly to the network requests.

Example 6-29 Testing the cluster database connection

```
db2 => connect to r_sample user db2inst1
Enter current password for db2inst1:

   Database Connection Information

 Database server        = DB2/AIX64 10.5.2
 SQL authorization ID   = DB2INST1
 Local database alias   = R_SAMPLE
db2 =>

db2 => select NAME from PRODUCT

NAME
--------------------------------
Snow Shovel, Basic 22 inch
Snow Shovel, Deluxe 24 inch
Snow Shovel, Super Deluxe 26 inch
Ice Scraper, Windshield 4 inch

  4 record(s) selected.

db2 =>

db2 => list applications

Auth Id  Application    Appl.      Application Id DB       # of
                        Name      Handle Name             Agents
-------------------------------------------------------------
DB2INST1 db2bp         7         129.40.119.203.61834.131212221231 SAMPLE   1

db2 =>
```

6.6.2 Test the failover to secondary node and validate DB2

To perform a manual failover on a PowerHA resource group, use the following command on the cluster, as shown in Example 6-30:

```
/usr/es/sbin/cluster/utilities/clRGmove -s 'false'  -m -i -g '<resource group
name>' - n '<node>'
```

Example 6-30 DB2 services manual failover

```
root@blues(/)# /usr/es/sbin/cluster/utilities/clRGmove -s 'false'  -m -i -g
'db2rg' -n 'jazz'
Attempting to move resource group db2rg to node jazz.

Waiting for the cluster to process the resource group movement request....

Waiting for the cluster to stabilize....................
Resource group movement successful.
Resource group db2rg is online on node jazz.

Cluster Name: db2cluster
```

```
Resource Group Name: db2rg
Node                         State
---------------------------- ---------------
blues                        OFFLINE
jazz                         ONLINE
root@blues(/)#
```

After the cluster stabilizes, it is time to verify that the database connection that points to the cluster IP address is working, as shown in Example 6-31.

Example 6-31 Testing the database connection on the secondary cluster node

```
db2 => connect to r_sample user db2inst1
Enter current password for db2inst1:

   Database Connection Information

 Database server        = DB2/AIX64 10.5.2
 SQL authorization ID   = DB2INST1
 Local database alias   = R_SAMPLE

db2 => select name from product

NAME
-------------------------------
Snow Shovel, Basic 22 inch
Snow Shovel, Deluxe 24 inch
Snow Shovel, Super Deluxe 26 inch
Ice Scraper, Windshield 4 inch

  4 record(s) selected.
```

By checking the tests results (Example 6-31), you can determine that DB2 is working on both cluster nodes and PowerHA is working with the DB2 services.

> **Note:** For more information about DB2 v10.5 administration, see "IBM DB2 10.1 for Linux, UNIX, and Windows documentation" in the IBM Knowledge Center:
>
> http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/index.jsp

7

Smart Assist for SAP 7.1.3

This chapter is an implementation guide that is based on the design for SAP NetWeaver non-database components: SAP Central Services, enqueue replication server, application server instances, and SAP global variables. It covers installation by using the IBM PowerHA SystemMirror 7.1.3 installation automation tool: *Smart Assist for SAP*. The installation was tested with PowerHA 7.1.3, SAP NetWeaver 7.30, and IBM DB2 10.1.

This chapter also documents customization options and deployment alternatives. It includes the following topics:

- Introduction to SAP NetWeaver high availability (HA) considerations
- Introduction to Smart Assist for SAP
- Installation of SAP NetWeaver with PowerHA Smart Assist for SAP 7.1.3
- Install SAP NetWeaver as highly available (optional)
- Smart Assist for SAP automation
- OS script connector
- Additional preferred practices
- Migration
- Administration
- Documentation and related information

7.1 Introduction to SAP NetWeaver high availability (HA) considerations

SAP NetWeaver is the technology platform of SAP business applications, such as enterprise resource planning (ERP), supply change management (SCM), cross-component products, and many others. It contains some single points of failures and provides hot standby capability. A primary focus is on the SAP Central Services (CS) and the enqueue replication server (ERS) as a rotating hot standby pair allowing for continuous business operations during and after failovers.

In 2013, SAP enhanced this functionality with the SAP HA API. The API links SAP and cluster products. The major benefits are planned downtime reduction and operational improvements. For more information, see Achieving High Availability for SAP Solutions:

`http://scn.sap.com/docs/DOC-7848`

7.1.1 SAP NetWeaver design and requirements for clusters

The information on the SAP NetWeaver 7.4 web page highlights the design as described by the SAP installation guide as relevant to the PowerHA Smart Assist for SAP as of 2013:

`http://help.sap.com/nw_platform`

This documentation complements but does not replace the official SAP guides.

Deployment options

Smart Assist for SAP supports the SAP Business Suite 7 for several variations, as described in the following sections:

- The infrastructure design
- The software and middleware stack
- The front end

The infrastructure design

The following IBM white paper describes HA deployments for SCM and SAP liveCache (see the infrastructure chapter for best practices for the hardware and network layers):

Invincible Supply Chain - SAP APO Hot Standby liveCache on IBM Power Systems

`http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100677`

A primary focus is to make failures in the infrastructure transparent to the application. Virtualization best practices minimize the impact of outages or planned maintenance to business operations at the hardware level.

Chapter 6, “Implementing DB2 with PowerHA” on page 103, describes additional best practices while setting up PowerHA in general.

Although not in the scope of this chapter, the following disaster recovery (DR) technologies should be considered (this is a general statement, because DR is not supported with Smart Assists):

- Limited distance (synchronous replication):
  - IBM HyperSwap (see Chapter 8, “PowerHA HyperSwap updates” on page 215)
  - IBM SAN Volume Controller (SVC) stretched cluster
  - Synchronous mirroring

  - Virtual I/O Server (VIOS) capabilities
  - DB features such as DB2 HADR (High Availability and Disaster Recovery)
- Unlimited distance (asynchronous replication)
  - DB features such as DB2 HADR 10.1+
  - Libelle
  - rsync (*not* recommended)

The software and middleware stack

The high availability entities of an SAP application, such as CRM, ERP, and other NetWeaver based systems, are inside the SAP technological platform, not the application itself. Therefore, the following sections focus on the SAP Central Services (CS), enqueue replication server (ERS), and app server instances of an SAP NetWeaver environment.

> **Important:** It is absolutely essential that each SAP instance is configured with its own virtual IP. This is regardless of whether it is controlled by the cluster or not, because this is a decision made during installation. As soon an instance is included into the SAP Landscape Virtualization Manager (LVM) or in a cluster, this becomes a prerequisite.

Figure 7-1 shows an overview of the software and middleware stack.

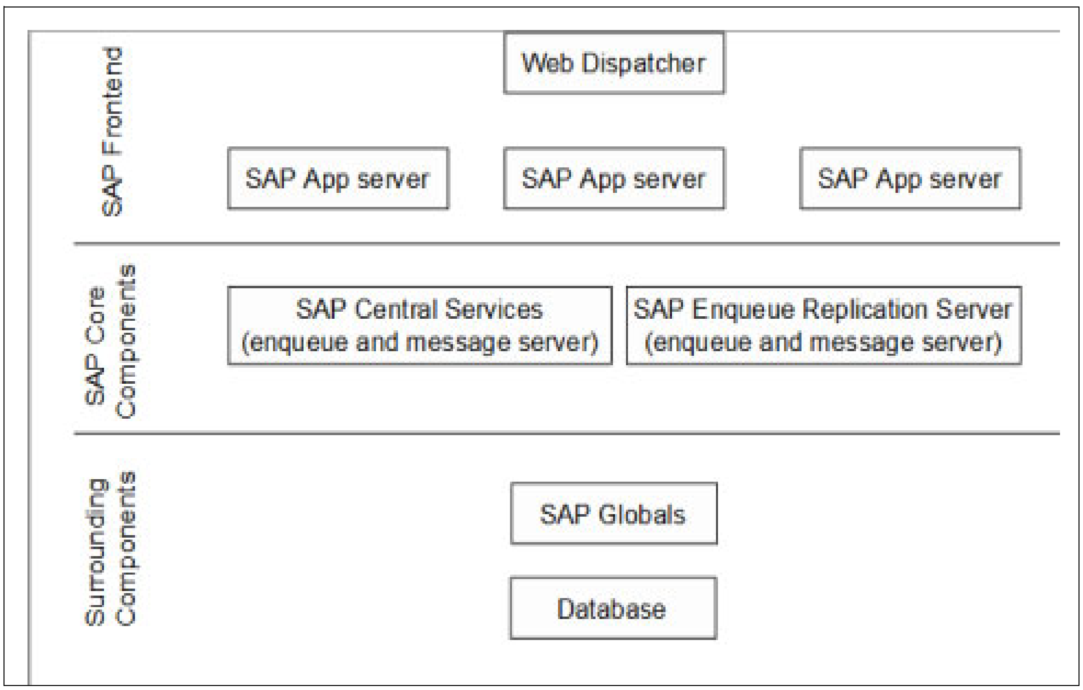


Figure 7-1 Software and middleware stack overview

The front end

The SAP application servers are installed in different nodes in a redundant configuration, and a load balancer is used as a front end. This makes the application continuously accessible to the user.

Load balancing is an SAP extension that is not included in Smart Assist for SAP functions. However, PowerHA can be used to make a SAP Web Dispatcher highly available. For help in creating an HA design for the Web Dispatcher, there are several information pages on the SAP Community Network and in sections in the SAP installation and planning guides.

The following options are valid for the SAP application server in a Smart Assist for SAP deployment:

- SAP application servers can be controlled by PowerHA as a local instance for startup orchestration and restart.
- SAP application servers can be controlled by PowerHA as a moving instance between cluster nodes. This is typically done for administrative purposes, because the restart of an SAP application server can take too long for business continuity purposes.
- SAP application servers can be installed outside or within the cluster nodes and not be controlled by PowerHA (SAP default).

Typical deployments have a mixture of the described options. There is no requirement to include them in the cluster. The only essential consideration is that there will be always enough instances available to handle the workload.

> **Note:** Controlling the SAP application server instances from inside PowerHA does not remove the requirement of setting up application servers with identical capabilities on different nodes, as documented by SAP.
>
> For all nodes, it is essential that there is no possibility of placing multiple instances that have the same instance numbers on the same node.

It is important to understand that for a high availability (HA) installation, there is nothing like a traditional central instance anymore, because the entities enqueue and message server are separated and put into a new instance called the Central Service instance (CS).

In Figure 7-2, we assume that Application Server A is responsible for spool requests. This would require a second application server on a different node that is also responsible for spool requests and has sufficient resources to handle the entire workload in case Application Server A goes down. If Application Server B is responsible for batch processing, there must be an Application Server B that can handle the workload in addition to its normal load on a second node.

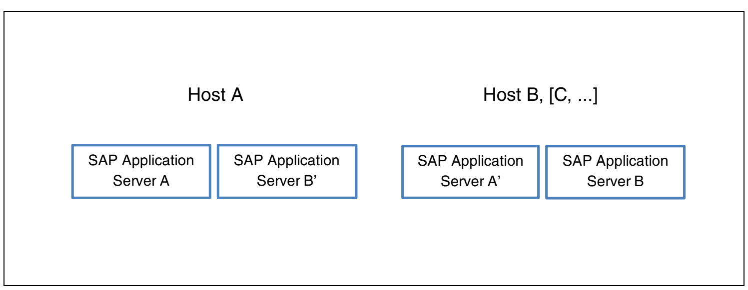


Figure 7-2 Redundant application server instances

When to cluster central and replication server instances

This section describes when to cluster the SAP application server instances for these purposes:

- Administration

  Having an SAP application server that is capable of moving between nodes will keep one instance always up and running. For production servers, this capability is of lower value, because the startup process (especially for the Java application server) can be very slow. But for development or other purposes, this might be quite useful.

- Automation

  For SAP application servers within the cluster LPARs, it might be helpful to include them in the PowerHA start process as opposed to starting them automatically on LPAR reboot. The PowerHA start process ensures that all prerequisites are fulfilled. That is not the case at boot time. PowerHA provides two options: Either configure the application servers for a single node or configure PowerHA to relocate them.

The NetWeaver HA core components: Central Services and ERS

The SAP NetWeaver installation configuration, using a stand-alone enqueue with an enqueue replication, allows failovers to be apparent to the business logic. The hot standby peers are called *SAP Central Services* (consisting of the enqueue and message server) and enqueue replication server.

> **Note:** SAP has a dedicated installation option in the `SWPM/sapinst` for HA setups. There are special installation guides that explain the required steps.

For the stand-alone enqueue of a stack, an SAP instance is explicitly created in `/usr/sap/<SID>/ASCSxx`.

The corresponding replicated enqueue is in `/usr/sap/<SID>/ERSyy` (green in Figure 7-3).

The same exists for the Java stack in these directories:

- `/usr/sap/<SID>/SCSzz`
- `/usr/sap/<SID>/ERSww` (red in Figure 7-3)

Again, it is not of technical relevance whether the Advanced Business Application Programming (ABAP) SAP Central Services (ASCS) and SAP Central Services (SCS) are initially started on different nodes or on the same node. It is more of a design and risk reduction consideration.

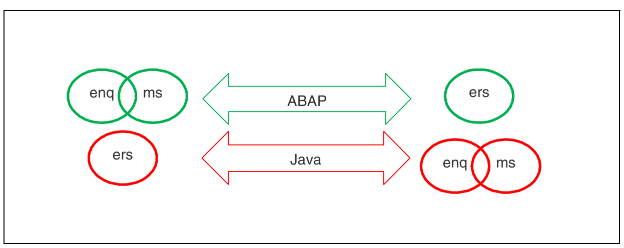


Figure 7-3 CS and ERS instances

The components of the cluster are the enqueue and message server, referred to hereafter as *ENSA* or *CS* if the message server is included. The corresponding Enqueue Replication Server is referred to as *ERS*. Smart Assist for SAP creates two resource groups (*RGs*) for each installed SAP stack (ABAP, Java), as shown in Figure 7-4. The PowerHA design always ensures that as PowerHA activates resources, the ERS processes are located on a cluster node separate from its CS instance. In the case where only one node is left, only the CS processes will be started (no ERS processes are started).

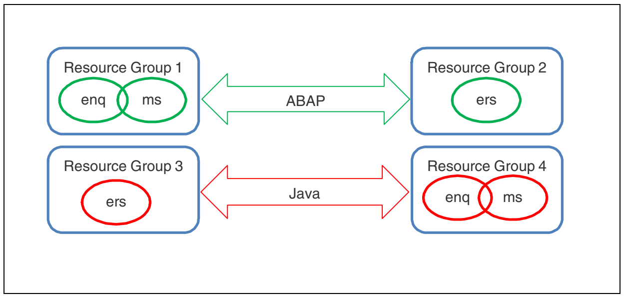


Figure 7-4 Resource groups for CS and ERS instances

The ERS can be enabled by three different methods that are supported by SAP:

SAP polling
: SAP polling is used if the cluster cannot control the ERS and locate this instance appropriately. This mechanism depends on a configuration in the SAP instance profiles and a script to monitor the local node. Each node has an ERS instance running (either active or inactive).

Cluster-controlled
: The cluster-controlled approach is what PowerHA supports. The enablement is to be performed by setting certain SAP instance profiles (see “Change the instance profiles of the AS, ASCS, SCS, and ERS” on page 191 for details). With this approach, no matter how many nodes are included, there is only one ERS. In the event of a failover, the CS instance is automatically relocated to the active ERS node.

Hardware solution
: This is a unique feature of IBM System z, where the coupling facility is used.

Surrounding components: Databases

The databases provide different options for an HA environment. It is strongly recommended to use a hot or at least warm standby.

A cold standby brings the database into an inactive state for rollbacks until operations can be continued. This can cause an outage of business operations for minutes or even hours.

A hot standby database can roll back the currently failed job. It can be connected by the application for other transactions in typically less than three minutes and have full performance capability.

SAP provides a broad selection of database solutions as well as Multiple Components in One Database (MCOD), three- or two-tier deployments.

To address this, Smart Assist is built in a modular fashion. Just add another application to Smart Assist for SAP by selecting the corresponding Smart Assist solution. The documentation can be found in 7.10, “Documentation and related information” on page 214. You can request guidance through the ISICC information service (isicc@de.ibm.com) or through your IBM Business Partner.

Surrounding components: SAP global and SAP transport directories

The SAP global directory is shown in Figure 7-5 as `<sapmnt>`. This directory splits into the SAP systems that use `<SAPSID>`. This `/<sapmnt>/<SAPSID>/` subdirectory must be available to all nodes on which one or more instances of this system are running or will run after a failover or SAP Logical Volume Manager (LVM) operation (such as when using the SAP relocation tool). This typically includes nodes that are not part of the HA cluster.

The SAP transport directory is to transport updates across the SAP landscape. Therefore, it is shared by multiple SAP systems as compared to `/<sapmnt>/<SAPSID>/`, which is shared among the instances of a single system.

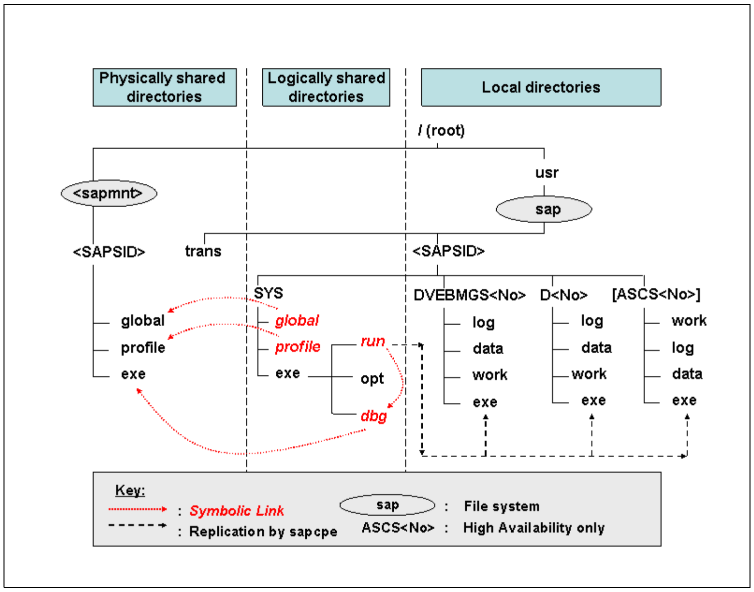


Figure 7-5 File system layout of an ABAP stack (ERP or SCM) without the ERS directory

In Figure 7-6, we have two SAP systems, called *PRD* and *QAS*. This figure does not describe best practices for how to locate PRD and QAS instances and how to make the shares available. The figure is intended to show mount requirements in the context of the SAP landscape.

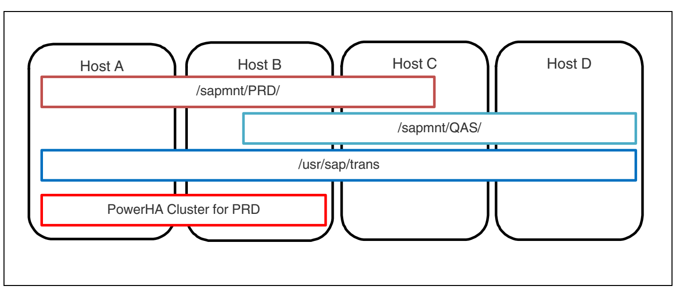


Figure 7-6 SAP global and SAP transport

The SCS and ERS instances of the PRD system are configured to Host A and Host B. This system is extremely important, so it is highly available by PowerHA. To handle the expected workload, the PRD application server instances are divided into three hosts: Host A, Host B, and Host C. This division requires the `/sapmnt/PRD` shared file system to be mounted to all three nodes.

Another system, QAS, is not clustered but is also running on the three nodes. QAS is a separate system, so it has its own SAP global file system, `/sapmnt/QAS`, which must be mounted on all nodes but not on Host A.

Due to the SAP transport landscape requirement of having to transport new developments and upgrades from a system, the `/usr/sap/trans` file system must be mounted in all nodes.

> **Note:** For special cases, SAP also provides the option of operating in a configuration where `/usr/sap/trans` is not a shared file system. This must be handled according to the regular SAP documentation. In that case, it is not relevant to clustering.

It is a prerequisite for starting an SAP instance that the SAP global directory be highly available. Therefore, take special care of the redundancy of the components, including storage redundancy, virtualization, and capabilities of the chosen technology.

There are different technologies that provide HA capabilities for a shared file system on IBM AIX, which are valid for a PowerHA based cluster with SAP NetWeaver:

- PowerHA crossmounts for each SAP system. This is proven PowerHA technology at no additional license cost. There are two different deployment options:
  - Use Smart Assist for NFS to set up an NFS crossmount. This provides the option to set up an NFSv4 or NFSv3.
  - Create an NFSv3 crossmount manually.

  Check that all remaining hosts have the permission and automated setup to mount the shared file systems.
- A central, highly available NFS server that uses PowerHA (typical deployment option). Centralizing the NFS server brings benefits in maintenance, patching, and operations. However, an outage of this cluster has a larger effect compared to separated crossmounts for each SAP cluster.
- A storage filer, such as the V7000U (NFS) or SONAS (GPFS), can be used to provide the shared file systems to all nodes.
- GPFS as an AIX file system is a separately purchased item, but it provides robustness.

The storage and file system layout

Based on the best practices mentioned for the hardware infrastructure that are described in 7.1.1, "SAP NetWeaver design and requirements for clusters" on page 132, the file systems on the appropriately attached storage volumes must be created as preparation for the implementation.

PowerHA supports different valid architectures, depending on the instance type:

- Application server instances can be set up bound to a node or moving between nodes.
- Moving application server instances can have either a local (duplicated) file system on the nodes or a shared file system that moves along with the resource group.
- ERS and SCS instances must be able to move between nodes. They can be deployed based on either a local and duplicated file system or a shared file system.
- Host agents, diagnostic agent (DAA), the central `/usr/sap` and OS-related file systems are local to each node.
- SAP gobal and transport directories are on a shared file system when deployed, based on an NFS crossmount. A list of additional options is described in 7.1.1, "SAP NetWeaver design and requirements for clusters" on page 132.

- The database file system setup depends on the selected database and the type. Follow the instructions in 7.10, “Documentation and related information” on page 214 for the chosen database option.

The decision for which mount strategy to use is based on what can be maintained best by the existing skills onsite, the overall architecture, and where the SAP log files should be written.

When using a local disk approach, PowerHA does not monitor the availability of the file system.

The storage layout must separate LUNs that are used for a local file system that is specific to a node from those file systems that are moving between nodes. Furthermore, the file systems being moved must be separated for each instance to allow for independent moves. Therefore, each SAP instance should have its own LUNs and file systems.

> **Note:** Older dual stack installations often combined both SAP Central Services instances (ASCS and SCS) into one resource group on one disk with one IP. Also, ERS instances in older releases were often installed with the hostname of the LPAR or system as an SAP dependency. This setup can be migrated from a PowerHA base product. But it cannot be moved to the new Smart Assist for SAP capabilities without first meeting the required prerequisites.

Supported SAP NetWeaver versions

This new release uses the SAP HA API as described on the SAP HA certification web page:

http://scn.sap.com/docs/DOC-26718

However, to remain compatible with an earlier version, PowerHA supports all current Business Suite 7 releases for SAP NetWeaver without the SAP HA API.

7.1.2 SAP HA interface and the SAP HA certification criteria

To fully use all new features, a minimum SAP NetWeaver kernel of 7.20 with NetWeaver 7.30 and patch level 423 is required. Business Suite 7 releases for SAP

In earlier days, if an SAP system was stopped from the SAP Management Console (MMC) or other tools, the cluster reacted with a failover, which interrupted upgrades and other SAP maintenance tasks.

SAP provides the option to integrate cluster products with SAP for start, stop, and move activities. Integration of start, stop, and move operations of SAP with the cluster allows SAP operators and SAP tools to automatically perform these activities without interruption and allows special processes to link with the cluster administrator, from a technical point of view.

SAP HA API version 1.0 is implemented with the Smart Assist for SAP 7.1.3 release. The enablement is optional, and it can be enabled or disabled at the SAP instance level.

> **Note:** The SAP HA certification certifies that mandatory components (SCS and ERS) are separated and the Software Upgrade Manager (SUM) can safely operate on clustered instances. It does not cover cluster robustness and test considerations.

7.2 Introduction to Smart Assist for SAP

Starting with version 7.1, PowerHA includes Smart Assist agents with no additional license fees.

You still have the freedom to create a cluster with homemade scripts or to customize solutions. However, using Smart Assist brings four significant advantages, at no additional cost, compared to custom solutions:

- Speed of deployment (relevant to TCO):

  The setup effort is reduced to a few hours, compared to weeks for a custom solution (especially for larger applications, such as SAP).

- Repeatable and proven setup:

  Smart Assist is pretested, and you benefit from a lifecycle and migration support when SAP provides new features. This can improve cluster uptime.

- Three-phase approaches for deployment:

  a. Discovers running application

  b. Verifies setup before clustering

  c. Adds cluster configuration and scripts

- Full IBM product support, including scripts and cluster configuration, in addition to the base product support.

7.2.1 SAP HA interface enablement

One key enhancement of PowerHA 7.1.3 is the compliance with the SAP HA interface. This is an optional feature that is usable when the SAP minimum release requirements are met. The SAP HA API version supported is 1.0 (when the First Edition of this book was published in 2014).

The key element in Figure 7-7 on page 141 is the *sapstartsrv* process of each instance, which requires PowerHA to start, stop, and monitor scripts to plug into this infrastructure. These scripts can still serve all of the 7.20 kernel-based NetWeaver releases, starting with NetWeaver 7.00.

The benefit is that the cluster can distinguish between an intentional stop and a failure of an SAP NetWeaver instance. All cluster scripts and the SAP tools, such as the SUM, MMC, SAP LVM, and so on, can take advantage of this infrastructure.

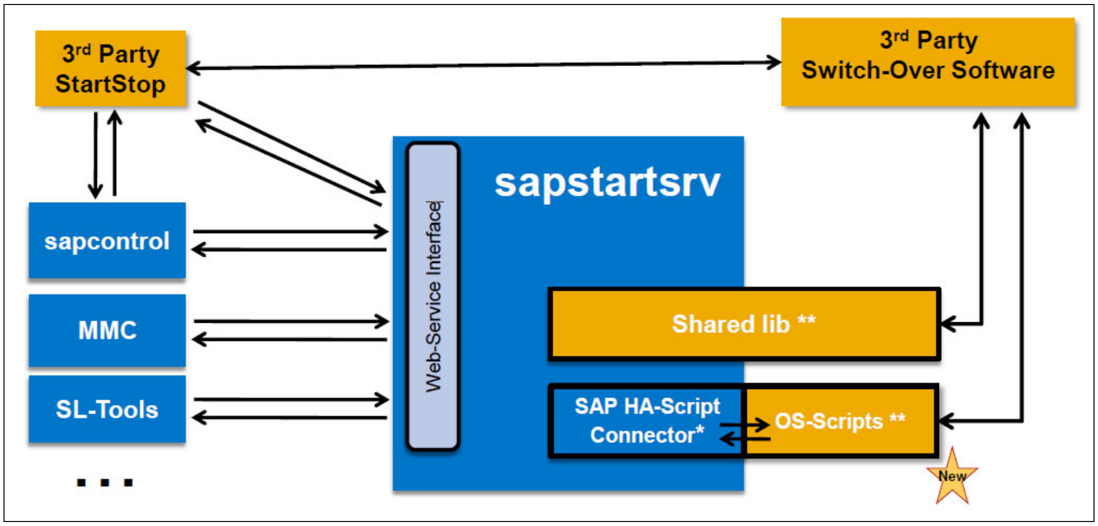


Figure 7-7 SAP's SAP HA API architecture

Figure 7-8 on page 142 shows how PowerHA plugs into the framework.

Third party switch-over software

PowerHA as the SAP third party switch-over software includes a new generation of start, stop, and monitoring capabilities for the new SAP HA API and optimizes planned downtime. It also still serves previous SAP NetWeaver deployments without the SAP HA API.

By SAP design, this software covers only the SCS and ERS. IBM has added functionality to handle application server instances. Databases are not enabled in SAP HA API version 1.0.

OS scripts

The OS script connector, which is known to SAP through the SAP instance profiles, provides the IBM counterpart piece to the SAP HA script connector. Through this connectivity, PowerHA can distinguish between an intentional stop or start of the instance and a failure. The OS scripts can be customized for each instance to enable or disable the functionality without changing the instance profile.

s

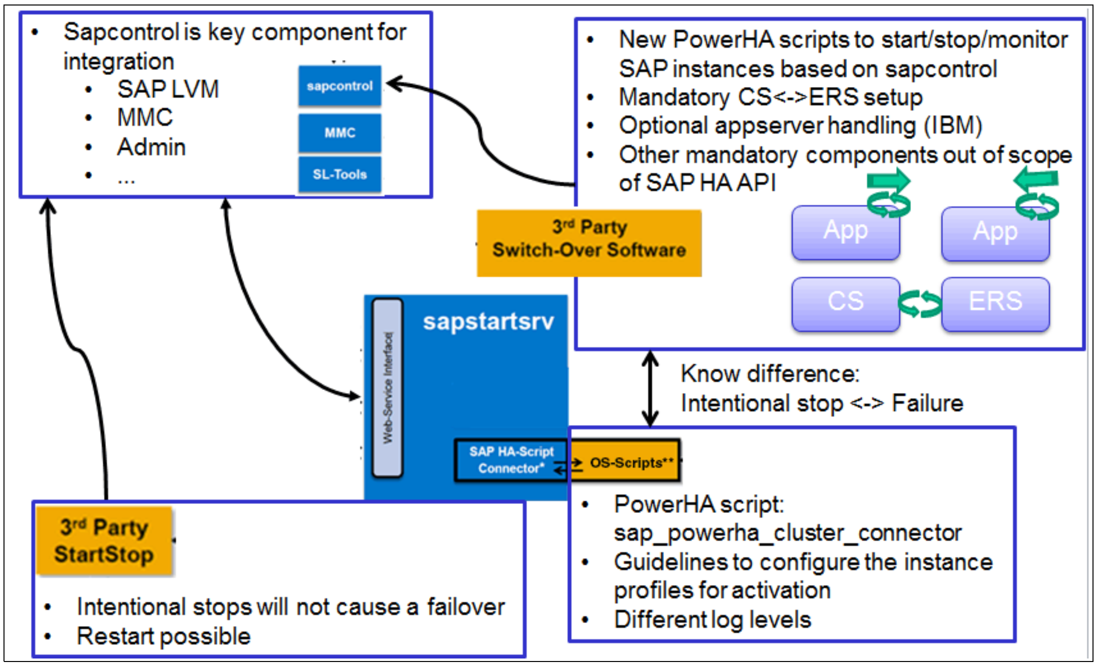


Figure 7-8 SAP HA API exploitation by Smart Assist for SAP

7.2.2 Infrastructure design: PowerHA

Smart Assist is built on a two- or three-node cluster. The default deployment is a two-node cluster, although Smart Assist supports the same number of nodes in the cluster as PowerHA. For more information, see 7.3.4, “PowerHA basic two-node deployment” on page 148.

7.2.3 Infrastructure design: Smart Assist for SAP

Smart Assists are built in a modular manner. When deploying an SAP ERP with an Oracle database, Smart Assist for Oracle, Smart Assist for SAP, and Smart Assist for NFS can be run one after the other. Additional homemade or customized scripts can be added when avoiding any dependencies to Smart Assist deployments.

In Smart Assist for SAP, there are three submenus:

- ASCS/SCS
- ERS
- Application server instances

Each of them creates a dedicated PowerHA resource group with a share-nothing approach. Each can move independently, as shown in Figure 7-2 on page 134 and Figure 7-4 on page 136.

7.3 Installation of SAP NetWeaver with PowerHA Smart Assist for SAP 7.1.3

This section provides the required preparation and deployment details to successfully cluster an SAP NetWeaver with an optional SAP HA API integration.

Additional considerations and information

Keep the following considerations in mind while using Smart Assist for SAP:

- Smart Assist naming conventions should not be changed, although this is possible and supported.
- Each SAP instance must have its own virtual IP.
- Plan on making your infrastructure highly available by using the required disk layout.

Additional resource groups for other applications running on the same cluster nodes can be created. However, any dependencies between the resource groups should be avoided. Dependencies can easily cause side effects that can be hard to test for, because they typically occur only in some cases.

7.3.1 Operating system and PowerHA software

The following tasks must be completed before using Smart Assist for SAP.

Plan

Review the PowerHA installation prerequisites that are described in 4.2, “PowerHA SystemMirror 7.1.3 requirements” on page 50.

The following are the required minimum releases of the IBM software:

- AIX operating system (OS) version 6.1 or later
- PowerHA version 7.1.3 or later

The following are PowerHA 7.1.3 prerequisites or recommended:

- Attach a dedicated shared disk for CAA of 1-460 GB to all cluster nodes.
- (Optional but recommended) Plan for FSWCOMM.
- Each cluster node’s hostname must be the same as the PowerHA node name. These names can not be resolvable to an IP address that matches one of the SAP instances.

The following AIX filesets are required:

- bos.rte.libc 7.1.3.0 # Base Level fileset
- rsct.basic.hacmp 3.1.5.0 # Base Level fileset
- rsct.basic.sp 3.1.5.0 # Base Level fileset
- bos.ahafs 7.1.3.0 # Base Level fileset
- bos.cluster.rte 7.1.3.0 # Base Level fileset
- bos.clvm.enh 7.1.3.0 # Base Level fileset
- devices.common.IBM.storfwork.rte 7.1.3.0 # Base Level fileset
- rsct.compat.basic.hacmp 3.1.5.0 # Base Level fileset
- rsct.compat.clients.hacmp 3.1.5.0 # Base Level fileset
- rsct.core.rmc 3.1.5.0 # Base Level fileset
- bos.64bit 7.1.3.0 # Base Level fileset
- bos.adt.include 7.1.3.0 # Base Level fileset

- bos.adt.prof 7.1.3.0 # Base Level fileset
- bos.adt.syscalls 7.1.3.0 # Base Level fileset
- bos.mp64 7.1.3.0 # Base Level fileset
- bos.rte.control 7.1.3.0 # Base Level fileset
- bos.rte.security 7.1.3.0 # Base Level fileset
- mcr.rte 7.1.3.0 # Base Level fileset

The following are the IP heartbeat considerations:

- Multicast versus point to point (see 7.3.4, “PowerHA basic two-node deployment” on page 148).
- Plan to make your infrastructure highly available as described in 7.1, “Introduction to SAP NetWeaver high availability (HA) considerations” on page 132.

Smart Assist for SAP supports a traditional two and three node cluster deployment. Although Smart Assist supports the same number of nodes in the cluster as PowerHA. Evaluate your business requirements compared to the increased complexity. A typical PowerHA setup for SAP NetWeaver consists of two nodes.

Ensure you have all nodes appropriately sized based on the expected workload. This includes disk sizes, CPU and memory. For assistance you can request support from the ISICC sizing team at isicc@de.ibm.com.

Install

After installing the operating system, preferably on a dedicated disk and in a dedicated volume group, the PowerHA software needs to be installed.

On each node, the following PowerHA software components must be installed as a minimum to create the base cluster and configure an NFS crossmount and SAP NetWeaver:

- cluster.adt.es
- cluster.doc.en_US.es.pdf
- cluster.doc.en_US.assist.sm
- cluster.es.assist.
- cluster.es.server
- cluster.license
- cluster.es.migcheck
- cluster.es.cspoc
- cluster.man.en_US.es.data
- cluster.es.nfs

Verify that you have downloaded the latest PowerHA Service Pack, and be sure to update the following files:

- `/etc/hosts`: Insert all node names and all service IPs that you plan to include in the cluster.
- `/etc/cluster/rhosts`: Insert all nodes by IP.

Verify

Perform the following verifications:

- Verify on each node that the same operating system and PowerHA level are installed.
- Verify on each node that the operating system and PowerHA versions are updated with the latest fixes or PTFs.
- Ensure that the size of /tmp, /home, /var and /usr is at least 3 GB (and monitor regularly for space).

7.3.2 Storage disk layout for SAP NetWeaver

Review the following options to plan, implement, and verify the disk attachment for the SAP PowerHA setup.

Plan

The application disks are to be separated from the operating system's disk. This results in having a set of disks for each of the following elements:

- The rootvg
- The SAP code under `/usr/sap`
- The independently moving SAP instances

Additional disks might be required for the SAP global and the transport directories and for the database.

The following sections describe the disk considerations and options, which are grouped by categories:

- Basic disks
- SAP SCS and ERS
- PowerHA crossmount for SAP directories and SAP application server instances

> **Note:** The database disk layout is described in the documentation of the database deployment in 7.10, "Documentation and related information" on page 214.

Basic disks

Table 7-1 shows the disks that are available for the basic installation of SAP, AIX, and PowerHA. It is recommended to separate the components on different disks, not just different file systems.

Table 7-1 Basic discs for an SAP NetWeaver, AIX, and PowerHA Installation

Disk	Instance	Mount point	Nodes
1	Operating system		Local to each node
2	SAP	/usr/sap	Local to each node
3	SAP-SID[a]	/usr/sap/SID	Local to each node
4	PowerHA CAA disk		Attached to both (shared)

a. Optional separation. Typically performed if more than one SAP system in the same host.

PowerHA crossmount for SAP directories and SAP application servers

As a starting value for the installation, the size can be set between 4 - 8 GB. This is not sufficient for production. The appropriate size depends on the type and expected workload.

PowerHA supports two options:

- Local disk
- Shared disk

A local disk stays with the node, and the file system structure must be copied to the second node (instructions are given later in the implementation flow under "SAP sapcpe copy tool" on page 207). The advantage is an easy disk attachment, because there is a 1:1 relationship between the LUN and the host. Disadvantages include a larger storage requirement and the inability of PowerHA to monitor the local disk in the same manner as a shared disk. Also consider the SAP logging implication:

- With a local disk approach (see Table 7-3), the SAP logs are written per node.
- With a shared disk approach (see Table 7-2), SAP continuously writes to the same log.

Either of the two approaches works for the cluster. A key consideration is whether the approach fits the available administrative skill set and your overall strategy. Also, consider the SAP Landscape Virtualization Manager (LVM) requirements.

Although each disk option is available for each installation and can be used independently of the others, it is highly recommended to use only one option (local or shared disk) for all of your implementations, for consistency.

Table 7-2 Disk layout for shared disk

Disk	Instance	Mount point	Nodes
1	ASCS	`/usr/sap/SID/ASCSxx`	Attached to both (shared)
2	SCS	`/usr/sap/SID/SCSxx`	Attached to both (shared)
3	ERS (for ASCS)	`/usr/sap/SID/ERSxx`	Attached to both (shared)
4	ERS (for SCS)	`/usr/sap/SID/ERSxx`	Attached to both (shared)
5-n	D*/J*	`/usr/sap/SID/[D*|J*]`	Attached to both (shared)

For the alternative, a local file system can be implemented where the instances can reside on a dedicated disk. However, this implementation will consist of additional subdirectories in `/usr/sap/<SID>` within the same file system.

Table 7-3 Disk layout for local disk

Disk	Instance	Mount point	Nodes
1	ASCS	`/usr/sap/SID/ASCSxx`	Local to each node
2	SCS	`/usr/sap/SID/SCSxx`	Local to each node
3	ERS (for ASCS)	`/usr/sap/SID/ERSxx`	Local to each node
4	ERS (for SCS)	`/usr/sap/SID/ERSxx`	Local to each node
5-n	D*/J*	`/usr/sap/SID/[D*|J*]`	Local to each node

PowerHA crossmount for the SAP global and transport directories (optional)

The mount point for the SAP globals (Table 7-4) depends on how the directories are available to all NFS clients.

Table 7-4 Disk layout for local disk

Disk	Instance	Mount point	Nodes
1	SAP Global	`/sapmnt/SID or /sapmnt`	Attached to both nodes, mounted on all nodes that this SID is running
2	SAP Transport	`/usr/sap/trans`	Attached to both nodes, mounted across the transport landscape

Install

The scenario in this example uses an IBM SAN Volume Controller stretched cluster base. It includes VDisk mirroring for data redundancy through multiple Fibre Channel attachments, using NPIV adapters in a VIOS implementation. You can use other disk architectures that provide similar capabilities, granularities, and high availability.

For setup details, see *IBM SAN Volume Controller Stretched Cluster with PowerVM and PowerHA*, SG24-8142:

http://www.redbooks.ibm.com/abstracts/sg248142.html?Open

Verify

When the disks are made available to the operating system, the attachment can be verified by using the **lspv** command and comparing the physical volume ID (pvid) between the nodes. A shared disk displays identical pvids, but a local disk displays different pvids.

Table 7-5 shows a prepared PowerHA Smart Assist for SAP cluster where the local disks are active and the shared disks are not.

Table 7-5 Example PowerHA with Smart Assist for SAP

<table>
<tr><th>Node A</th><th>Node B</th></tr>
<tr><td><pre>
#lspv
hdisk0   00f6ecb5780c7a60 rootvg active
hdisk2   00f6ecb5acf72d66 caavg_private   active
hdisk1   00f6ecb5221b7a85 vgaerscss
hdisk3   00f6ecb5221b79e9 vgsapcss   active
hdisk4   00f6ecb5221b7954 vgscscss
hdisk5   00f6ecb5221b78c6 vgascscss
hdisk6   00f6ecb5221b783a vgsapcss   active
hdisk7   00f6ecb5221b77ac vgsap   active
hdisk8   00f6ecb5221b771a vgerscss
hdisk9   00f6ecb5221b7677 vgsapmnt
hdisk10 00f6ecb522e5c449 vgtrans
hdisk11 00f6ecb529e455e7 vgsapcssap1   active
</pre></td><td><pre>
#lspv
hdisk0   00f6ecb5780c7a68 rootvg active
hdisk2   00f6ecb5acf72d66 caavg_private   active
hdisk1   00f6ecb5221b7fca vgsapcss active
hdisk3   00f6ecb5221b7f3d vgsap active
hdisk4   00f6ecb5221b7ebb vgsapcssap1   active
hdisk6   00f6ecb5221b7a85 vgaerscss
hdisk7   00f6ecb5221b7954 vgscscss
hdisk8   00f6ecb5221b78c6 vgascscss
hdisk9   00f6ecb522e5c449 vgtrans
hdisk10 00f6ecb5221b7677 vgsapmnt
hdisk11 00f6ecb5221b771a vgerscss
hdisk12 00f6ecb5410b0e66 rootvg   active
hdisk13 00f6ecb5714a97df vgsapcss   active
</pre></td></tr>
</table>

> **Note:** The hdisk numbers are not necessarily identical on each node for shared disks. Table 7-5 shows examples highlighted in blue for shared disk pvids.

The size of the disks can be verified by using the **getconf DISK_SIZE /dev/hdisk<x>** command.

7.3.3 Set global required OS and TCP/IP parameters

These tasks must be performed on both nodes. For more information, see the SAP Installation Guide:

http://help.sap.com/nw_platform

Relevant online SAP service (OSS) notes

Check the following SAP Notes to get the latest information:

- 1048686
- 973227
- 856848
- 1121904
- 1023047

Also browse the SAP Notes page to verify whether additional SAP notes were published after publication of this book:

https://service.sap.com/notes

Change root user limits

Some installation and configuration steps are run as `root` user. Set the soft and hard limits for CPU time, file size, data segment size, RSS size, and stack size to unlimited for the root user by using the following command:

```
chuser fsize='-1' data='-1' stack='-1' rss='-1' nofiles='32000' \
                       cpu_hard='-1' fsize_hard='-1' data_hard='-1' \
                       stack_hard='-1' rss_hard='-1'  \
                       root
```

Note: Check for updates according to the referenced SAP OSS notes.

7.3.4 PowerHA basic two-node deployment

This section describes one option to deploy a standard two-node PowerHA cluster.

Plan

There are two significant differences in the design of PowerHA 7.1 and later, compared to PowerHA 6.1, which should be considered in the planning stage:

- Multicast IP:

  PowerHA 7.1 and later uses multicasting for heartbeat and cluster communication. Therefore, the switches in the environment should be enabled for multicast traffic. If necessary, modify the switch settings. The **mping** test tool functions similarly to the point-to-point IP test tool, **ping**, and it can be used to test multicast connections. Use the **mping** tool first at the AIX level to make sure that the multicast packets are flowing between the nodes. The **mping** tool requires that you start **mping** on the receive node first, to look for a particular multicast address, and then send a packet from the other node, using **mping** for that particular multicast address. Any multicast communication issues must be resolved before starting the cluster.

  This implies the all networks defined to PowerHA need to be multicast-enabled.

  Also, starting with PowerHA 7.1.3, Unicast IP is again supported as with PowerHA 6.1.

- Shared repository disk:

  The heartbeat disk that was supported as an optional communication path in PowerHA 6.1 is no longer necessary or supported from 7.1. However, a shared disk is now mandatory for a centralized cluster software repository.

  This repository disk stores some of the configuration information centrally and provides the disk heartbeat function. Currently, only one disk is supported as a repository disk. Therefore, this disk should be highly available. In the proof of concept, this disk is mirrored at the hardware level by the SVC over multiple storage servers. For single storage solutions, such as the IBM DS8000® disk storage, this disk is protected at the storage level by RAID only. This single disk implementation is a current restriction of PowerHA.

Configuring PowerHA with Unicast

This section describes the steps for PowerHA configuration, using Unicast as primary heartbeat mechanism.

1. Set up IP addresses and hostnames in `/etc/hosts`.

   Add all cluster-used IP addresses and host names to `/etc/hosts` on each cluster node. Ensure that the host name matches the node name, as shown in Example 7-1.

   Example 7-1 /etc/hosts entries

   ```
   […]
   #HA1 nodes
   10.17.184.187   as0007lx as0007lx.wdf.sap.corp
   10.17.184.188   as0008lx as0008lx.wdf.sap.corp
   […]
   ```

2. Set up cluster nodes IP addresses in /etc/cluster/rhosts

   Add the IP addresses of all cluster nodes to /etc/cluster/rhosts, as shown in Example 7-2. Then, copy this file to all cluster nodes.

   Example 7-2 /etc/cluster/rhosts entries

   ```
   10.17.184.187
   10.17.184.188

   ~
   "/etc/cluster/rhosts" [Read only] 3 lines, 29 characters
   ```

Create a two-node cluster.

The basic setup of the PowerHA cluster software depends on these primary actions:

- Ensure that clcomd is active
- Define the cluster name and nodes
- Define the cluster repository and IP address
- Synchronize the cluster

1. Ensure that the *clcomd* daemon is running on both nodes:

   ```
   #lssrc -s clcomd
   Subsystem          Group              PID         Status
   clcomd             caa              6226140       active
   ```

   If the daemon has no *active* status, use the following command for activation:

   ```
   #startsrc -s clcomd
   ```

2. Define a name for the cluster, and select the second node of the cluster, as shown in Example 7-3:

   `smitty cm_setup_menu → Setup a Cluster, Nodes and Networks`

   Example 7-3 PowerHA initial cluster setup

```
Setup Cluster, Nodes and Networks (Typical)

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

                                                        [Entry Fields]
* Cluster Name                                       [SAP_DUAL_localdisk]
  New Nodes (via selected communication paths)       [as0004lx]          +
  Currently Configured Node(s)                        as0003lx



F1=Help             F2=Refresh           F3=Cancel           F4=List
F5=Reset            F6=Command           F7=Edit             F8=Image
F9=Shell            F10=Exit             Enter=Do
```

3. Select **Unicast** as the heartbeat mechanism and select the **repository disk**, as shown in Example 7-4. In this case, no multicast address is needed.

   `smitty cm_setup_menu → Define Repository Disk and Cluster IP Address`

Example 7-4 Define repository disk and cluster IP

```
Define Repository and Cluster IP Address

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

                                                        [Entry Fields]
* Cluster Name                                         SAP_DUAL_localdisk
* Heartbeat Mechanism                                  Unicast               +
* Repository Disk                                      []                    +
  Cluster Multicast Address                            []
      (Used only for Multicast Heart Beating)


      +--------------------------------------------------------------------------+
      |                            Repository Disk                               |
      |                                                                          |
      | Move cursor to desired item and press Enter.                             |
      |                                                                          |
      |   hdisk2  (00f6ecb511226888) on all cluster nodes                        |
      |   hdisk5  (00f6ecb5112266c9) on all cluster nodes                        |
      |                                                                          |
      | F1=Help                F2=Refresh             F3=Cancel                  |
F1=Help| F8=Image               F10=Exit               Enter=Do                   |
F5=Rese| /=Find                 n=Find Next                                       |
F9=Shel+--------------------------------------------------------------------------+
```

Nothing should be defined on this disk, no volume group or logical volumes. PowerHA finds suitable disks for selection, and then creates its own volume group. This disk is used for internal cluster information sharing and heartbeat information. In other words, it is fully reserved for the cluster.

4. Confirm that you want to synchronize the cluster nodes:

   `smitty cm_ver_and_sync`

5. Start cluster services on node one as shown in Example 7-5 (this is a *prerequisite* for the subsequent tasks):

   `smitty clstart`

Example 7-5 Start cluster services

```
Start Cluster Services

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

                                                        [Entry Fields]
   * Start now, on system restart or both         now                     +
     Start Cluster Services on these nodes        [as0003lx, as0004lx]    +
   * Manage Resource Groups                       Automatically           +
     BROADCAST message at startup?                false                   +
     Startup Cluster Information Daemon?          true                    +
     Ignore verification errors?                  false                   +
     Automatically correct errors found during    Interactively           +
     cluster start?


F1=Help              F2=Refresh             F3=Cancel            F4=List
F5=Reset             F6=Command             F7=Edit              F8=Image
F9=Shell             F10=Exit               Enter=Do
```

Configuring PowerHA with multicast

This implementation example is based on the multicast setup. Enable your network for the multicast communication. Instructions are provided in 7.10, “Documentation and related information” on page 214.

1. Set up IP addresses and hostnames in `/etc/hosts`.

   Add all cluster IP addresses and hostnames to /etc/hosts on each cluster node. Ensure hostname equals the node name.

2. Set up cluster nodes hostname IP addresses in `/etc/cluster/rhosts`.

   Add the IP addresses of all cluster nodes in `/etc/cluster/rhosts`. Then, copy this file to all cluster nodes.

Create a two-node cluster

The basic setup of the PowerHA cluster software depends on three primary steps. First, set up the cluster name and the involved cluster nodes. Second, define the repository disk and the cluster IP address which are based on a multicast IP address. Third, verify and synchronize the cluster configuration.

1. Ensure that the *clcomd* daemon is running on both nodes

```
#lssrc -s clcomd
Subsystem          Group              PID          Status
clcomd             caa                6226140      active
```

If the daemon has no status of active, tthe activation is done with the following command:

```
#startsrc  -s clcomd
```

2. **smitty cm_setup_menu → Setup Cluster, Nodes and Networks**

   Example 7-6 shows the initial PowerHA cluster setup.

Example 7-6 PowerHA initial cluster setup

```
Setup Cluster, Nodes and Networks (Typical)

                                                        [Entry Fields]
* Cluster Name                                          [SAP_DUAL_SharedDisk]
  New Nodes (via selected communication paths)          [as0008lx]
  Currently Configured Node(s)                          as0007lx
```

3. **smitty cm_setup_menu → Setup a Cluster, Nodes and Networks → Standard cluster Deployment → Define Repository Disk and Cluster IP Address**.

   Example 7-7 shows the menu to define the repository disk and the cluster IP.

Example 7-7 Define repository disk and cluster IP

```
Define Repository and Cluster IP Address

                                                         [Entry Fields]
* Cluster Name                                           SAP_DUAL_SharedDisk
* Heartbeat Mechanism                                    Multicast             +
* Repository Disk                                        [(00f6ecb5acf72d66)]  +
  Cluster Multicast Address                              [ ]
      (Used only for Multicast Heart Beating)


  +----------------------------------------------------------------------+
  |                             Repository Disk                          |
  |                                                                      |
  | Move cursor to desired item and press Enter.                         |
  |                                                                      |
  |   hdisk2  (00f6ecb5acf72d66) on all cluster nodes                    |
  |                                                                      |
  | F1=Help                 F2=Refresh              F3=Cancel            |
  | F8=Image                F10=Exit                Enter=Do             |
  | /=Find                  n=Find Next                                  |
  +----------------------------------------------------------------------+
```

Nothing should be defined on this disk, no volume group or logical volumes. PowerHA finds a suitable disks for selection and then create its own volume group. This disk is used for internal cluster information sharing and heartbeat information. It is fully reserved for the cluster.

Example 7-7 leaves the choice to PowerHA to select the right Multicast IP.

4. From the smitty cm_setup_menu, select **Setup a Cluster, Nodes and Networks** and then **Verify and Synchronize Cluster Configuration**.
5. Start cluster services on both nodes (**clstart**). This is a prerequisite for the subsequent tasks.

Verify

The CAA disk is created on both nodes and seen in the **lspv** command output as Example 7-8 shows.

Example 7-8 lspv command output

```
HDISK           PVIDVolumeGrpStatus
[…]
caa_private0   00f641d4f32707c6     caavg_private active
[…]
```

The lssrc **-ls clstrmgrES** command returns a cluster state of "ST_STABLE."

7.3.5 OS groups and users for SAP and SAP DB

The user IDs and group IDs of the SAP system on the operating system must be the same on all servers for all users. The user and group management can be performed by an SAP or third-party software or using the PowerHA internal facilities.

This section gives instructions for the PowerHA user management option.

Plan

The required users (Table 7-7 on page 154) and groups (Table 7-6) are named according to the SAP SID that is planned for this installation. If the database is also installed, additional groups and users are required. In Table 7-7 on page 154, the `<sid>` placeholder is to be replaced by the SID (in lowercase letters).

Make a list of all LPAR instances where the SAP system is running. For distributed user ID management, PowerHA also provides a Smart Assist to make your LDAP directory highly available. In addition, SAP and third-party tools provide user ID management. Select the technology that best fits your business requirements.

Important: Do not use the SAP global file system as the home directory for the SAP users. Smart Assist has removed all runtime dependencies on that directory to avoid disruption to business operations in case of an NFS outage.

Note: You can find a detailed list of prerequisites for users and groups on the SAP NetWeaver 7.4 page on the SAP website:

http://help.sap.com/nw_platform#section2

Users might change between releases, so please verify the correctness of this information.

Table 7-6 SAP Central Services groups

Group	Admin
sapinst	false
sapsys	false

Table 7-7 SAP Central Services users

User	Group	Home	Limits
`<sid>adm`	`pgrp=sapsys groups=sapsys,sapinst`	`home=/home/<sid>adm shell=/bin/csh`	`fsize=-1 cpu=-1 data=-1 stack=-1 core=-1 rss=65536 nofiles=32000`
`sapadm`	`pgrp=sapsys groups=sapsys,sapinst`	`home=/home/sapadm shell=/bin/csh`	`fsize=-1 cpu=-1 data=-1 stack=-1 core=-1 rss=65536 nofiles=32000`
`<dasid>adm`	`pgrp=sapsys groups=sapsys`	`home=/home/<dasid>adm shell=/bin/csh`	`fsize=-1 cpu=-1 data=-1 stack=-1 core=-1 rss=65536 nofiles=32000`

Install with PowerHA user management

The PowerHA C-SPOC facility can be used to manage users and groups cluster-wide. The smitty fastpath for it is **`smitty cl_usergroup`**.

Create OS groups

The following steps describe how to create OS groups:

1. **`smitty cl_usergroup`** → **Groups in a PowerHA SystemMirror cluster** and then **Add a Group to the cluster.**
2. Select the method to use. We used LOCAL for this scenario, as shown in Example 7-9.

Example 7-9 Selecting the authentication and registry mode

```
+--------------------------------------------------------------------------+
|                Select an Authentication and registry mode                |
|                                                                          |
| Move cursor to desired item and press Enter.                             |
|                                                                          |
|   LOCAL(FILES)                                                           |
|   LDAP                                                                   |
|                                                                          |
| F1=Help                 F2=Refresh              F3=Cancel                |
| F8=Image                F10=Exit                Enter=Do                 |
| /=Find                  n=Find Next                                      |
+--------------------------------------------------------------------------+
```

3. The design is based on a modular approach, so the group needs to be created on all nodes. Therefore, press Enter in the screen that follows without entering any details.
4. Create all groups (Example 7-10 on page 155) only with the credentials defined in Table 7-6 on page 153. Ensure that the same group IDs are defined on all cluster nodes.

Example 7-10   Define group

```
[Entry Fields]
  Select nodes by resource group
   *** No selection means all nodes! ***

* Group NAME                                  [sapinst]
  ADMINISTRATIVE group?                       false           +
  Group ID                                    []              #
  USER list                                   []              +
  ADMINISTRATOR list                          []              +
  Initial Keystore Mode                       []              +
  Keystore Encryption Algorithm               []              +
  Keystore Access                             []              +



F1=Help               F2=Refresh             F3=Cancel             F4=List
F5=Reset              F6=Command             F7=Edit               F8=Image
F9=Shell              F10=Exit               Enter=Do
```

Create OS users

The following steps describe how to create OS users:

1. **smitty cl_usergroup** → **Users in a PowerHA SystemMirror cluster** → **Add a user to the cluster.**
2. Select the method to use. For this example, we selected **LOCAL**.
3. The design is based on a modular approach, so the users need to be created on all nodes. Therefore, press Enter in the screen that follows without entering any details.
4. Create all users (Example 7-11) only with the credential defined in the Table 7-7 on page 154.

Example 7-11   Add user to cluster

```
Add a User to the Cluster

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

[TOP]                                                    [Entry Fields]
  Select nodes by resource group
   *** No selection means all nodes! ***

* User NAME                                          [ha2adm]
  User ID                                            []                   #
  ADMINISTRATIVE USER?                                false               +
  Primary GROUP                                      [sapsys]             +
  Group SET                                          [sapsys,sapinst]     +
  ADMINISTRATIVE GROUPS                              []                   +
  Another user can SU TO USER?                        true                +
  SU GROUPS                                          [ALL]                +
  HOME directory                                     [/home/ha2adm]
  Initial PROGRAM                                    []
  User INFORMATION                                   []
  EXPIRATION date (MMDDhhmmyy)                       [0]
  Is this user ACCOUNT LOCKED?                        false               +
  User can LOGIN?                                     true                +
```

```
  User can LOGIN REMOTELY?                          true                         +
  Allowed LOGIN TIMES                              []
  Number of FAILED LOGINS before                   [0]                           #
[MORE...33]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

5. Verify and synchronize the cluster configuration.

Verify

1. Verify on all LPARs that the SAP system is running the same name, ID, and tunables, as shown in Example 7-12.

Example 7-12 Verify users and groups

```
#lsuser -c -a id pgrp groups shell login fsize cpu data stack core rss nofiles ALL
      | grep sap
root:0:system:system,bin,sys,security,cron,audit,lp,sapinst:/usr/bin/ksh:true:-1:-
1:-1:-1:-1:-1:32000
ha1adm:205:sapsys:sapsys,sapinst:/bin/csh:true:-1:-1:-1:-1:-1:65536:32000
sapadm:207:sapsys:sapsys,sapinst:/bin/csh:true:-1:-1:-1:-1:-1:65536:32000
daaadm:208:sapsys:sapsys,sapinst:/bin/csh:true:-1:-1:-1:-1:-1:65536:32000

#lsgroup ALL | grep sap
sapinst id=210 admin=false users=root,ha1adm,sapadm,daaadm adms=root
registry=files
sapsys id=236 admin=false users=ha1adm,sapadm,daaadm
registry=files
```

2. Make sure that the ulimit of the root user is set to `-1` (unlimited) as shown in Example 7-13.

Example 7-13 Verify root settings

```
#ulimit -a
time(seconds)        unlimited
file(blocks)         unlimited
data(kbytes)         unlimited
stack(kbytes)        unlimited
memory(kbytes)       unlimited
coredump(blocks)     unlimited
nofiles(descriptors) unlimited
threads(per process) unlimited
processes(per user)  unlimited
```

7.3.6 IP alias considerations

The sections that follow describe the IP considerations for the SAP NetWeaver instances.

Plan

For the overall SAP architecture, especially clusters (SAP LVM and other SAP tools), the full capabilities are achieved only when installing each instance with a dedicated virtual IP.

Changing from a hostname-based installation to a virtual one typically requires the knowledge and skills to change it or reinstallation of the affected SAP systems.

The network infrastructure with its different zones should be planned by an SAP architect, because considerations range from communication aspects to security rules for specific SAP business applications. Ensure that the layout is fully redundant from a path perspective.

Table 7-8 gives IP planning guidance.

Table 7-8 IP aliases plan

Instance	Nodes	Network	Dedicated IP alias
ASCS	All	`<sap design>`	Not negotiable
SCS	All	`<sap design>`	Not negotiable
ERS for ABAP	All	`<sap design>`	Not negotiable
ERS for Java	All	`<sap design>`	Not negotiable
Application Server 1	All or local to node A	`<sap design>`	Optional, for test and development, non-production
Application Server 2	All or local to node B	`<sap design>`	Optional, non-production
Application Server *n*	All or local to node X	`<sap design>`	Optional, non-production
NFS crossmount	All	`<sap design>`	Not negotiable

Install

Prepare the virtual IPs by adding them into the `/etc/hosts` file on all nodes. They are brought online later.

Verify

Ensure that the `/etc/hosts` is the same on all nodes.

7.3.7 Create the file systems for the SAP installation

This section summarizes the file system creation options, based on the chosen disk layout.

Plan

Establishing the directory layout depends on the type of disk attachment chosen in 7.3.2, “Storage disk layout for SAP NetWeaver” on page 145.

Install

The setup of disks requires three steps:

1. Set up a local file system bound to a single node.
2. Set up a shared moving file system.
3. Set up NFS crossmount (optional).

Create node-specific local file systems

Create the file systems by using C-SPOC or the standard AIX methods. These file systems do need to be moved by the cluster nor must be monitored for availability.

Splitting a larger volume group (VG) into multiple logical volumes makes sense for production systems. In any case, the cluster configuration does not monitor the local volumes as part of the resource group (RG).

Besides the rootvg volume group, with its file systems and disks, the SAP file systems shown in Table 7-9 need to be created. The table shows the mount point and the suggested VG and logical volume (LV) naming conventions.

Table 7-9 Local SAP file systems

File system mount point	VG type or name	LV name	Comment
`/usr/sap`	`vgsap`	`lvsap`	Mandatory
`/usr/sap/<SID>`	`vgsap<SID>`	`lvsap<SID>`	Required in case of multiple SID installations on the same node
`/usr/sap/<SID>/<App1>`	`vgsap<SID><App1>`	`lvsap<SID><App1>`	Required only if node-bound application servers are planned (minimum 1 per node)
`/usr/sap/<SID>/<App2>`	`vgsap<SID><App2>`	`lvsap<SID><App2>`	Required only if node-bound application servers are planned (minimum 1 per node)

First, verify that the hdisks are not shared between nodes using the **lspv** command.

In Example 7-14, the AIX commands are listed for manual VG, LV, and file system creation. On each node, the commands are executed for local VGs.

Example 7-14 Commands to manually create VG, LV, and file system

```
mkvg -y <vgname> -S hdisk<no>

varyonvg <vgname>
mklv -y'<lvname>' -t'jfs2' -e'x' -u4 <vgname> <490> hdisk<no>

mkdir <mnt>
crfs -A -v jfs2 -d <lvname> -m <mnt> -p 'rw' -a agblksize=<4096> -a logname=INLINE
```

Note: This is a sample scenario. Sizes, types, and log names might differ.

Create the file system for the shared disk layout

Use the C-SPOC facility to create the file systems that are moved between the cluster nodes. This can also be scripted using AIX commands resulting in the same setup.

Perform the following steps for each shared disk:

1. Select **smitty cspoc → Storage → Volume Groups → Create a Volume Group**.
2. Select both nodes, as shown in Example 7-15 on page 159.

Example 7-15 Select both nodes

```
+--------------------------------------------------------------------------+
|                                Node Names                                |
|                                                                          |
| Move cursor to desired item and press F7.                                |
|     ONE OR MORE items can be selected.                                   |
| Press Enter AFTER making all selections.                                 |
|                                                                          |
| > as0003lx                                                               |
| > as0004lx                                                               |
|                                                                          |
| F1=Help                 F2=Refresh              F3=Cancel                |
| F7=Select               F8=Image                F10=Exit                 |
| Enter=Do                /=Find                  n=Find Next              |
+--------------------------------------------------------------------------+
```

3. Pick the hdisk according to Example 7-16.

Example 7-16 Select the physical volume, for the shared disk

```
+--------------------------------------------------------------------------+
|                          Physical Volume Names                           |
|                                                                          |
| Move cursor to desired item and press F7.                                |
|     ONE OR MORE items can be selected.                                   |
| Press Enter AFTER making all selections.                                 |
|                                                                          |
| > 00f6ecb5112266c9 ( hdisk5 on all cluster nodes )                       |
|                                                                          |
| F1=Help                 F2=Refresh              F3=Cancel                |
| F7=Select               F8=Image                F10=Exit                 |
| Enter=Do                /=Find                  n=Find Next              |
+--------------------------------------------------------------------------+
```

4. Configure the volume group as shown in Figure 7-9, and adjust your sizing and mirroring prerequisites.

   In our example and as a best practice, the mirroring is performed at the storage level, using IBM SVC technology. Therefore, this is not configured here.

```
+--------------------------------------------------------------------------+
|                            Volume Group Type                             |
|                                                                          |
| Move cursor to desired item and press Enter.                             |
|                                                                          |
|   Legacy                                                                 |
|   Original                                                               |
|   Big                                                                    |
|   Scalable                                                               |
|                                                                          |
| F1=Help                 F2=Refresh              F3=Cancel                |
| F8=Image                F10=Exit                Enter=Do                 |
| /=Find                  n=Find Next                                      |
+--------------------------------------------------------------------------+
```

Figure 7-9 Select Volume Group type

The VG type selected in Figure 7-9 on page 159 is **Scalable**.

5. Configure the VG and adjust your sizing and mirroring prerequisites as shown in Figure 7-10.

Attention: Leave the resource group name empty.

```
Create a Scalable Volume Group

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

                                                          [Entry Fields]
  Node Names                                           as0003lx,as0004lx
  Resource Group Name                                 []                      +
  PVID                                                 00f6ecb5112266c9
  VOLUME GROUP name                                   [<vgname>]
  Physical partition SIZE in megabytes                 4                      +
  Volume group MAJOR NUMBER                           [39]                     #
  Enable Fast Disk Takeover or Concurrent Access       Fast Disk Takeover     +
  Volume Group Type                                    Scalable
  CRITICAL volume group?                               no                     +

  Maximum Physical Partitions in units of 1024         512                    +
  Maximum Number of Logical Volumes                    256                    +
  Enable Strict Mirror Pools                           no                     +
  Mirror Pool name                                    []
```

Figure 7-10 Add the Scalable Volume Group

6. To create the logical volumes (one or more) for the VGs that you created, select **smitty cspoc → Storage → Logical Volumes → Add a Logical Volume**.

   Inline logs have the advantage of moving automatically along with the file system whenever the owning RG is moved. A dedicated file system log is generally a good choice, performance-wise. See Figure 7-11 on page 161.

Rule of thumb: For database data volumes, the dedicated file system log is preferable. For SCS and ERS, where few changes apply, inline logs are fine.

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

[TOP]                                                    [Entry Fields]
  Resource Group Name                                  <Not in a Resource Group>
  VOLUME GROUP name                                    vgascscss
  Node List                                            as0003lx,as0004lx
  Reference node                                       as0003lx
* Number of LOGICAL PARTITIONS                        [1000]                  #
  PHYSICAL VOLUME names                                hdisk5
  Logical volume NAME                                 [lvascscss]
  Logical volume TYPE                                 [jfs2]                  +
 [MORE...26]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 7-11 Adding a logical volume

7. Create the file systems for the previously created volume groups by using C-SPOC as follows: **smitty cspoc → Storage → File Systems → Add a File System**. Select the appropriate volume group and create the file systems as *enhanced journaled* file systems according to the deployment-specific sizing. See Figure 7-12.

```
 Add an Enhanced Journal File System on a Previously Defined Logical Volume

[TOP]                                          [Entry Fields]
  Resource Group                               <Not in a Resource Gr>
* Node Names                                  as0007lx,as0008lx
  Logical Volume name                         lvascscss
  Volume Group                                vgascscss
* MOUNT POINT                                 [/usr/sap/<SID>/ASCS00]
  PERMISSIONS                                 read/write
  Mount OPTIONS                              []
  Block Size (bytes)                          4096
  Inline Log?                                 yes
```

Figure 7-12 Adding an Enhanced Journal File System

8. Verify and synchronize the cluster configuration

Using SAP global file systems (optional)

The focus here is an NFS v3-based crossmount, which is the typical solution for SAP systems. Keep in mind the deployment considerations highlighted in “Surrounding components: SAP global and SAP transport directories” on page 137.

These are alternatives, but they are not within the scope of this chapter:

- Using the NFS v4 Smart Assist for NFS provides the capability to assist in the setup.
- In all other cases, ensure that the file systems are mounted or attached to all nodes and are eligible to be automatically mounted.

Be sure to consider the deployment considerations highlighted in “Surrounding components: SAP global and SAP transport directories” on page 137.

Procedure for Smart Assist for NFS v3 crossmount deployment

The following procedure is for a Smart Assist for NFS v3 crossmount deployment:

1. Create the shared VG with its file systems as described in "Create the file system for the shared disk layout" on page 158.

   Do not enter a resource group at this point. Decide if this crossmount serves `/sapmnt` (all SIDs) or `/sapmnt/SID` (one per SID) and adjust the mount point accordingly in case one crossmount per SID. See Table 7-10.

   Table 7-10 hdisk, mount point, VG type

hdisk Node1, Node2	File system mount point	VG type or name	LV name
hdisk7, hdisk3 same pvid	`/export/usr/sap/trans`	Enhanced concurrent in none concurrent mode: vgtrans	lvtrans
hdisk6 / hdisk5 same pvid	`/export/sapmnt/`	Enhanced concurrent in none concurrent mode: vgsapmnt	lvsapmnt

2. Add the service IP to the `/etc/hosts` file on each node.
3. Create the service IP alias by selecting **smitty sysmirror → Cluster Applications and Resources → Resources → Configure Service IP Labels/Addresses → Add a Service IP Label/Address**. See Figure 7-13.

```
              Add a Service IP Label/Address
[Entry Fields]
* IP Label/Address                                    as0009lx
  Netmask(IPv4)/Prefix Length(IPv6)                        []
* Network Name                                        net_ether_01
```

Figure 7-13 Adding a service IP label and address

4. Create the mount directories */sapmnt* and */usr/sap/trans* on both nodes (**`mkdir -p <dir>`**).
5. Select **smitty sysmirror → Cluster Applications and Resources → Make Applications Highly Available (Use Smart Assists) → NFS Export Configuration Assistant → Add a Resource Group with NFS exports**. See Figure 7-14 on page 163.

```
Add a Resource Group with NFS exports

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

                                                        [Entry Fields]
* Resource Group Name                                [rg_nfs]
* Primary Node                                       [as0003lx]                           +
* Takeover Nodes                                     [as0004lx]                           +
* Service IP Label                                   [as0009lx]                           +
  Netmask(IPv4)/Prefix Length(IPv6)                  []
* Volume Groups                                      [vgsapmnt vgtrans]                   +
* Filesystems/Directories to Export (NFSv2/3)        [/export/sapmnt /export/usr/sap/trans] +
* Filesystems/Directories to Export (NFSv4)          [NONE]                               +
* Filesystems/Directories to NFS Mount               [/sapmnt:/export/sapmnt]             +
* Stable Storage Path (NFSv4)                        [AUTO_SELECT]                        +

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 7-14 Adding a resource group with NFS exports

6. Edit the resource group configuration and change it to “Online on first available node” rather than “Home node only.”
7. Synchronize and verify the cluster.

> **Note:** If this step fails, you might must remove the rg_nfs manually before retrying. Otherwise, the following error message appears:
>
> ERROR: The specified object already exists: “rg_nfs.”

Verify

1. Use the follow AIX commands to identify and verify the mapping between the storage disk and hdisk:
   - **`getconf DISK_SIZE /dev/hdisk<x>`**
   - **`lspv`** (compare pvids)
   - **`lscfg –vpl hdisk<x>`**
   - **`lsdev –Cc disk`**
   - **`lsdev –Cc disk –F ‘name physloc’`**
2. Ensure that the local file systems are identical on all nodes.
3. Verify LPARs external to the cluster that host additional application server instances.

7.3.8 Bring the IP and file system resources online

SAP Smart Assist automatically adds and verifies the resources. The most convenient approach is to bring the SAP-required resources online manually to prepare for the SAP system installation.

> **Note:** Check to be sure that the cluster services are running and the cluster is synchronized and verified. NFS crossmounts must be mounted. You can use the **`clmgr online cluster`** command to assist with this task.

Plan

Bring the following resources online:

On node 1

- All node-specific IPs that belong to node 1 (for example, the application server instances).
- All service IPs that move along with a resource group. At this point, you can activate the IPs of SCSes or ASCSes and of ERSes on the same node. This makes the SAP installation process more convenient. Smart Assist handles all other necessary items.

For the file system resources:

- All node-specific file systems should be mounted.
- All shared file systems moving along with an RG.

On node 2

On node 2, bring online the following IP resources:

- All node-specific IPs that belong to node 2 (for example, the application server instances).
- All IPs that have not been brought online on node 1 yet but are required for the SAP installation

For file system resources:

- All node-specific file systems should be mounted.
- All shared file systems which have not been brought online on node 1 yet but are required for the SAP installation.

Install

In this section, we use a few AIX commands to configure the environment.

AIX commands

Execute the following commands on the node that the resource belongs to.

For the virtual IP address:

```
ifconfig <netw e.g. en0> alias <vip> netmask xxx.xxx.xxx.xxx
```

Varyon enhanced concurrent volume groups:

```
#varyonvg -c <vgname>
#mount <mountpoint>
```

The **`lspv`** command shows that the VG is online in “concurrent” mode.

> **Troubleshooting varyonvg -c**
>
> In some cases, **`varyonvg -c`** does not work due to disk reservations. If you check and the VG is not varied **`-on`** for any other node, you can recover as Example 7-17 on page 165 shows.

Example 7-17 Troubleshooting varyonvg -c

```
#varyonvg -O <vg name>

#mount -o noguard <mount point e.g. /usr/sap/SID/ASCSxx>
mount: /dev/lv01 on <mount point e.g. /usr/sap/SID/ASCSxx>
Mount guard override for file system.
The file system is potentially mounted on another node.
Replaying log for /dev/lv01.

#umount <mount point e.g. /usr/sap/SID/ASCSxx>
#varyoffvg  <vg name>
#varyonvg -c <vgname>
#mount <mountpoint>
```

7.3.9 Final preparation

Two final steps are required before starting the SAP installation.

Check SAP instance numbers

Check that the designated instance numbers are not duplicated on any host. Take into consideration which instances can potentially be moved to which host in all combinations (also consider SAP LVM operations, if used).

Verify SAP default ports

Verify that the default SAP ports are not used for other services in `/etc/services`:

- If you do not use duplicate ports, remove the non-SAP entries.
- If required, enter free port numbers during the installation.

A list of SAP ports can be found during the installation or in the installation guide on the SAP NetWeaver 7.4 web page:

http://help.sap.com/nw_platform

7.4 Install SAP NetWeaver as highly available (optional)

The following steps need to be performed:

1. Identify the SAP software and SAP manuals.
2. Set up the SAP installer prerequisites.
3. Install the SAP Central Services for ABAP or Java.
4. Install the ERS for ABAP or Java.
5. Install the redundant application server instances.
6. Install the add-ons.

7.4.1 Identify the SAP software and SAP manuals

For full capabilities, including the SAP HA API in the version 1.0, SAP requires a minimum NetWeaver 7.30 with a 7.20 Kernel Patch Level (PL) of 423. Verify the minimum SAP HA API level matching your SUM SP level.

For non-SAP HA API-enabled deployments, any NetWeaver version, starting from version 7.00 and based on Kernel 7.20, is supported.

Read the following SAP Notes:

- SAP Note 1693245
- SAP Note 1751819 (The scripts use EnqGetStatistics. Required minimum: 720 PL42.)
- SAP Note 1678768
- SAP Note 181543 (Read before you request the SAP license.)

The SAP installation manuals can be found on the SAP NetWeaver 7.4 web page:

http://help.sap.com/nw_platform

The SAP software used for this chapter is the SAP NetWeaver 7.40 with an SAP Kernel of 7.20, Patch 402, and patches for the enqueue server executable file of level 423.

7.4.2 Set up the SAP installer prerequisites

In this section, for a standard installation of SAP NetWeaver 7.40, the following preparation tasks must be performed. However, the master document for the SAP installation is the official SAP installation guide.

Start on the host where the majority of the file systems and virtual IP are brought online during the node preparation. Then continue with the other hosts.

1. Establish an X forward. In this scenario, VNC was used. The following steps are required:
   a. Install the VNC server executable on AIX.
   b. Download the VNC client to your workstation and start the VNC server on the LPAR:
   ```
   #vncserver
   You will require a password to access your desktops.
   Password:
   Verify:
   New ‘X’ desktop is <host>:<display no e.g. 1>
   Creating default startup script /home/root/.vnc/xstartup
   Starting applications specified in /home/root/.vnc/xstartup
   Log file is /home/root/.vnc/<host>:1.log
   ```
   c. Export the display on the host:
   ```
   export DISPLAY=:1
   ```
2. Connect through the VNC client:
   ```
   <host>:1
   ```
3. Verify that it is working from the client. For example, execute **xclock** to verify whether a window is opened.
4. Create the SAP installation directories:
   a. Use at least two levels above /. Typically: `/tmp/<my sapinstdir>`
   b. Allocate enough space to `/tmp` (min 5 GB free space).

7.4.3 Run the SAP Software Provisioning Manager verification tool

In this section, the SAP Software Provisioning Manager verification tool is used by completing the following steps:

1. Download the Software Provisioning Manager from the SAP Service Market Place:

   http://service.sap.com/swdc

   Select **Support Packages and Patches à A – Z Index à S à SLToolset à SLToolset <release> à Entry by Component à Software Provisioning Manager à Software Provisioning Manager 1.0 à <Operating System>**.

2. Unpack the Software Provisioning Manager (SWPM) archive to a local directory. The SAPCAR is on your installation media on the Kernel CD:

   ```
   SAPCAR –xvf <download directory>/<path>/<Archive>.SAR -R <unpack directory>
   ```

3. Run the environment verification option from inside your install directory:

   ```
   <unpack directory >/sapinst
   ```

For this book, we focus on the SCS and ERP along with a primary application server. Select all options that apply to your specific setup. See Figure 7-15.

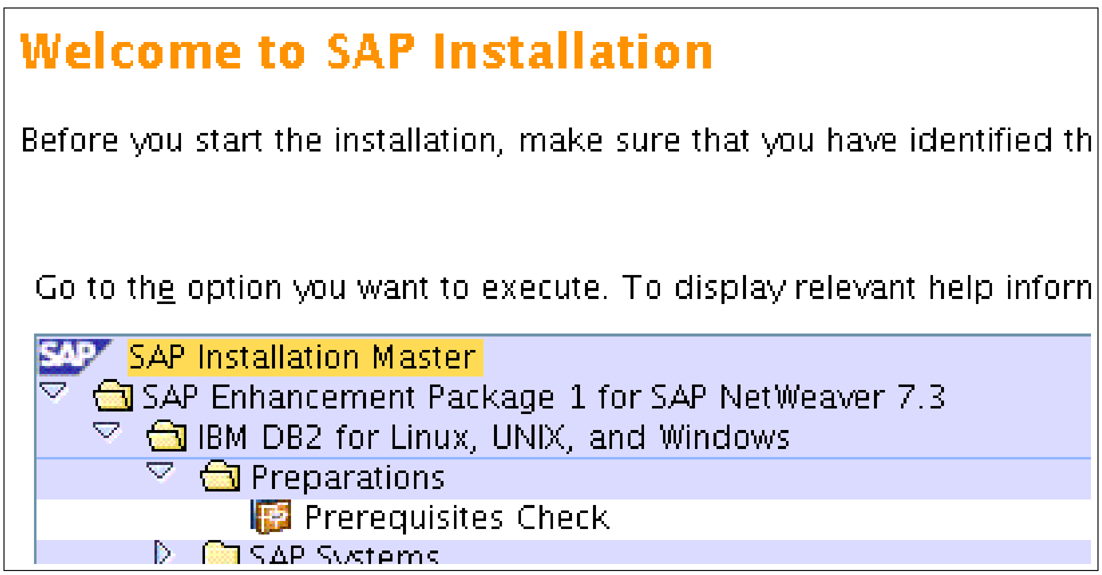


Figure 7-15 SWPM: Start prerequisite check

Select the instance types shown in Figure 7-16.

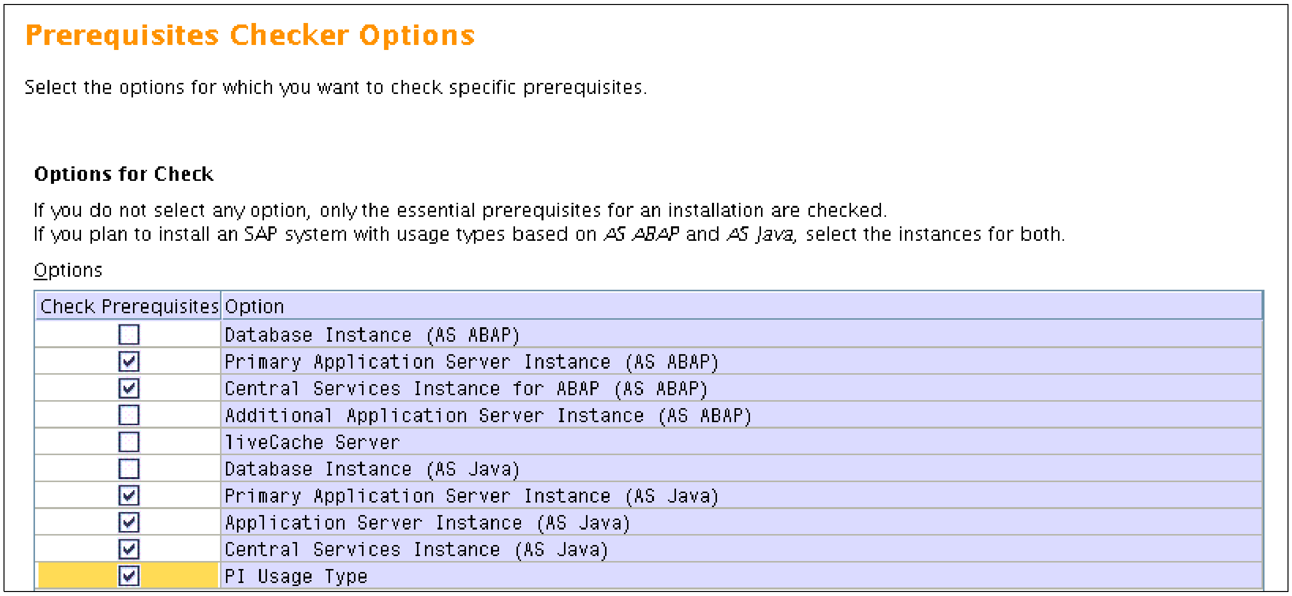


Figure 7-16 SWPM: Select instance types

In the following two panels, select your SAP kernel CD in the Media Browser as input and verify the parameters in the Parameter Summary overview.

Review the results and resolve any conflicts as shown in Figure 7-17.

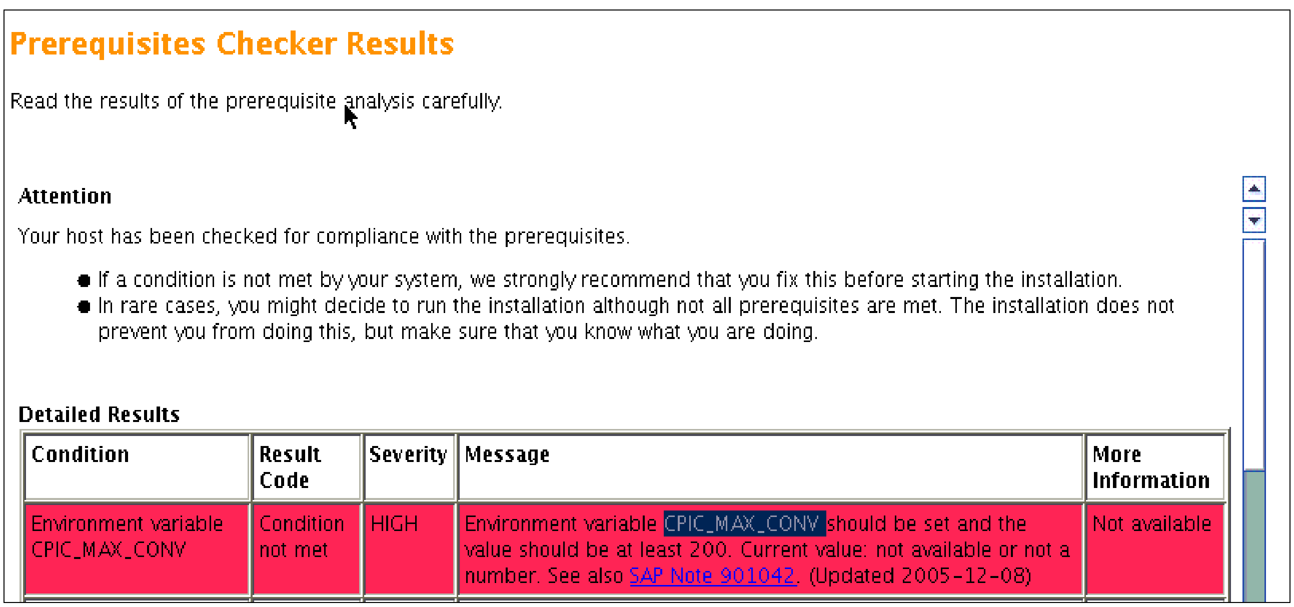


Figure 7-17 SWPM: Verification results

In this case (Figure 7-17), the *CPIC_MAX_CONV* variable was missing in `.sapenv_<>.sh` or `.csh` in the `sidadm` user home.

Finish the verification and maintain a copy of the successful completion as shown in Figure 7-18.

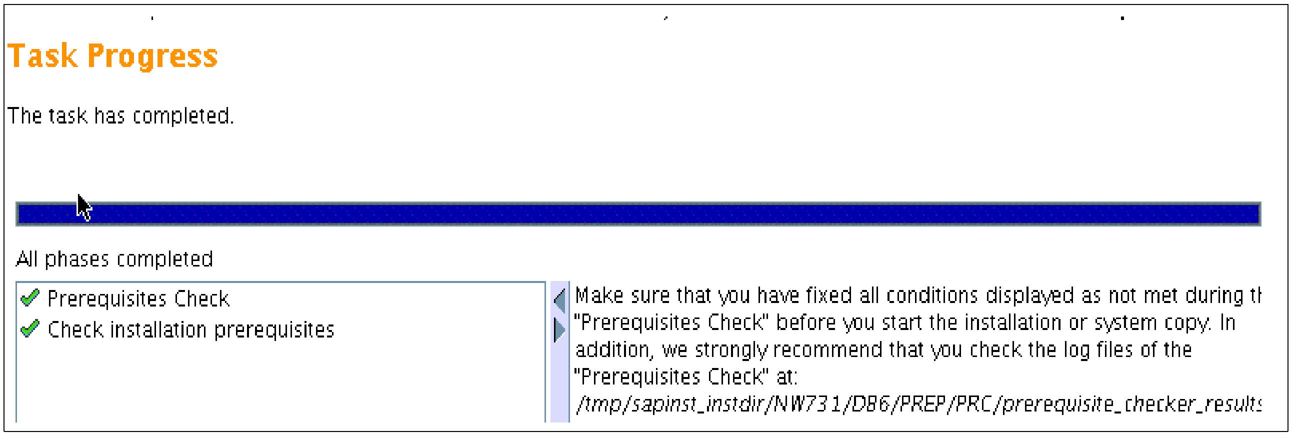


Figure 7-18 SWPM: Verification completion

7.4.4 SAP NetWeaver installation on the primary node

The high-level flow when implementing the IBM PowerHA SystemMirror 7.1.3 is described in the following sections, with examples. This information does not replace the official SAP sizing, tuning, and installation guides. Also, keep in mind that this installation is performed on a simple demonstration system.

Install Central Services and ERS

Run **sapinst** from a new or empty installation directory. If you would like to keep the installation information, empty the directory before continuing. If you encounter problems during the installation, you will need the installation logs. Therefore, we recommend keeping all installation logs until the installation is completed successfully.

Central Services (CS) stands for either of the two stacks: ABAP and Java. If a dual stack is installed for a cluster, this task is performed twice: once for ASCS and once for SCS. The examples shown here are from an ABAP stack.

1. Start the SWPM installer:

   ```
   …/sapinst SAPINST_USE_HOSTNAME=<ip alias for CS instance>
   ```

2. Provide basic configuration data for the SAP system and SAP users. The next screens require the following information:

   – Choose to use either the FQDN (fully qualified domain name) or short hostname setups. All verifications based on the example installation are performed by using short hostname installations.

   – In the Media Browser, provide the Kernel media.

   – In the Master Password panel, choose your master password for all users.

   – In the two SAP System Administrator panels that follow, ensure that the User and Group IDs match the previously created IDs. If you have skipped this preparation step, ensure that these IDs are available on all nodes, and create the users accordingly. See Figure 7-19 on page 170.

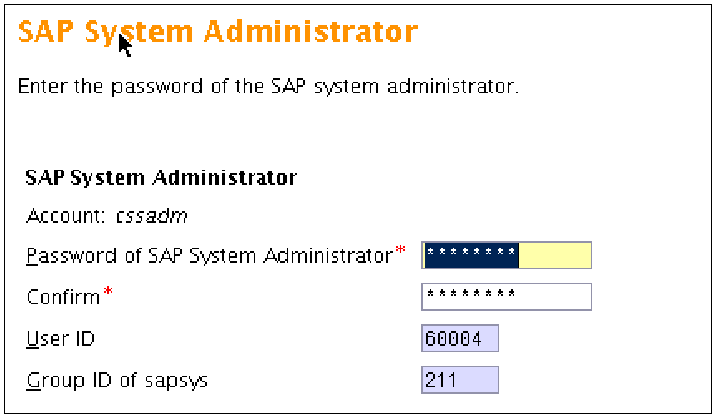


Figure 7-19 CS installation: Verify that user and group IDs are available on all nodes

Create the SCS instance identifiers and ERS automation

Depending on the NetWeaver release, the ERS preparation step looks like Figure 7-20.

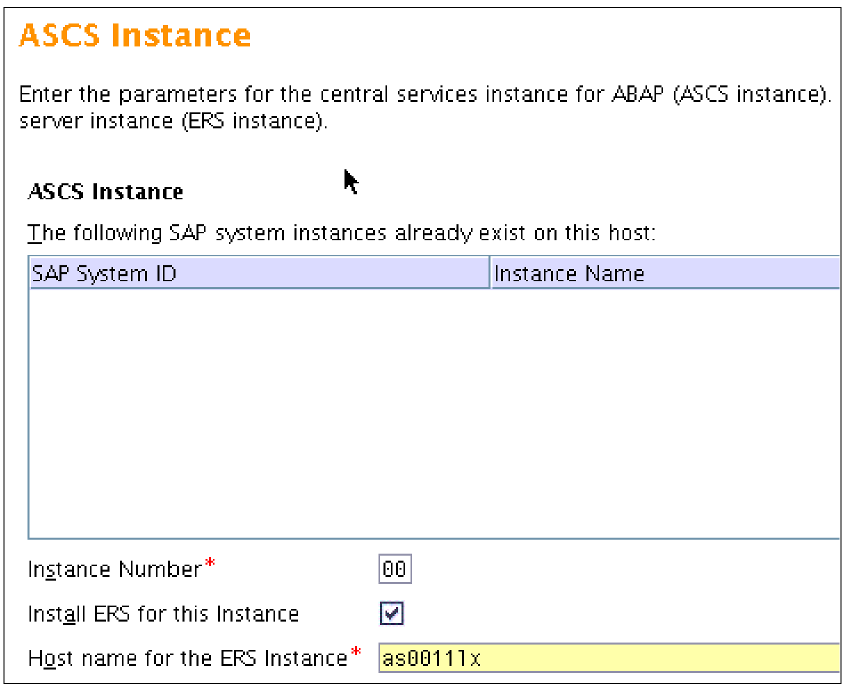


Figure 7-20 CS installation: Create the SCS and ERS

Before creating the ports, ensure that you have prepared `/etc/services` on all nodes. Figure 7-21 shows the message port menu.

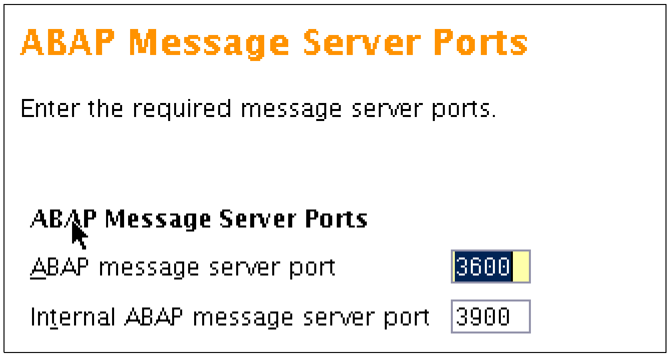


Figure 7-21 CS installation: Create message server ports

Check and run installation

Verify the parameters in the final screen, and then run the installation (see Figure 7-22).

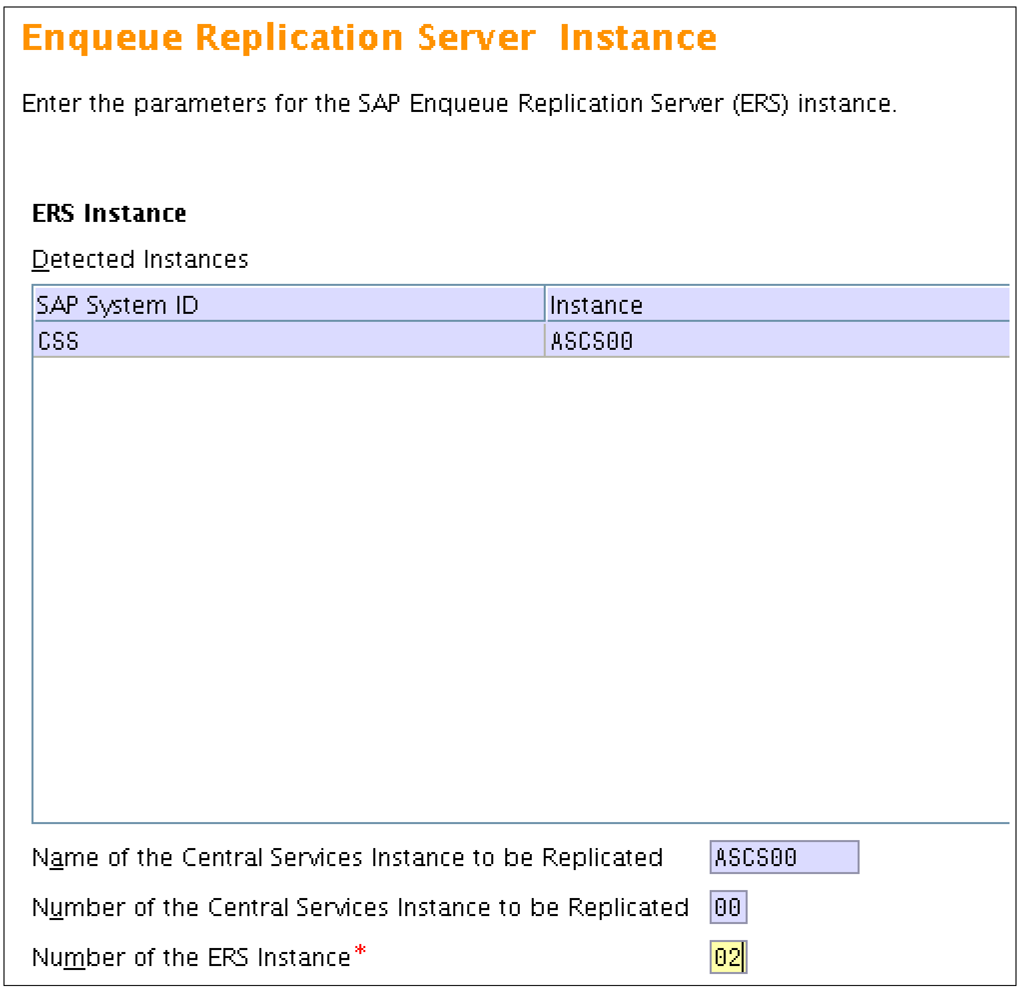


Figure 7-22 CS installation: Finalize ERS setup

If a dual-stack environment is installed, repeat the previous steps with dedicated IPs, ports, and numbers. Also, configure the tuning and sizing to the specific workload that is expected.

Database installation

The SAP DB instance must be installed before the SAP application server. The database can reside on the same pair of LPARs, but it is typically installed on its own dedicated pair of LPARs.

This dedicated LPAR choice has its advantages, especially when using hot standby database technology. Also, maintenance and administration are easier and less risky with a clear separation between the database server and the application server (AS). This should also help in terms of uptime and migrations.

Depending on the type and vendor of the database, you can find documentation links in 7.10, "Documentation and related information" on page 214.

For more information about clusters for SAP landscapes, contact your IBM representative or the ISICC information service (isicc@de.ibm.com).

IBM solutions are available for the following databases:

- IBM
  - DB2
  - DB2 HADR
  - DB2 PureScale
- Oracle
  - Oracle Data Guard
  - Oracle RAC
- maxDB
- liveCache
- Sybase

SAP application server instance and DAA installations

The SAP concept for the application servers is to build them in a redundant manner rather than setting them up in a failover configuration. This is because the failover times are typically unacceptable from a business point of view. Therefore, verify that the SAP components are installed and enabled in a redundant fashion for high-availability purposes.

The following installation is based on an ABAP stack. Perform these steps on redundant hosts located on different physical servers.

For the name and number, there are two considerations:

- The name and number are a specific ID for SAP on a per-host basis. Separating instances on different hosts makes it easier to reuse the same name and number.
- If using SAP tools and cluster control to move instances between nodes, it is mandatory to install them on dedicated disks with dedicated IPs.

Run **`sapinst`** from a new or empty installation directory. If you would like to keep the installation information, empty the directory before continuing. If you encounter problems during the installation, you will need the installation logs. We recommend keeping all installation logs until the installation is completed successfully.

Start the SWPM installer

In this step we start the SWPM installer: `.../sapinst SAPINST_USE_HOSTNAME=<ip alias for the app server instance>`.

Figure 7-23 shows the initial installer screen.

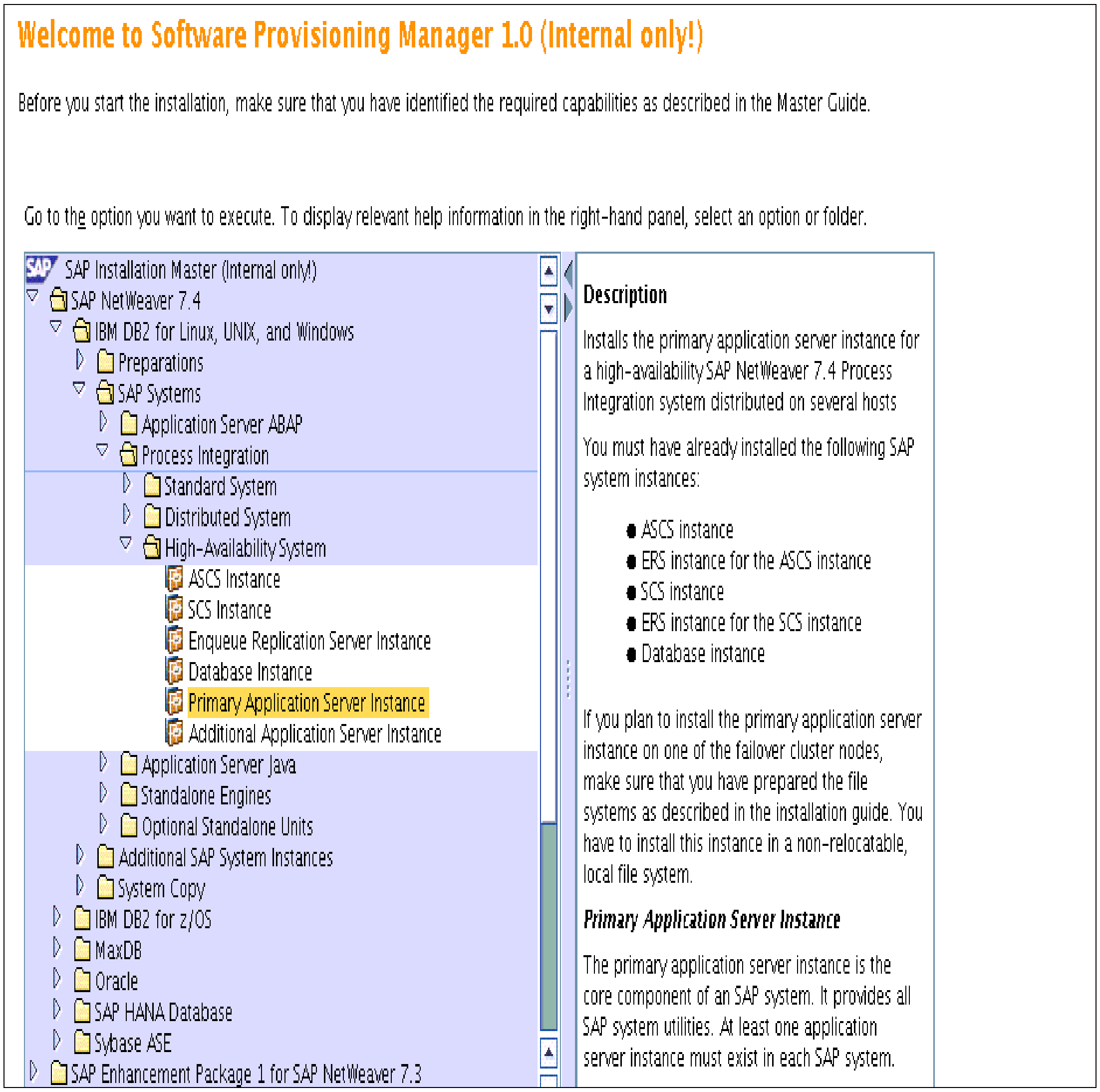


Figure 7-23 AS installation: Select Primary AS instance and use VIP

Provide basic configuration data for the SAP system and users

Enter the SAP profile directory, as shown in Figure 7-24.

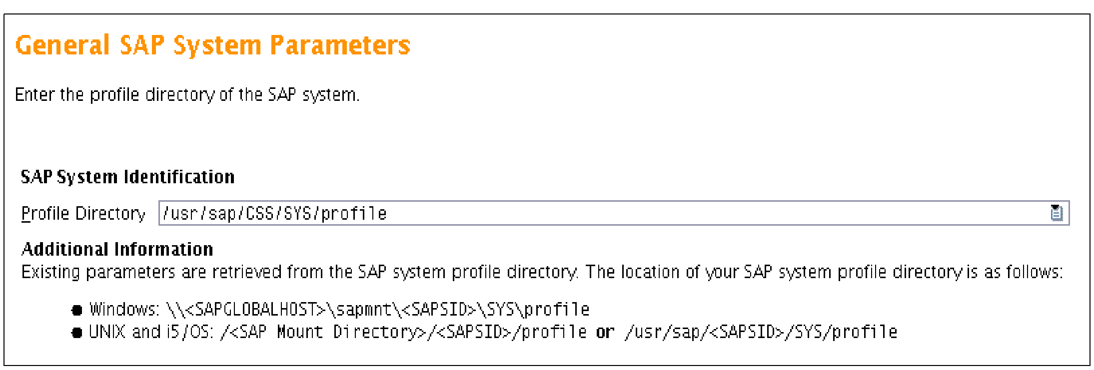


Figure 7-24 AS installation: Provide global profile directory

The following screens require the following information:

- Choose to use either the FQDN (Full Qualified Domain Name) or short hostname setup. All verifications based on the example installation are performed based on short hostname installations.
- In the Media Browser screen, provide the Kernel media.
- Enter your master password for all users in the Master Password panel.

Provide the database-connect users for the ABAP or Java schema. For the example, we used the following defaults:

For ABAP, these are the defaults:

- Schema = `sap<sid>`
- User = `sap<sid>`

For Java,these are the defaults:

- Schema = `sap<sid>db`
- User = `sap<sid>db`

Provide the remaining required media as prompted by the Media Browser panel.

Create application server instance identifiers

Define the AS instance number as shown in Figure 7-25.

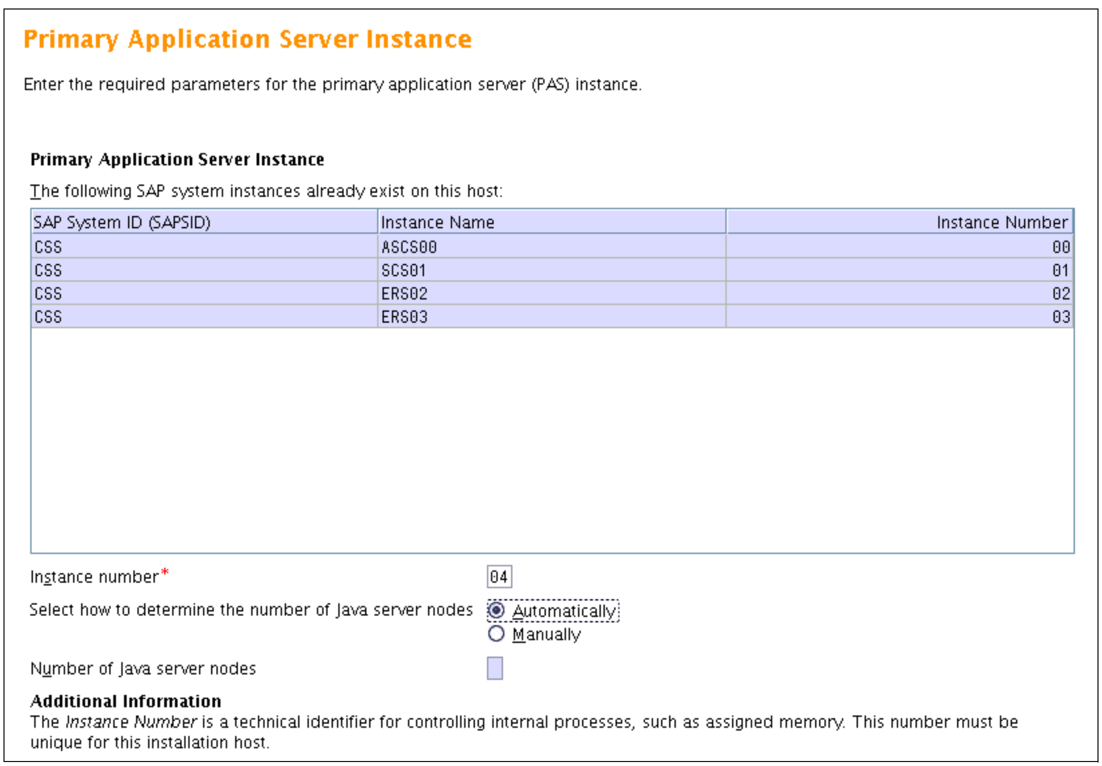


Figure 7-25 AS installation: Define the AS instance number

Provide the message server ports as defined during the SCS installation step (see Figure 7-26).

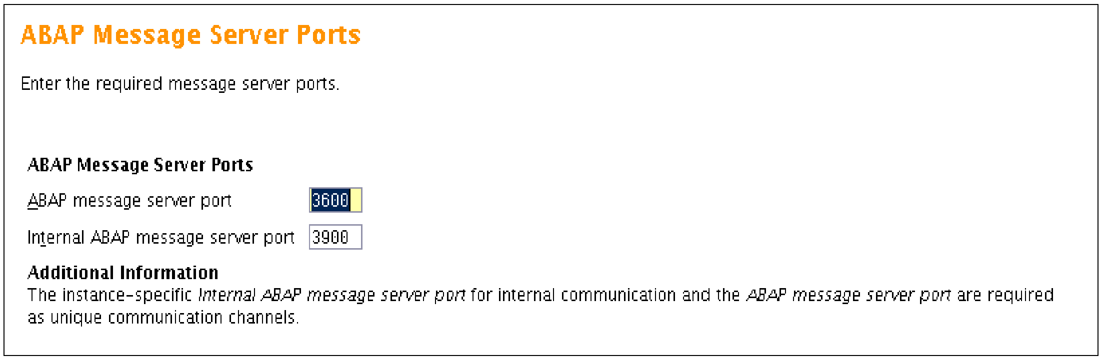


Figure 7-26 AS installation: Provide message server ports for SCS instance

Define user and RAM parameters

For the RAM definitions, review the SAP Installation Guide and use the recommended sizing:

`http://help.sap.com/nw_platform`

Figure 7-27 shows the memory requirements for our example environment.

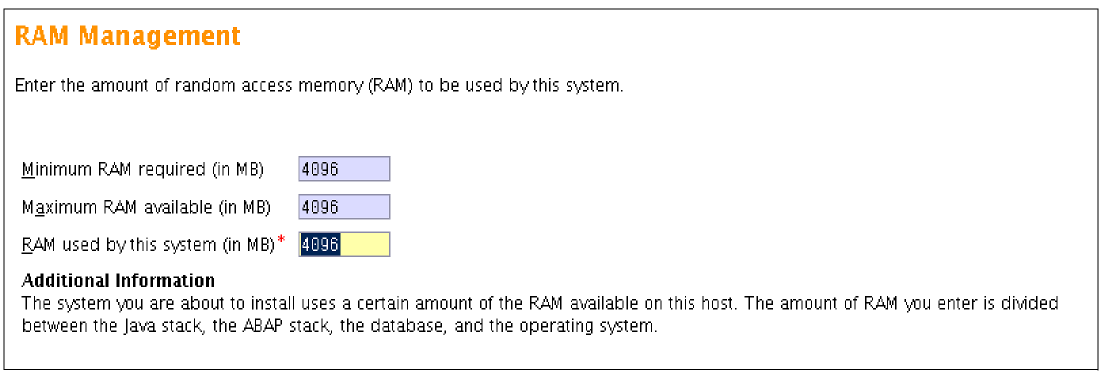


Figure 7-27 AS installation: RAM management

Enter the Internet Communication Manager (ICM) user password, as shown in Figure 7-28.

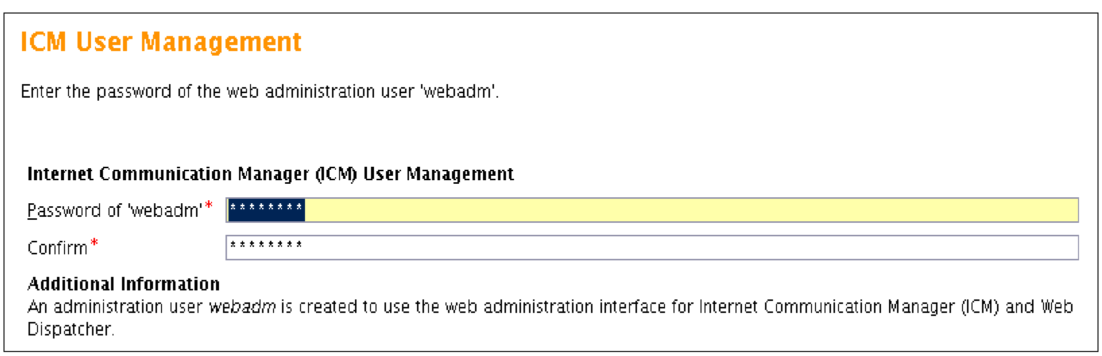


Figure 7-28 AS installation: ICM user management

Enter ABAP UME users, as shown in Figure 7-29.

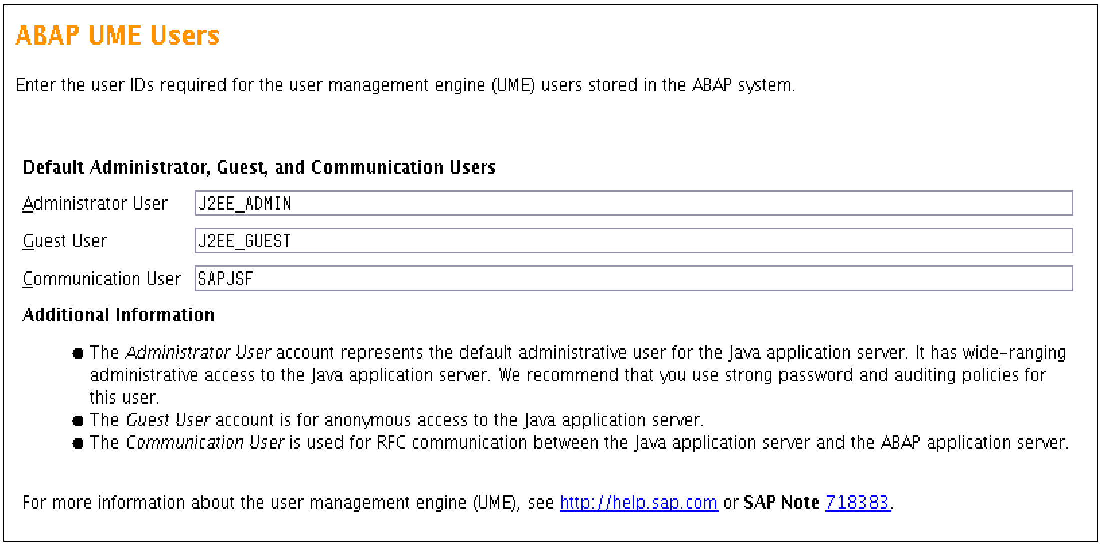


Figure 7-29 AS installation: ABAP UME users

Enter the administrator and communication user passwords, as shown in Figure 7-30.

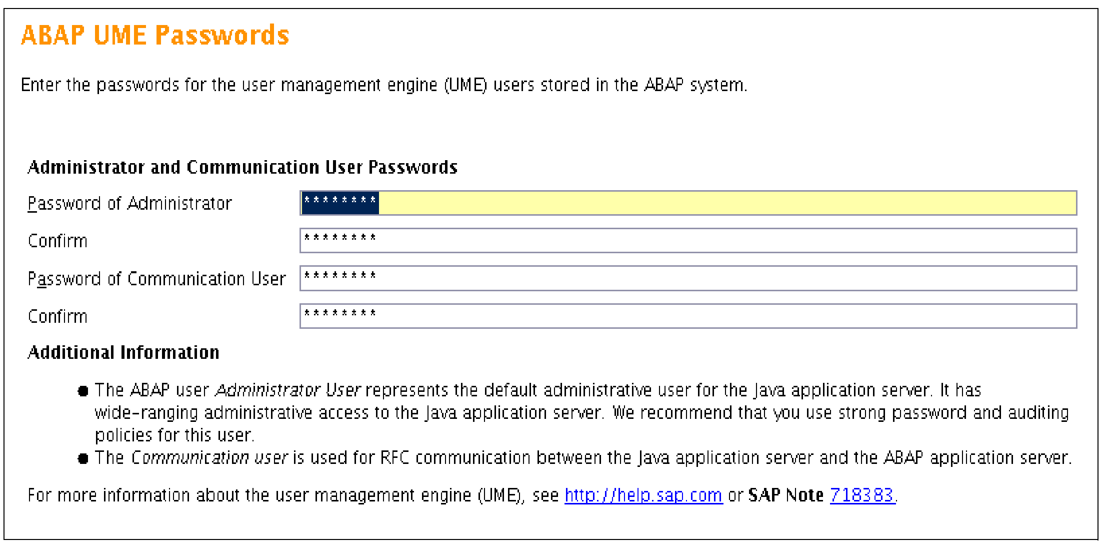


Figure 7-30 AS installation: ABAP UME passwords

Enter the DDIC (data dictionary) user password, as shown in Figure 7-31.

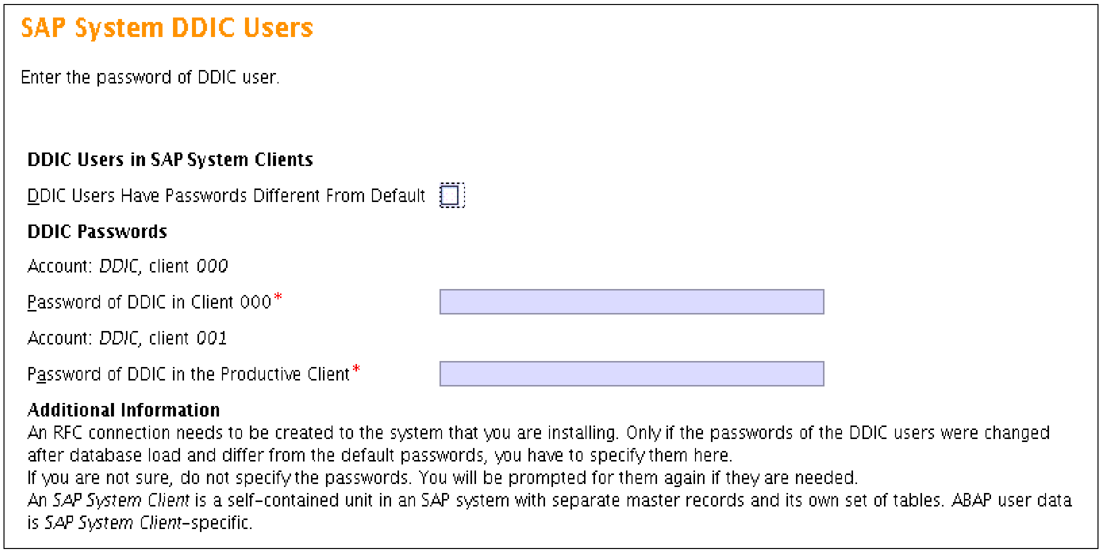


Figure 7-31 AS installation: SAP system DDIC users

Select archives to unpack, as shown in Figure 7-32.

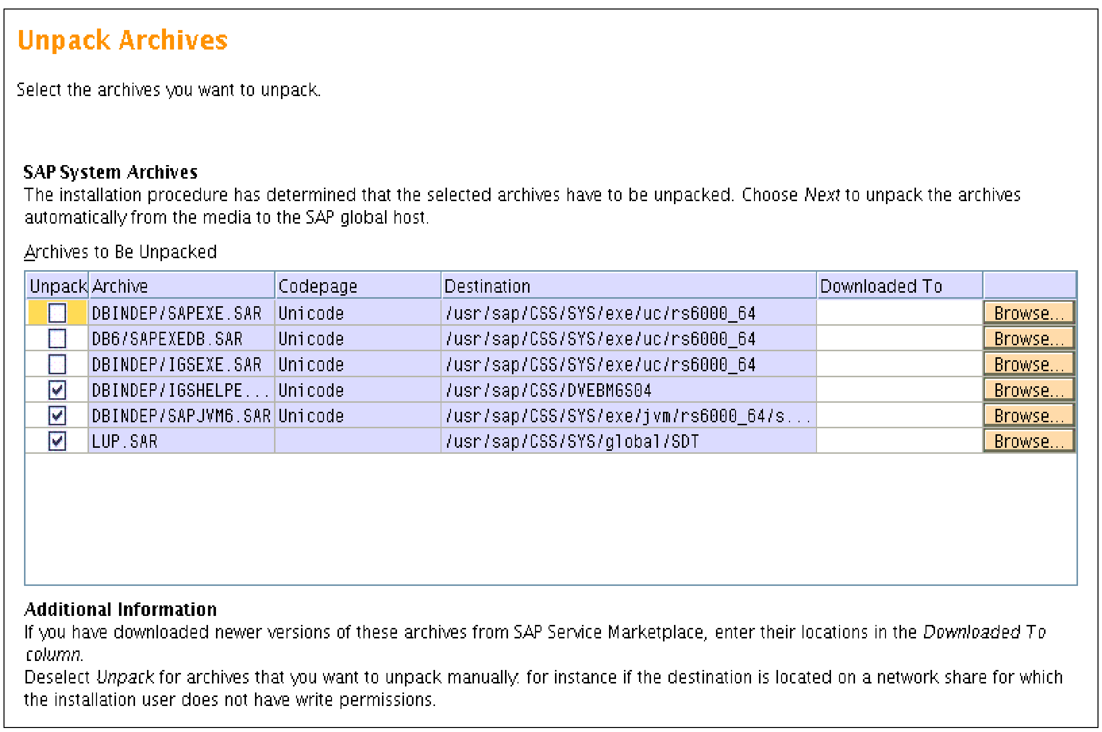


Figure 7-32 AS installation: Unpack archives

Define the SAP Diagnostic Agent (DAA)

Initiate the DAA installation and define the DAA system ID as shown in Figure 7-33.

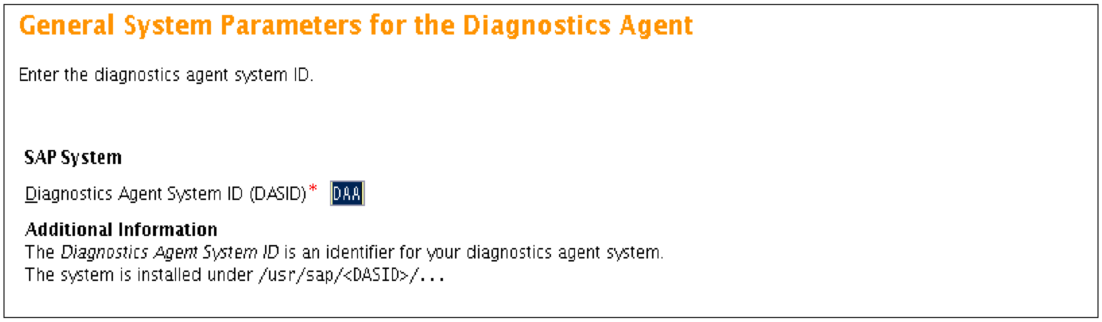


Figure 7-33 DAA installation: Define the DAA system ID

Enter the system administrator password, as shown in Figure 7-34.

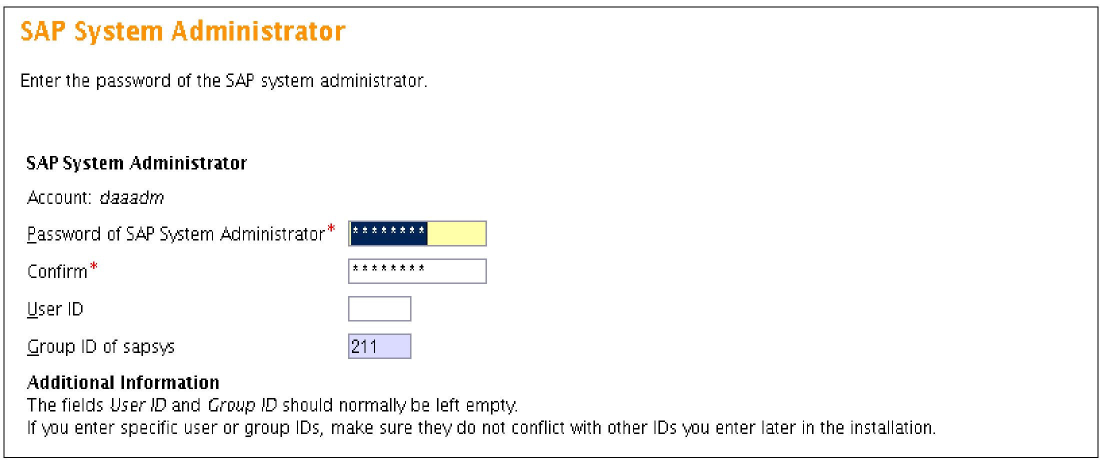


Figure 7-34 DAA installation: SAP system administrator

If the daaadm user is not in the profile, you will get the warning shown in Figure 7-35.

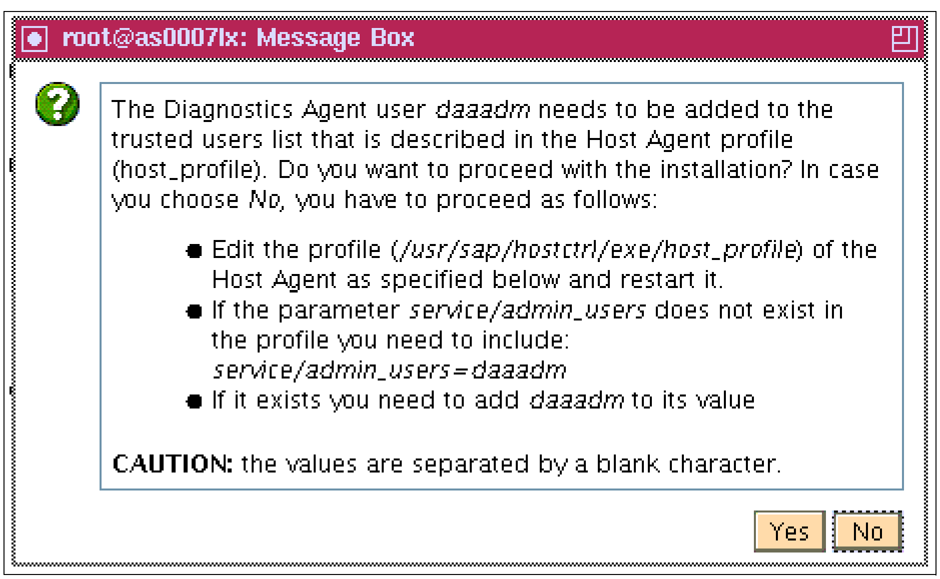


Figure 7-35 DAA installation: daaadm user warning

Define a unique instance number as shown in Figure 7-36.

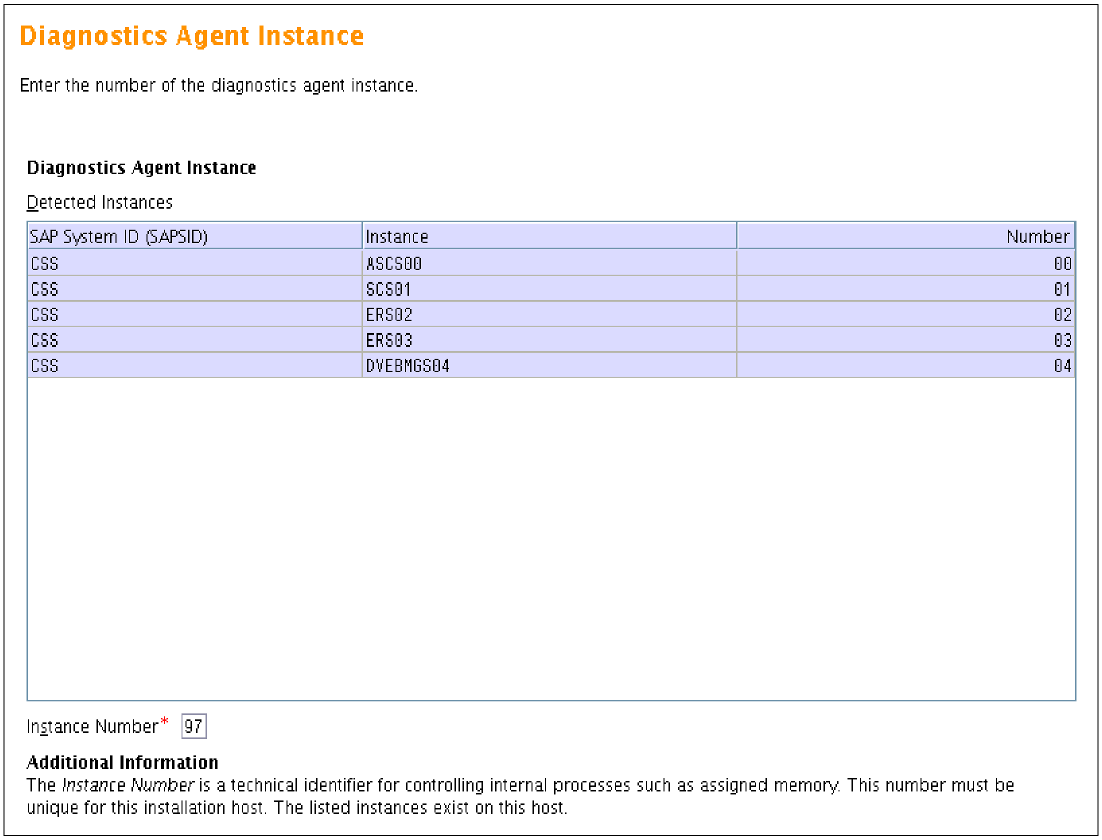


Figure 7-36 DAA installation: Define unique instance number

Define the SLD destination as shown in Figure 7-37.

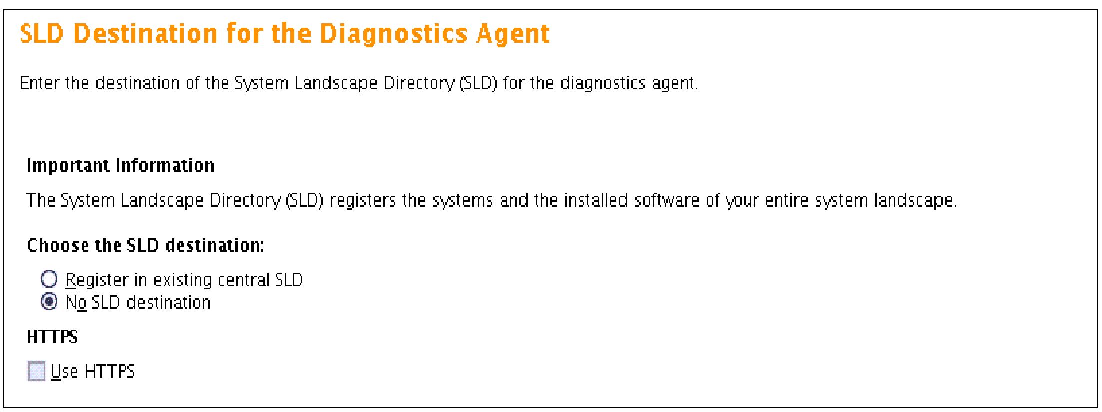


Figure 7-37 DAA installation: SLD destination for the diagnostic agent

Select the archives to unpack, as shown in Figure 7-38.

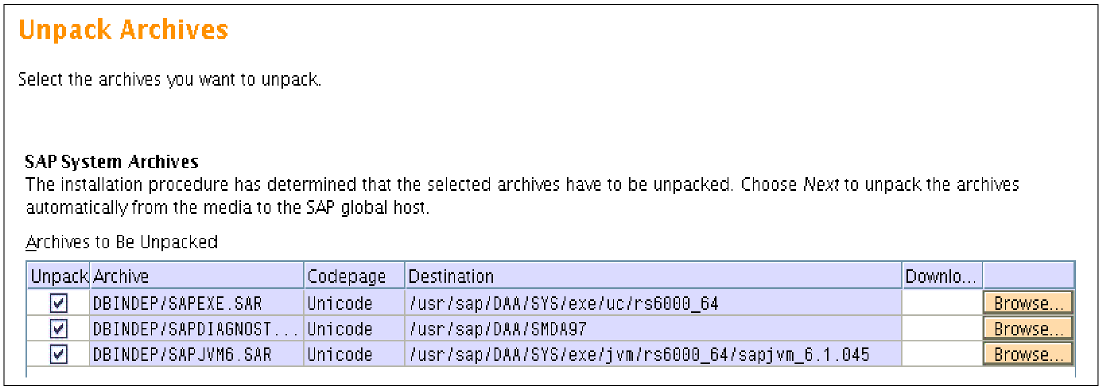


Figure 7-38 DAA installation: Unpack archives

Specify the NWDI landscape integration as shown in Figure 7-39.

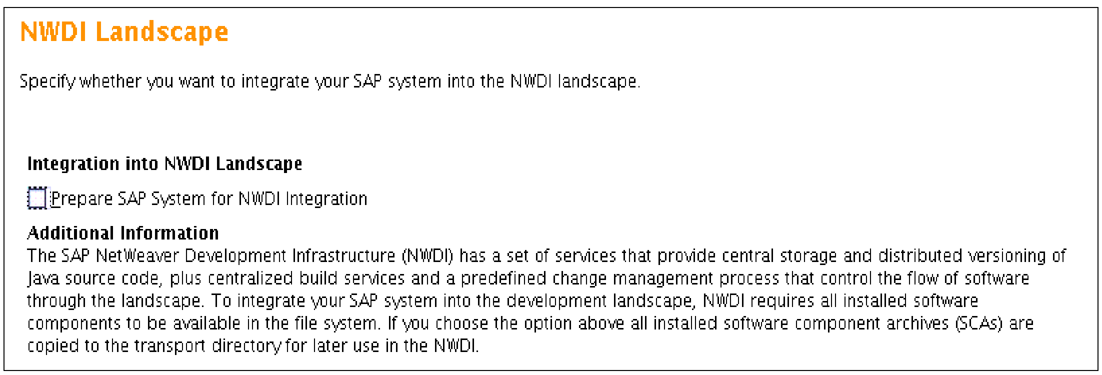


Figure 7-39 DAA installation: NWDI landscape

Finish and run the installation

Verify the parameters and run the installation as shown in Figure 7-40. In case additional AS instances are to be installed, remember that the DAA and host agent are installed once per host, not once per AS instance.

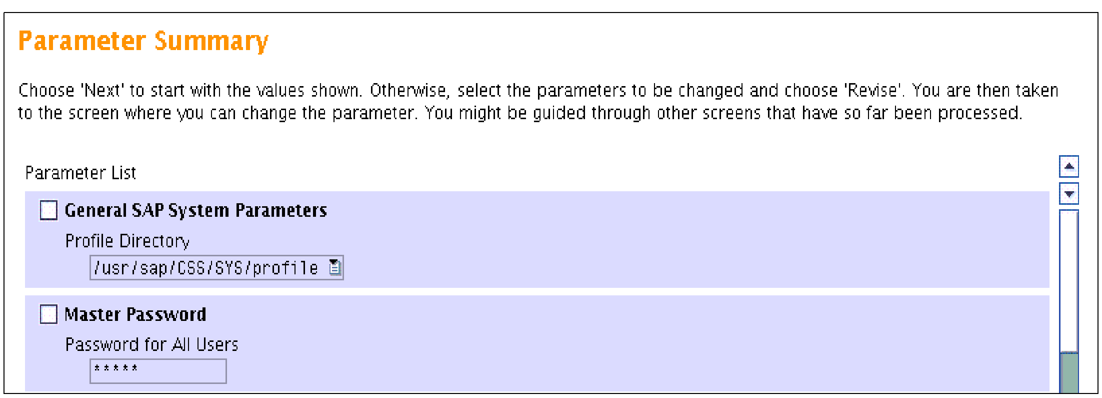


Figure 7-40 DAA installation: Parameter summary

Note: Repeat the same process on a second host or node to meet the required redundancy of the SAP system.

Install the SAPhost agent (optional)

Check that all hosts and nodes are involved with one DAA and that the host agent has been installed.

Use the SWPM to install the agents according to the SAP installation guide as shown in Figure 7-41.

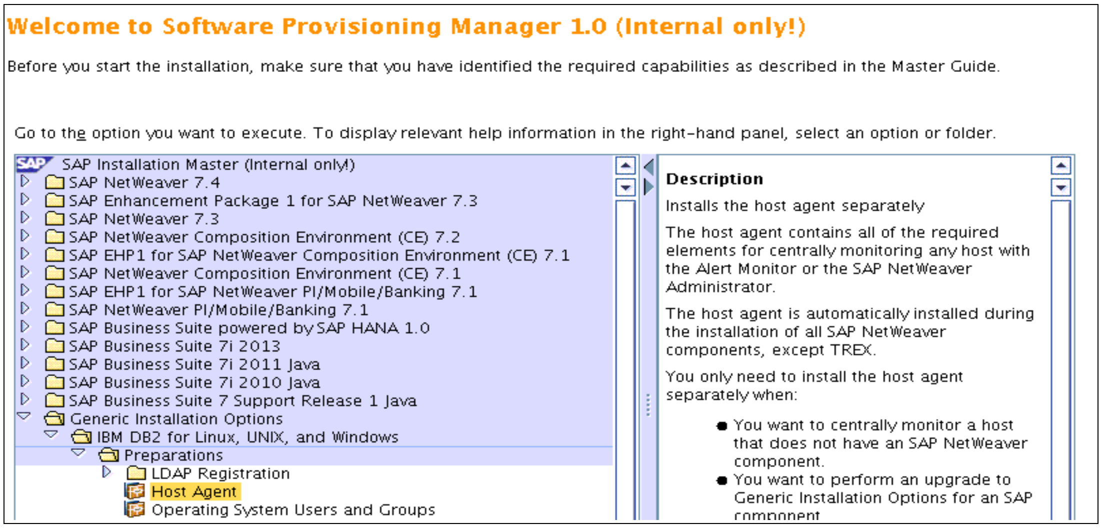


Figure 7-41 Installing the host agent SWPM option

The DAA instance installation option automatically pops up when using the SWPM on your first AS installation.

No special actions are required for PowerHA. This is a pure SAP prerequisite, outside of any cluster control.

Create users and groups with the same IDs (optional)

In case there are nodes that do not have all the SAP users defined, the SWPM provides a menu to create all required users. Use this tool to ensure the consistency of names and IDs on all hosts and nodes.

If a two-tier installation is performed, the database users are also required.

Run **sapinst** and choose the options as shown in Figure 7-42.

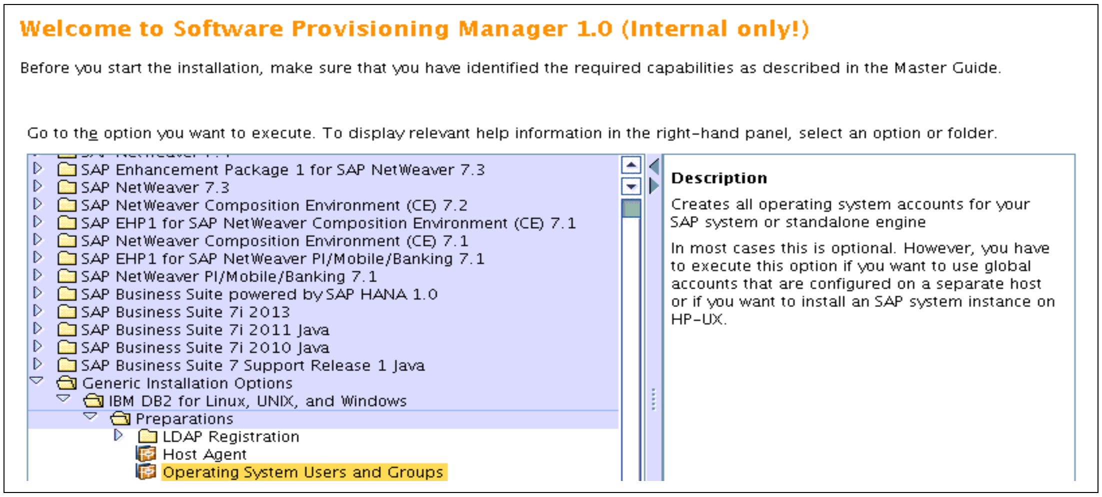


Figure 7-42 SWPM: SAP OS user creation, initial screen

Select all OS users by *not* checking the selection box, as shown in Figure 7-43.

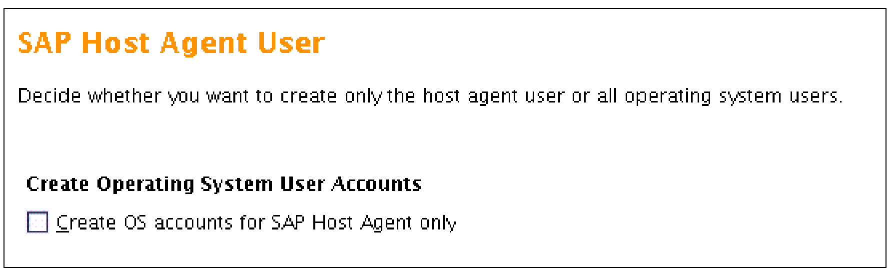


Figure 7-43 SWPM: SAP host agent user

Enter the SAP system ID and database parameters, as shown in Figure 7-44.

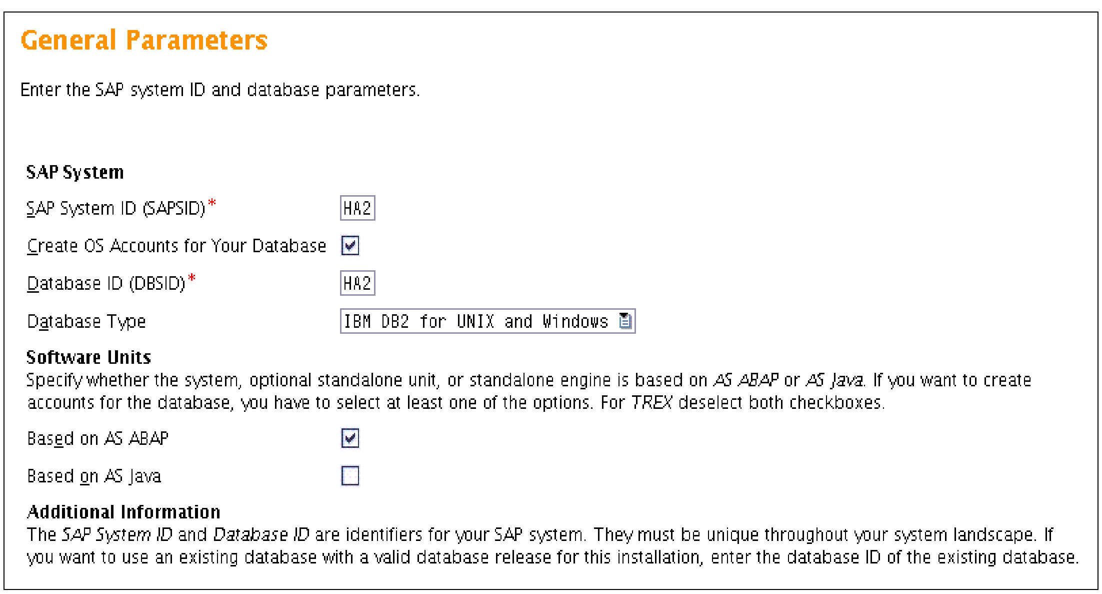


Figure 7-44 SWPM: General parameters

Enter OS users, as shown in Figure 7-45.

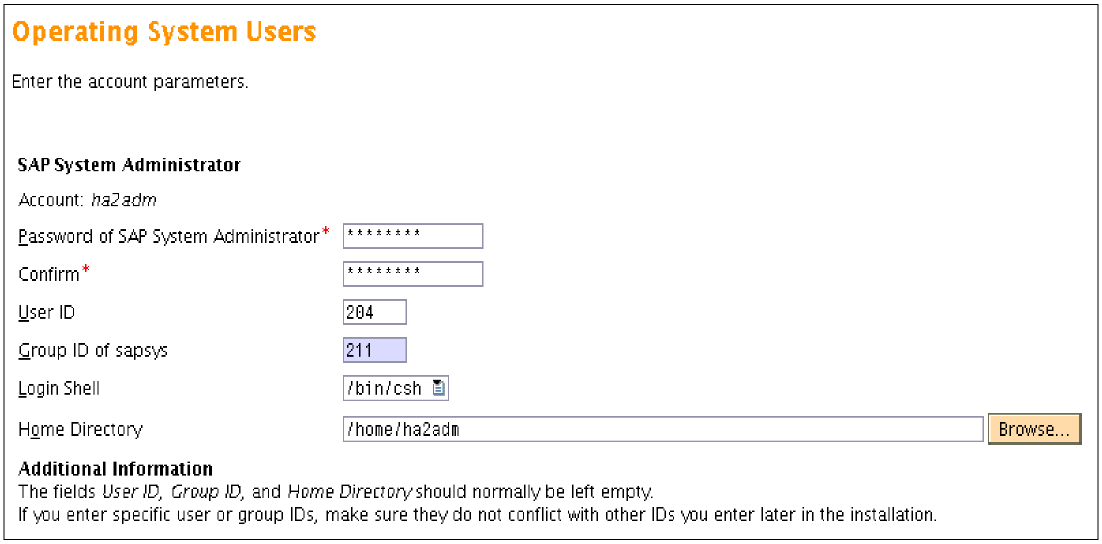


Figure 7-45 SWPM: Operating system users

Now finalize the process and verify the users and groups created.

Note: This does not apply for the daaadm user, because the daa is per node. But it is a recommended approach to have identical IDs.

7.5 Smart Assist for SAP automation

After the preparation of the cluster nodes and the SAP installation, you can run Smart Assist for SAP.

7.5.1 Prerequisites

The logic of Smart Assist requires SAP to comply with the following prerequisites (these prerequisites might change in future releases):

- The SAP file, `kill.sap`, located inside the instance working directory, is built with NetWeaver 7.30. This file must contain only the sapstart PID and no other PIDs.
- Verify the SAP copy function. The *sapcpe* utility is configured to let the monitor and stop scripts inside PowerHA act independently from NFS for all instances. Otherwise, set up *sapcpe* as described in 7.7, “Additional preferred practices” on page 207.
- The root user must have a PATH defined and permissions set to be able to execute **`cleanipc`** for each NetWeaver instance under the cluster control. In the scripts, **`cleanipc`** is invoked as this example shows:

```
eval "cd ${EXE_DIR}; ./cleanipc ${INSTANCE_NO} remove"
```

7.5.2 Prepare

This section describes the steps to prepare the environment before using Smart Assist for SAP.

Update /etc/services

Update the /etc/services. Keep these stipulations in mind:

- Conflicts must be resolved manually.
- All SAP-related entries must match on all nodes and external hosts.

Merge /usr/sap/sapservices on both nodes

Merge the /user/sap/sapservices on both nodes. Check that all of these conditions are met:

- The file is consistent on all applicable nodes.
- All instances to be clustered are listed.
- The attributes listed in this file (/usr/sap/sapservices) match the **ps -ef** command output (otherwise, change the file).

Stop the SAP environment on the cluster nodes

Stop SAP and the cluster but keep /sapmnt mounted on all nodes by following these steps:

1. Stop SAP:

```
#su - daaadm
as0007lx:daaadm 1> sapcontrol -nr <no> -function Stop
as0007lx:daaadm 1> sapcontrol -nr <no> -function StopService
#su - ha1adm
sapcontrol -nr <no> -function Stop
sapcontrol -nr <no> -function StopService
```

> **Note:** Repeat this for all available instance numbers on the cluster nodes.

2. For NFS in the cluster:
   Stop all RGs except the NFS RG, and unmanage the RGs.
3. For external NFS:
   No additional steps are required.
4. Stop the cluster:

```
root> clstop -g
```

Copy local directories in /usr/sap/<SID> to the other nodes

Because the SAP installation was performed on a dedicated node, the structures must be replicated to the other nodes as follows:

1. Unmount the shared directories before performing this action.
2. Ensure that the /usr/sap/sapservices file is consistent on all nodes before performing this task.
3. Ensure that /sapmnt is mounted. Otherwise, the logical links are not transferred.
4. Run the following command on node B:

```
scp -pr <nodeA>:/usr/sap/<SID> /usr/sap/
```

5. Start the cluster.

> **Note:** The DAA instance belongs to the application server and, in this case, is not copied.

Install the SAP components on node B (optional)

In case verification shows that there is no redundant AS server running on a different node or external host, perform the following actions to be compliant with the SAP HA installation requirements:

```
nodeA> su - <sid>adm
startsap all
```

Start the SWPM installer

On node B, call …/`sapinst SAPINST_USE_HOSTNAME=<vip of AS>`. The initial installation process is shown in Figure 7-46.

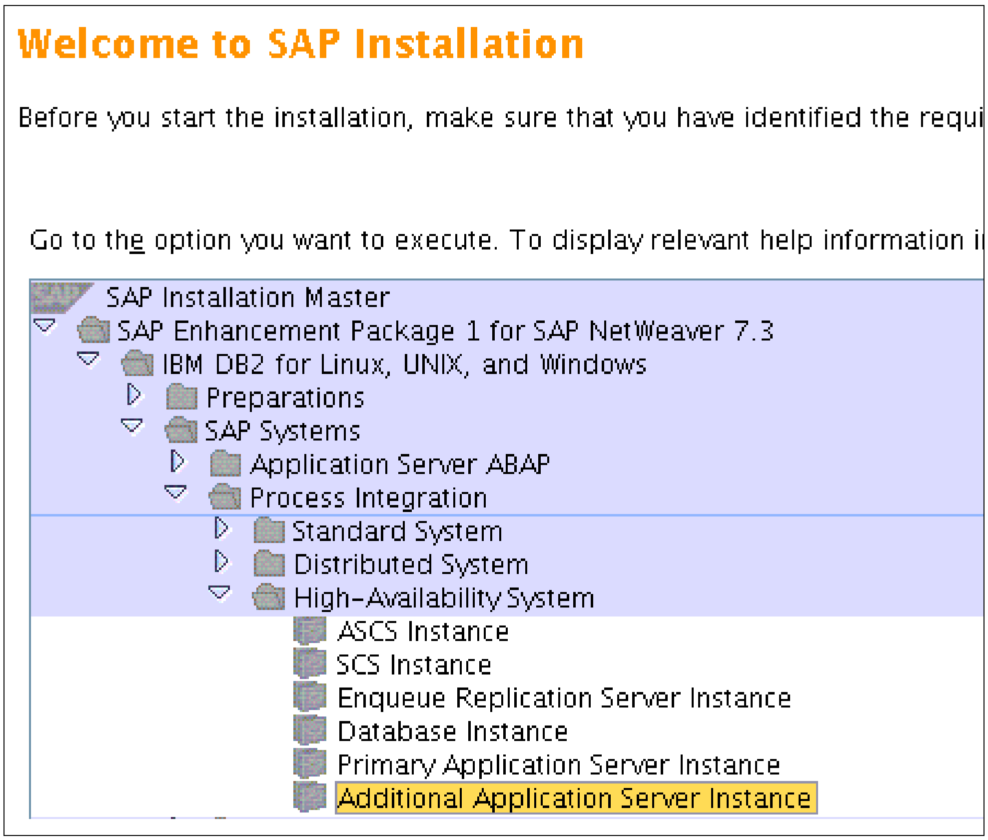


Figure 7-46 SWPM: Additional AS installation menu

Provide basic configuration data for the SAP system and users

Begin by entering the profile directory, as shown in Figure 7-47.

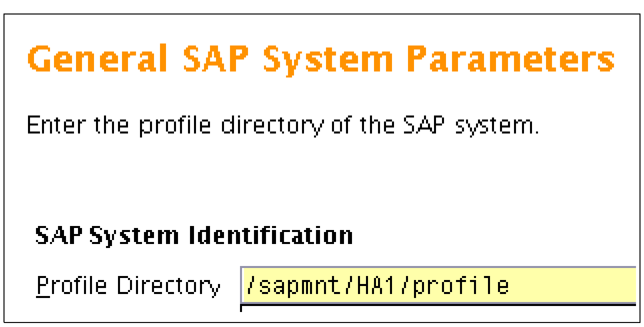


Figure 7-47 Additional AS installation: General SAP system parameters

Specify the media location, as shown in Figure 7-48.

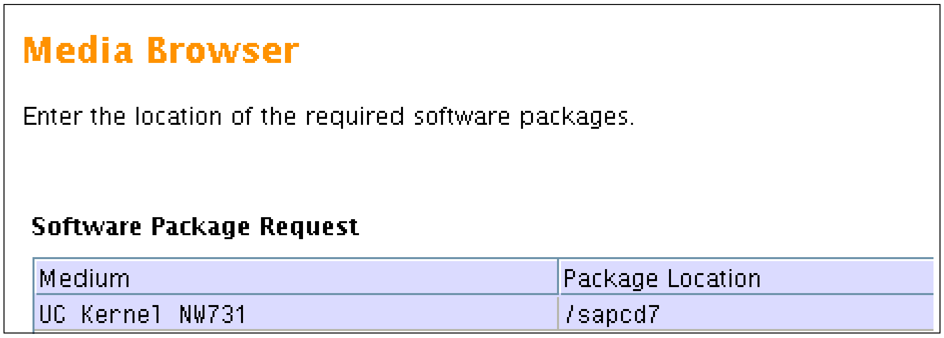


Figure 7-48 Additional AS installation: Media browser

Enter the master password, as shown in Figure 7-49.

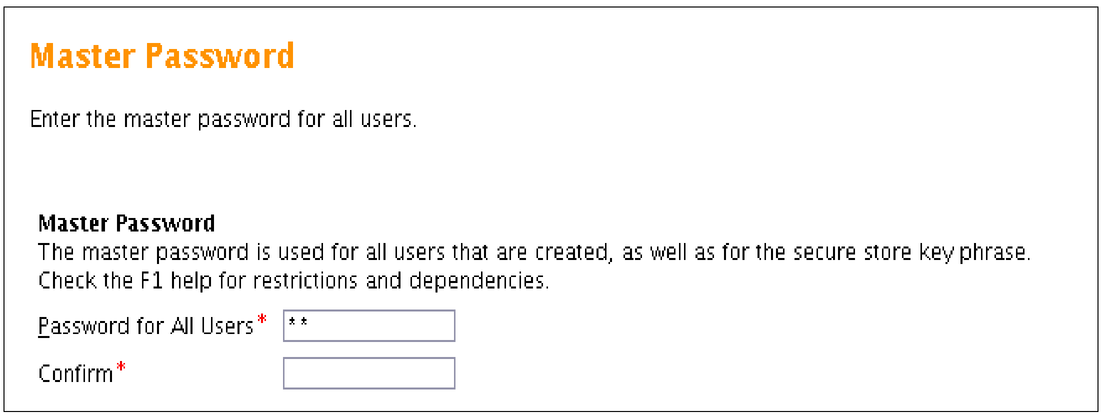


Figure 7-49 Additional AS installation: Master password

Specify the database connectivity

Enter the database ID and host, as shown in Figure 7-50.

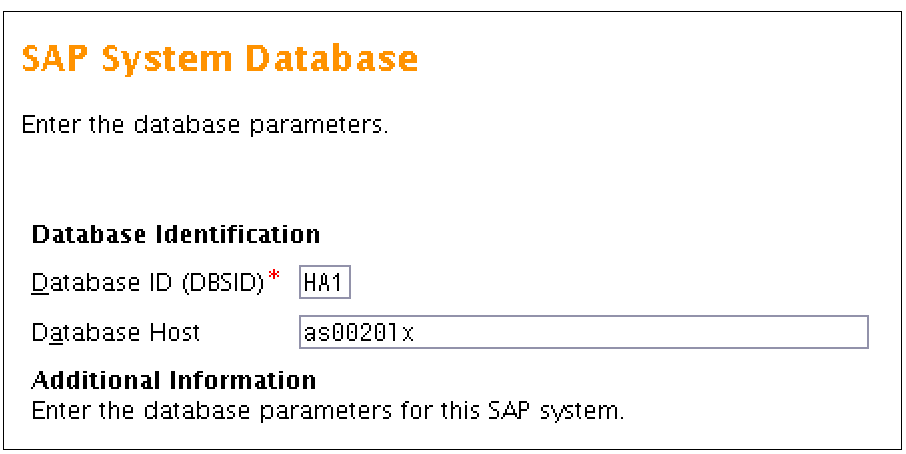


Figure 7-50 Additional AS installation: SAP system database

Enter the ABAP database connect user ID, as shown in Figure 7-51.

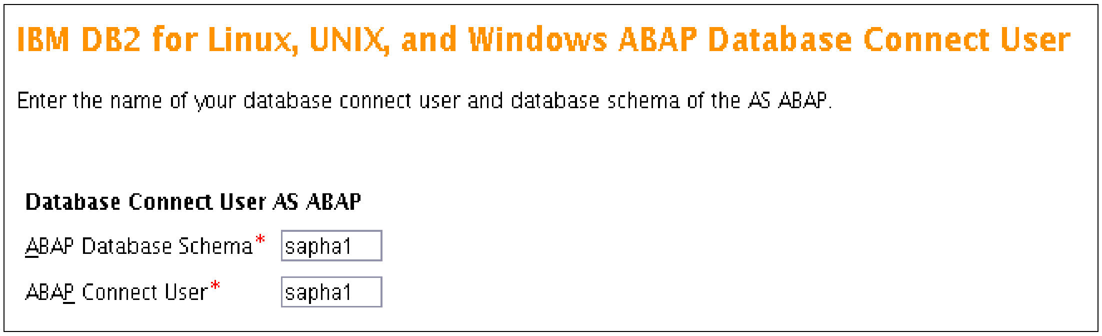


Figure 7-51 Additional AS installation: ABAP database connect user

Enter the Java database connect user ID, as shown in Figure 7-52.

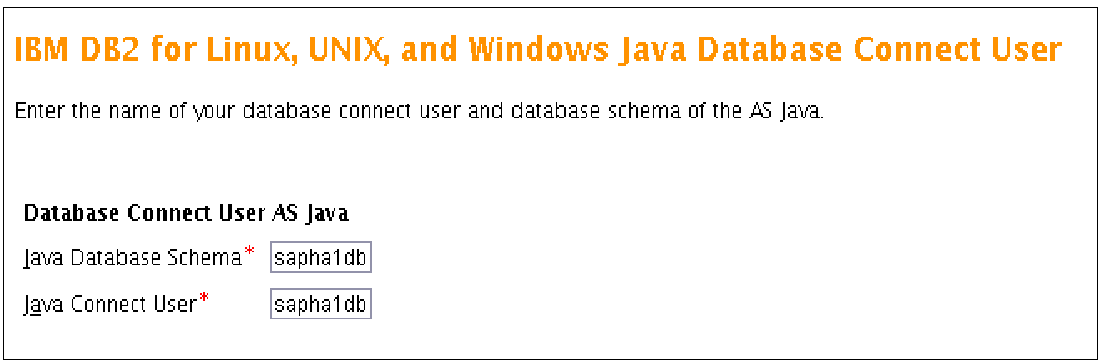


Figure 7-52 Additional AS installation: Java database connect user

Specify the CLI/JDBC media location, as shown in Figure 7-53.

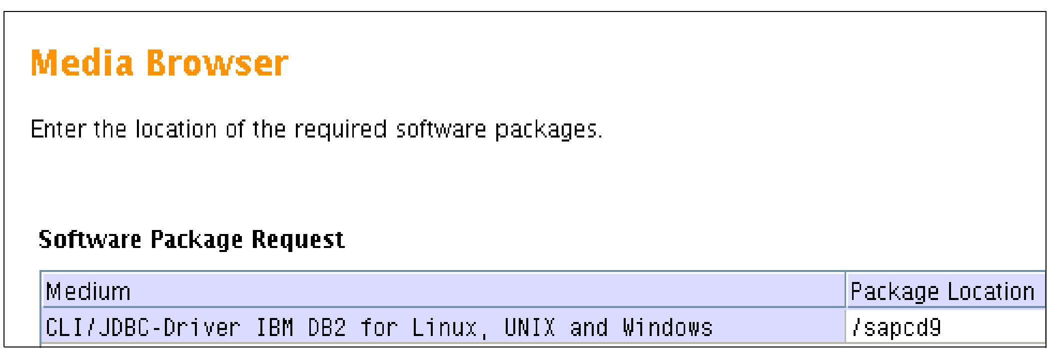


Figure 7-53 Additional AS installation: Media browser

Create the AS instance identifier

Review the AS installation parameters, as shown in Figure 7-54.

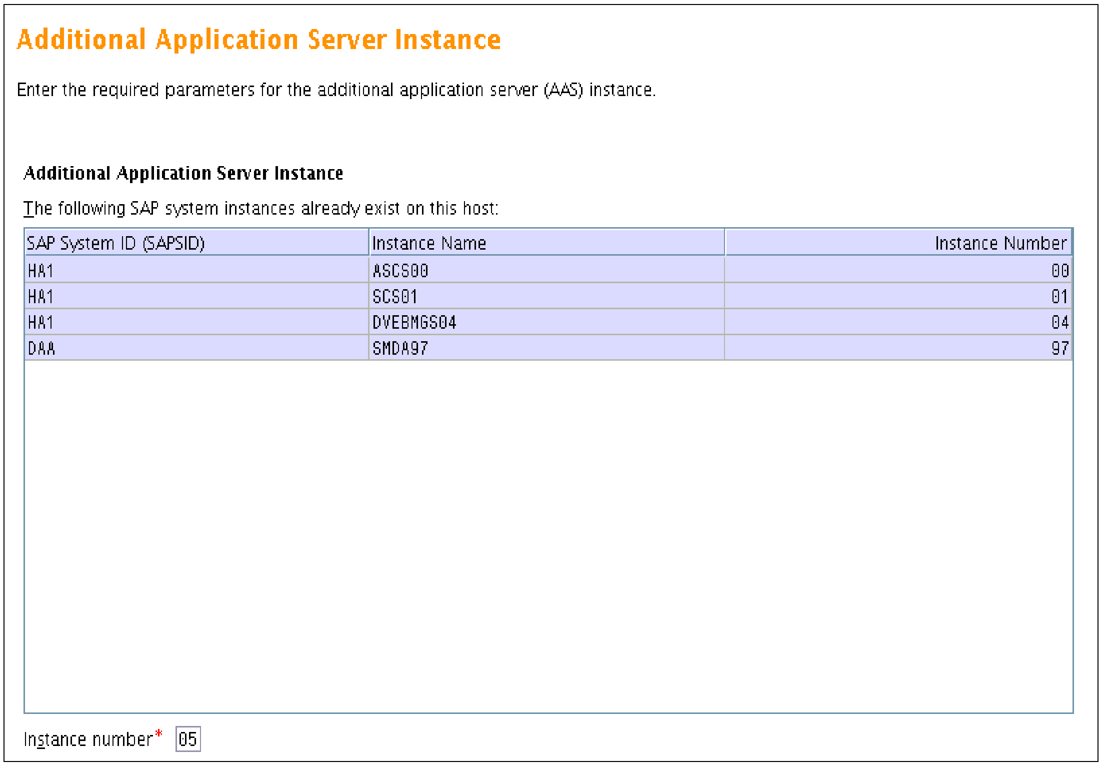


Figure 7-54 Additional AS installation: Review parameters

> **Note:** The additional application server instance (see Figure 7-54) is not aware of the other dialog instance installed on node A. If you plan to use SAP LVM or cluster application servers, ensure that the instance number differs from the remote instance.

Specify memory requirements, as shown in Figure 7-55.

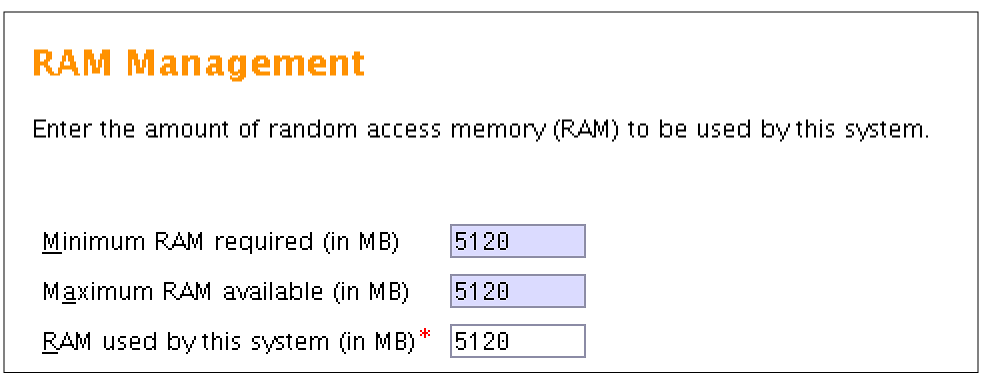


Figure 7-55 Additional AS installation: RAM management

Select archives to unpack, as shown in Figure 7-56.

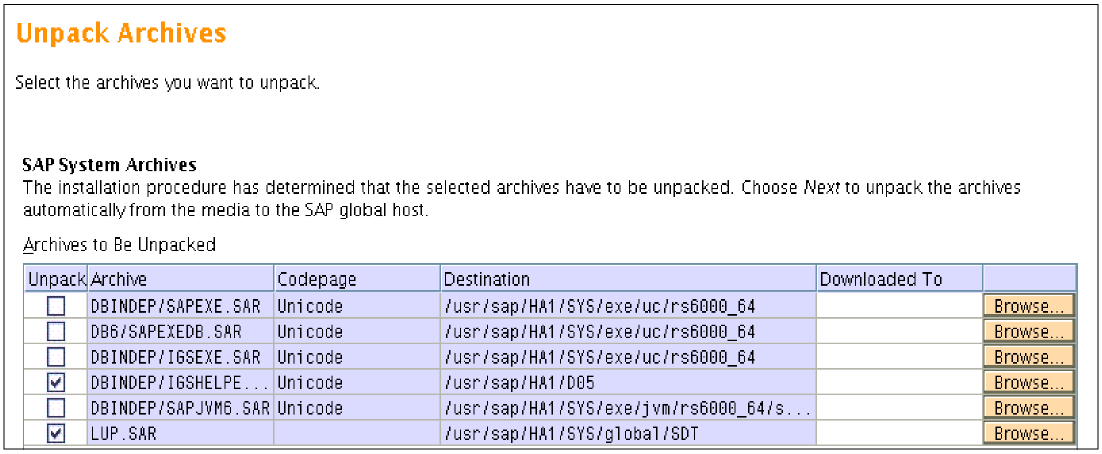


Figure 7-56 Additional AS installation: Unpack archives

Typically, use items that are preselected even if they differ from what Figure 7-56 shows.

Define the SAP Diagnostic Agent (DAA)

This must be performed in case there was no previously installed DAA on this node. Specify the DAA ID, as shown in Figure 7-57.

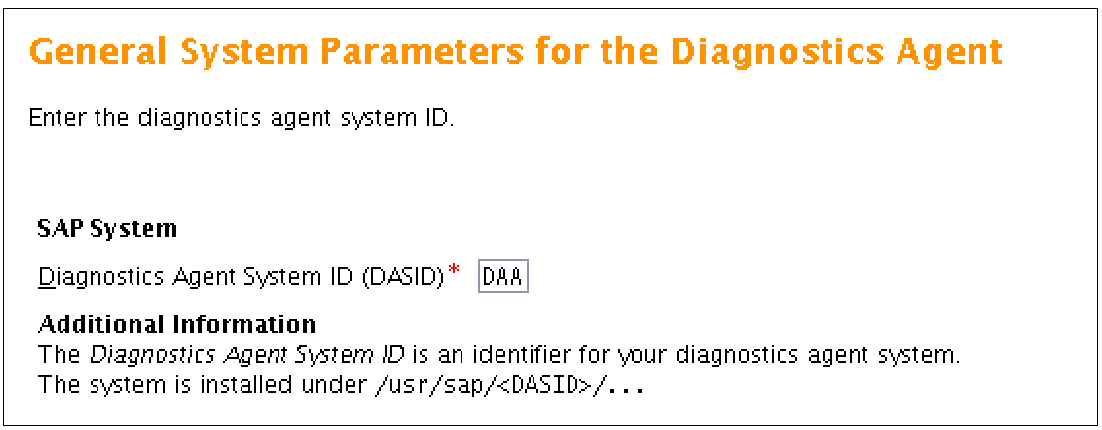


Figure 7-57 DAA installation: General system parameters

Define a unique instance number, as shown in Figure 7-58.

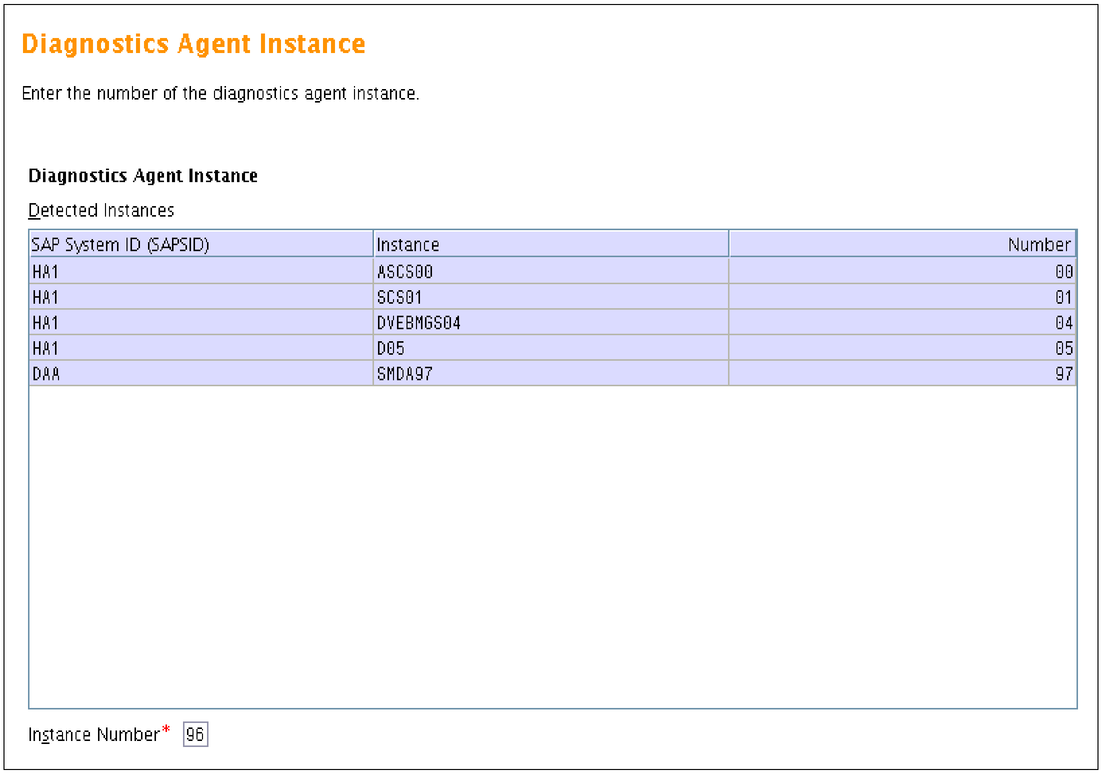


Figure 7-58 DAA installation: Diagnostic agent instance

Specify the DAA destination, as shown in Figure 7-59.

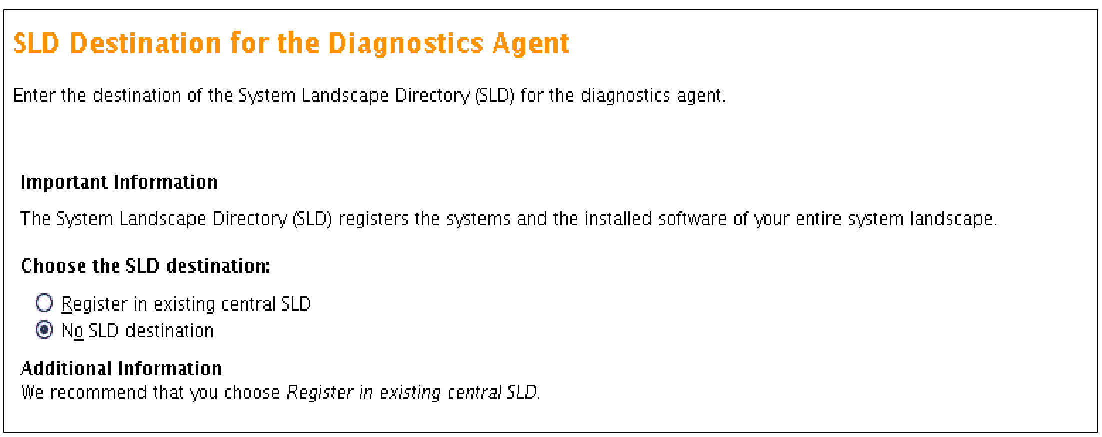


Figure 7-59 DAA installation: SLD destination for Diagnostic Agent

Select archives to unpack, as shown in Figure 7-60.

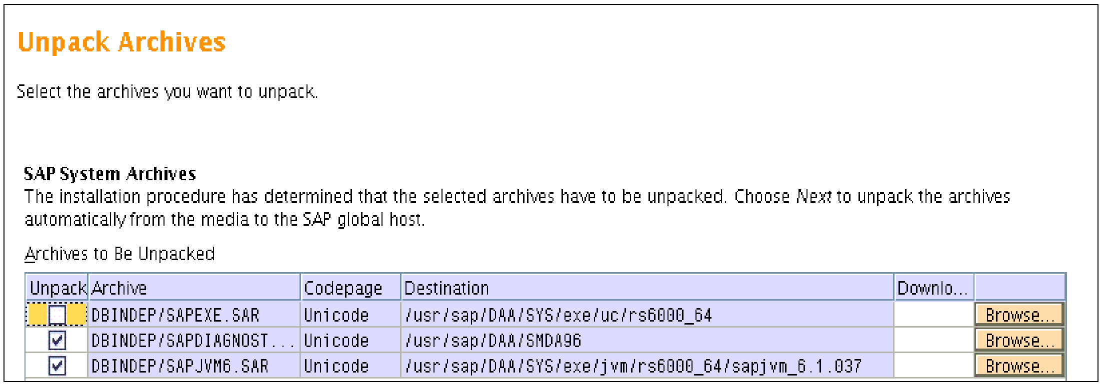


Figure 7-60 DAA installation: Unpack archives

Start the **sapinst** installation process. The *hostagend* should be installed automatically along with the instance.

Install SAP licenses on node A

Please read SAP Note 181543 before requesting the license. Then, install the SAP licenses:

```
<sid>adm 6> vi license.txt (copy the license text into this text file)
<sid>adm 7> saplicense -pinstall ifile=license.txt
```

Change the instance profiles of the AS, ASCS, SCS, and ERS

The SAP stack requires that you actively enable the instances for the enqueue replication facility. This involves all three instance types. Also keep these factors in mind:

- The ERS must be defined with sufficient resources to execute properly and keep its resources available.
- The SCS instance must know where to replicate the state to.
- The AS instances must know that, in case of an outage, they can shortly reconnect and should wait active.

To make the instance profile changes effective, an SAP restart is required after the changes. The database can stay online. Any external AS instances must go through the same procedure as the clustered AS instances.

For updates, see the “Setting Up the Replication Server” page on the SAP.com website:

http://bit.ly/1vbLQOO

Change AS instance profiles

For all enqueue clients on all nodes (the application servers), ensure that the parameters shown in Example 7-18 are set in the instance profiles.

Example 7-18 Changing AS instance profiles

```
enque/process_location = REMOTESA
enque/serverhost = <virtual host name of the enqueue server>
enque/serverinst = <instance number of the enqueue server>
enque/deque_wait_answer = TRUE
enque/con_timeout = 5000 #default value, might require change
enque/con_retries = 60 #default value, might require change
```

Change the Central Service (CS) instance profiles

For the Central Service instance holding the enqueue, the minimum changes that are typically required are shown in Example 7-19.

Example 7-19 Changing SCS instance profiles

```
#-----------------------------------------------------------------------
# Start SAP message server
#-----------------------------------------------------------------------
_MS = ms.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_02 = local rm -f $(_MS)
Execute_03 = local ln -s -f $(DIR_EXECUTABLE)/msg_server$(FT_EXE) $(_MS)
Restart_Program_00 = local $(_MS) pf=$(_PF)
#-----------------------------------------------------------------------
# Start SAP enqueue server
#-----------------------------------------------------------------------
_EN = en.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_04 = local rm -f $(_EN)
Execute_05 = local ln -s -f $(DIR_EXECUTABLE)/enserver$(FT_EXE) $(_EN)
Start_Program_01 = local $(_EN) pf=$(_PF)
#-----------------------------------------------------------------------
# SAP Message Server parameters are set in the DEFAULT.PFL
#-----------------------------------------------------------------------
ms/standalone = 1
ms/server_port_0 = PROT=HTTP,PORT=81$$
#-----------------------------------------------------------------------
# SAP Enqueue Server
#-----------------------------------------------------------------------
enque/table_size = 64000
enque/snapshot_pck_ids = 1600
enque/server/max_query_requests = 5000
enque/server/max_requests = 5000
enque/async_req_max = 5000
enque/encni/threadcount = 4
rdisp/enqname = $(rdisp/myname)
enque/server/replication = true
```

Change the ERS instance profiles

For the ERS instance, the profile settings shown in Example 7-20 are required.

Example 7-20 Changing the ERS instance profiles

```
#-----------------------------------------------------------------------
# Settings for enqueue monitoring tools (enqt, ensmon)
#-----------------------------------------------------------------------
enque/process_location = REMOTESA
rdisp/enqname = $(rdisp/myname)
#-----------------------------------------------------------------------
# standalone enqueue details from (A)SCS instance
#-----------------------------------------------------------------------
SCSID = <instance number of (A)SCS>
SCSHOST = <service IP alias>
enque/serverinst = $(SCSID)
enque/serverhost = $(SCSHOST)
# NOTE: you have to delete these lines. Set them to zero is not sufficient!
```

```
#Autostart = 1
#enque/enrep/hafunc_implementation = script
#-----------------------------------------------------------------------
# Start enqueue replication server
#-----------------------------------------------------------------------
_ER = er.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_02 = local rm -f $(_ER)
Execute_03 = local ln -s -f $(DIR_EXECUTABLE)/enrepserver$(FT_EXE) $(_ER)
Start_Program_00 = local $(_ER) pf=$(_PFL) NR=$(SCSID)
```

Note: PowerHA 7.1.3 does not support ERS polling as part of Smart Assist for SAP. The SAP ERS enablement options are discussed in 7.1.1, “SAP NetWeaver design and requirements for clusters” on page 132.

Start the SAP system and verify that it is executing properly

This allows the SAP Smart Assist to discover the instances per node.

The *sapstartsrv* processes should be running for the instances to be clustered:

```
# su - <sid>adm
# sapcontrol -nr <no> -function StartService <SID>
```

Example 7-21 shows the SCS and ERS sapstartsrv processes from the referenced installation.

Example 7-21 SCS and ERS sapstartsrv processes

```
#ps -fu ha1adm
     UID      PID     PPID   C    STIME    TTY  TIME CMD
ha1adm 12058694  1   0    15:42:55      -    0:00
/usr/sap/HA1/ERS03/exe/sapstartsrv pf=/usr/sap/HA1/SYS/profile/HA1_ERS03_as0013lx
—D
ha1adm 12583040  1   0    15:43:44      -    0:00
/usr/sap/HA1/ASCS00/exe/sapstartsrv
pf=/usr/sap/HA1/SYS/profile/HA1_ASCS00_as0010lx -D
ha1adm 17367052  1   0    15:43:20      -    0:00
/usr/sap/HA1/SCS01/exe/sapstartsrv
pf=/usr/sap/HA1/SYS/profile/HA1_SCS01_as0012lx -D
ha1adm 32506016  1   5    15:53:42      -    0:00
/usr/sap/HA1/ERS02/exe/sapstartsrv
pf=/usr/sap/HA1/ERS02/profile/HA1_ERS02_as0011lx -D
```

The Smart Assist discovery tool can be executed by the root user:

```
# /usr/es/sbin/cluster/sa/sap/sbin/cl_sapdiscover -t [GFS/AS/SCS/ERS/DB]
```

The discovery tool returns 0 if no instance can be found and 1 if one or more matching instances are found. See Example 7-22 on page 194.

Example 7-22 Smart Assist discovery tool output

```
## OUTPUT:
## -t GFS
## SAP Smart Assist:SAPNW_7.0:SAP NetWeaver Global
filesystem:SAPNW_7.0_SAPGFS:{0|1}
## -t SCS
## SAP Smart Assist:SAPNW_7.0:SAP NetWeaver (A)SCS
Instance:SAPNW_7.0_SCSINSTANCE:{0|1}
## -t ERS
## SAP Smart Assist:SAPNW_7.0:SAP NetWeaver ERS
Instance:SAPNW_7.0_ERSINSTANCE:{0|1}
## -t AS
## SAP Smart Assist:SAPNW_7.0:SAP NetWeaver AS Instance:SAPNW_7.0_ASINSTANCE:{0|1}
## -t DB
## SAP Smart Assist:SAPNW_7.0:SAP Database Instance:SAPNW_7.0_DBINSTANCE:{0|1}
##
## RETURNS:
##      0 on success
##      1 on failure
```

> **Note:** The term *SAPNW_7.0_** in the output of Example 7-22 is a legacy naming convention in Smart Assist. It supports NetWeaver versions above 7.0.

In case the instances cannot be discovered on the node where sapstartsrv is running, troubleshooting can be done as follows:

```
# cd /usr/es/sbin/cluster/sa/sap/sbin
# export VERBOSE_LOGGING="high"
# ./cl_sapdiscover -t [GFS/AS/SCS/ERS/DB]
```

Resolve all errors until all desired instances can be discovered.

Bring the PowerHA cluster software to the INIT state on both nodes

Ensure that the sapstartsrv processes are running, but not the instances. Then, stop the cluster with the **unmanage all RGs** option to bring the cluster into ST_INIT state.

The **`mount`** command still shows that all SAP file systems are mounted, including the SAP global file system.

7.5.3 Run Smart Assist for SAP

In general, Smart Assist agents have three phases:

Phase one: DISCOVERY
: The discovery is executed on the configured cluster nodes and identifies applications that can be clustered by the selected Smart Assist.

Phase two: VERIFICATION
: The verification verifies some but not all prerequisites to ensure a save addition.

Phase three: ADDITION
: The addition adds the RG to the cluster.

The SAP stack is sensitive to the acquisition order. Therefore, it is mandatory to add the components in the following order:

1. NFS, if part of the cluster
2. Database instance, if part of the cluster
3. CS instances, followed by the corresponding ERS instance
4. Application server instances

Follow these steps to run Smart Assist:

1. Start the Smart Assist menu and select **SAP Smart Assist** from the options:

   # smitty clsa → **Add an Application to the PowerHA SystemMirror Configuration**

2. In the next panel, select **Automatic Discovery and Configuration**.
3. As the final selection, mark the **SAP NetWeaver (A)SCS Instance** as shown in Figure 7-61. The SMIT panel might look different, depending on which instances or databases are discoverable from an SAP point of view.

```
Select The Specific Configuration You Wish to Create

  Move cursor to desired item and press Enter.

    SAP NetWeaver AS Instance        # as0003lx
    SAP NetWeaver ERS Instance       # as0003lx
    SAP NetWeaver (A)SCS Instance    # as0004lx

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

Figure 7-61 Select specific SAP configuration

   The ERS, AS, and (A)SCS instances are configured similarly but with different instance names, IP addresses, and VGs. In the following sections, only an ASCS addition is described. The instance type differences are highlighted if there are any.

   > **Note:** To perform the addition of an ERS instance, the SCS resource groups must be put into offline state. This is required as the SCS and ERS instances have dependencies which must be configured in the SCS resource group and its ODM. This is only possible when it is offline.

   Command line example to bring an RG offline:

   ```
   /usr/es/sbin/cluster/utilities/clRGmove -s 'false'  -d -i -g 'SAP_HA1_SCS01_RG'
   -n 'as0007lx'
   ```

4. In case of a dual stack implementation, both ASCS and SCS must be configured. For this scenario, ASCS00 was chosen, as shown in Example 7-23 on page 196.

Example 7-23 Selecting a SCS/ASCS instance menu

```
+--------------------------------------------------------------------------+
|                        Select a SCS/ASCS instance                        |
|                                                                          |
| Move cursor to desired item and press Enter.                             |
|                                                                          |
|   ASCS00                                                                 |
|   SCS01                                                                  |
|                                                                          |
| F1=Help                  F2=Refresh              F3=Cancel               |
| F8=Image                 F10=Exit                Enter=Do                |
| /=Find                   n=Find Next                                     |
+--------------------------------------------------------------------------+
```

5. Finalize the configuration parameters. For the following example, we use the SMIT panel as shown in Figure 7-62.

```
 Add SAP SCS/ASCS Instance(s) Details
                                                    [Entry Fields]
* SAP SYSTEM ID                                     HA1
* SAP SCS/ASCS Instance(s) Name(s)                  ASCS00
* Application Name                                  [SAP_HA1_ASCS00]
* Primary Node                                      as0007lx
* Takeover Nodes                                    [as0008lx]

* Service IP Label                                  as0010lx
* Network Name                                     [net_ether_01]

* Volume Group(s)                                  [vgascscss]

  DataBase Resource Group                          [ ]
```

Figure 7-62 Add SAP SCS/ASCS instance

Figure 7-62 field explanation:

- SAP SYSTEM ID: HA1

  Discovered fix value.

- SAP SCS/ASCS Instance(s) Name(s): ASCS00

  Discovered fix value.

- Application Name: [SAP_HA1_ASCS00]

  Recommended to keep standard naming conventions. But can be changed.

  ```
  Primary Node: as0007lx
  Takeover Nodes: [as0008lx]
  ```

  In case all instances have been started on the same node (including SCS and ERS), the Primary Node field is the node where the ERS sapstartsrv is currently running. Not the runtime primary node which must be different from the corresponding (A)SCS instance. Remember to revert the ERS node priorities as an additional step after configuring the initial cluster if it is not the reverse order of the SCS instance.

- Service IP Label: as0010lx

  Discovered fix value.

- Network Name: [net_ether_01]

  For SCS and ERS instances, this must be a network hosting the service IP label. For application server instances, this value can be set to `LOCAL` in case the IP is node bound.
- Volume Group(s): [vgascscss]

  If the instance is installed on a file system bound to a node, be sure to choose **LOCAL** as the value rather than the VG name.
- Database Resource Group:

  Entering the Database Resource Group creates a resource group dependency between the database and the SAP instance.

  Leave this field blank for SCS instances. It is introduced only for backward compatibility to SAP kernels earlier than 7.20, where an ASCS could not be started without the database being up and running. This field does not show up for ERS instances.

  An empty value is also the default for AS instances, but some specific consideration in your environment might lead to configuring the resource group dependency to the database, depending on the overall landscape architecture.

Repeat all of these steps for each SAP instance that is part of the cluster. Run a PowerHA synchronization and verification thereafter.

Example 7-24 shows a command-line example for the addition.

Example 7-24 Command-line example for the addition

```
/usr/es/sbin/cluster/sa/sap/sbin/cl_addsapinstance -t SCS -s'HA1' -i'SCS01'
-a'SAP_HA1_SCS01' -p'as0007lx' -T'as0008lx' -I'as0012lx' -n'net_ether_01'
-V'vgscscss'
```

7.5.4 Post process

1. Ensure that the changes performed to the instance profiles as described in this section are reflected in all locations. For example, the ERS has two physical locations for instance profiles. One is `/sapmnt/SID/profile` and the other is `/usr/sap/SID/ERSxx/profile`. If they are not synchronized through sapcpe, this must be fixed manually to *always* be updated on start of each instance. Another location is `/usr/sap/<SID>/SYS/profile/` which can potentially contain physical copies instead of links.
2. Ensure that the primary node of the ERS is different from the primary node of its SCS instance. This is not the case if the ERS was discovered on the same node as the SCS previously. Use the “Change/Show an Application’s PowerHA SystemMirror Configuration” SMIT Smart Assist submenu to change the setting after the addition of the ERS instances:
   a. Select **Change/Show an Application’s PowerHA SystemMirror Configuration,** and then select the ERS resource groups.
   b. Specify the primary node to the standby node of the corresponding SCS instance and adjust the takeover node list.
   c. Run a synchronization and verification.

   Figure 7-63 on page 198 shows sample output after the change is complete.

```
 Change/Show SAP ERS Instance(s) Details
Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                        [Entry Fields]
* SAP SYSTEM ID                                       HA2
* SAP ERS Instance(s) Name(s)                         ERS22
* Application Name                                   [SAP_HA2_ERS02]
* Primary Node                                       [as0008lx]                 +
* Takeover Nodes                                     [as0007lx]                 +
* Service IP Label                                   [as0011lx]                 +
* Network Name                                       [net_ether_01]             +
* Volume Group(s)                                    [<vg name or 'LOCAL'>]
```

Figure 7-63 Change/Show SAP ERS instance

3. Ensure that the following log file is created and can be written to by root after the Discovery and Addition. To obtain the log file location, use the following command:

   `/usr/es/sbin/cluster/sa/sbin/clquerysaapp -a SAP_GLOBALS_HA1 | grep LOGGER_LOGFILE`

4. Ensure that the following log file is created and can be written to by root and the SAP user <sid>adm after the discovery and addition. To get the location, use the following commands:

   a. Retrieve the SALOGFILEPATH directory:

   ```
   #cat /usr/es/sbin/cluster/sa/sap/etc/SAPGlobals | grep SALOGFILEPATH | grep
   -v KSSLOGFILE

   SALOGFILEPATH=$(/usr/es/sbin/cluster/utilities/clodmget -n -q
   "name=sapsa.log" -f value HACMPlogs)
   ```

   b. Execute the **clodmget** command to obtain the directory.

5. The file is named as defined in `/usr/es/sbin/cluster/sa/sap/etc/SAPGlobals`:

   `OSCON_LOG_FILE=$(echo "$SALOGFILEPATH/sap_powerha_script_connector.log")`

6. Review the following sections of this chapter to finalize the cluster:

   – 7.5.5, “Customize” on page 198
   – 7.6, “OS script connector” on page 206
   – 7.7, “Additional preferred practices” on page 207

7.5.5 Customize

To address IBM clients’ demands, Smart Assist for SAP provides options to change between valid behaviors. In addition, different logging and alerting mechanisms can be chosen.

In the Smart Assist menu (**smitty clsa**), the following highlighted SMIT panels can be used to change behaviors and resources including verification. See Figure 7-64 on page 199.

```
               Make Applications Highly Available (Use Smart Assists)

Move cursor to desired item and press Enter.

Add an Application to the PowerHA SystemMirror Configuration
Change/Show an Application's PowerHA SystemMirror Configuration
Change/Show the Resources Associated with Your Application
Change/Show the SAP Instance Attributes
Remove an Application from the PowerHA SystemMirror Configuration
```

Figure 7-64 SMIT clsa: Make applications highly available (User Smart Assists)

Note: Do not attempt to use any SMIT panels outside of the Smart Assist menus for tasks that can be performed from within those menus. If you do that, you miss the verification processes and you have no certainty that the appropriate changes actually took place.

The SMIT panel shown in Figure 7-65 is taken from a SCS instance called “ASCS00” with a SID HA1. If the values or panels differ for AS or ERS instances, this is highlighted. All other fields are defaults that are valid for CS, ERS, and AS instances.

Change or show an application’s PowerHA SystemMirror configuration

For the following description of changing the SAP Smart Assist details, look at the SMIT panel shown in Figure 7-65 and select these options: **smitty clsa** → **Change/Show an Application’s PowerHA SystemMirror configuration** → **Change/Show SAP SCS/ASCS Instance(s) Details**.

```
 Change/Show SAP SCS/ASCS Instance(s) Details
                                              [Entry Fields]
* SAP SYSTEM ID                               HA1
* SAP SCS/ASCS Instance(s) Name(s)            ASCS00
* Application Name                            SAP_HA1_ASCS00
* Primary Node                                [as0007lx]
* Takeover Nodes                              [as0008lx]

* Service IP Label                            [as0010lx]
* Network Name                                [net_ether_01]

* Volume Group(s)                             [vgascscss]

DataBase Resource Group                       []
```

Figure 7-65 Change/Show SAP SCS/ASCS instance

Figure 7-65 fields explanation:

- SAP SYSTEM ID, SAP SCS/ASCS Instance(s) Name(s) and Application Name:

  Displayed identifiers for the Resource to change.
- Primary Node:
  – For ERS instances: The primary node must be different from the primary node of the SCS instance.

- – For SCS instances: For dual stacks, the SCS instances can have the same or different primary nodes.
- ► Takeover Nodes:
  - – For all instances: Consider the order of takeover nodes when running on a three-node cluster, because that order will be honored in selecting the next priority node.
  - – For ERS instances: The first takeover node of the corresponding SCS instance should be its primary node. Subsequent takeover nodes should be specified in the same order, ending with the primary node of its SCS instance.
- ► Service IP Label:

  The value needs to be set to the virtual IP the SAP instance was installed with.
- ► Network Name:
  - – For SCS and ERS instances: This must be a real network serving the virtual IP.
  - – For application server instances: This value can be set to "LOCAL" in case the virtual IP is node bound.
- ► Volume groups:

  Here, you can create new VGs or change from shared to local volume groups. You can change it to "LOCAL" by using F4.
  - – Database Resource Group:

    Defines a Startafter Resource Group dependency between the Database Resource Group and this resource group. It only works if the database is installed into the same cluster (not the recommended installation option).
  - – Does not exist for ERS instances.

Change or show the resources that are associated with your application

For the following explanation of changing resources associated with the custom resource group, see Figure 7-66 on page 201.

1. Select **smitty clsa** → **Change/Show the Resources Associated with Your Application**.

```
Change/Show All Resources and Attributes for a Custom Resource Group
                                                    [Entry Fields]
  Resource Group Name                               SAP_HA1_ASCS00_RG
  Participating Nodes (Default Node Priority)       as0007lx as0008lx
* Dynamic Node Priority Policy                      [cl_highest_udscript_rc]
  DNP Script path                                   [/usr/es/sbin/cluster/]
  DNP Script timeout value                          [360]
  Startup Policy                                    Online On First Available
  Fallover Policy                                   Fallover Using Dynamic No
  Fallback Policy                                   Never Fallback
  Service IP Labels/Addresses                       [as0010lx]
  Application Controllers                           [HA1_ASCS00_AP]
  Volume Groups                                     [vgascscss ]
  Use forced varyon of volume groups, if necessary  false
  Automatically Import Volume Groups                false
  Filesystems (empty is ALL for VGs specified)      [ ]
  Filesystems Consistency Check                     fsck
  Filesystems Recovery Method                       sequential
  Filesystems mounted before IP configured          false
  ...
  Miscellaneous Data                                [ERS02]
  WPAR Name                                         []
  User Defined Resources                            []
```

Figure 7-66 Change/Show all resources and attributes of a custom resource group

Figure 7-66 field explanation for SAP-related fields:

- Resource Group Name and Participating Nodes (Default Node Priority):

  Displayed values.

- Dynamic Node Priority Policy, DNP Script path, and DNP Script timeout value:

  The three DNP values are displayed only for the SCS instances. They ensure that a SCS instance is always moved to the node where the ERS is actively running, even if it is not the typical takeover node. This setting is active only if more than two nodes are specified. For two-node clusters, leave them as they are, because they will be ignored.

- Startup Policy:

  For All SAP and NFS instances in SAP Landscapes it should be “Online On First Available Node”.

- Fallover Policy:
  - For CS: Fallover Using Dynamic Node Priority.
  - For ERS and AS: Fallover to Next Priority Node in the List.

- Fallback Policy:

  Should always be set to “Never Fallback.”

- Service IP Labels/Addresses:

  Must be set for NFS, ERS, and SCS instances. Can be empty for AS instances.

- Application Controllers:

  Keep the default naming conventions.

– Volume Groups:

  Can be empty if a local disk is used. Otherwise, the VG names are displayed and can be extended.

– Use forced varyon of volume groups, if necessary:

  Default value is *false*.

– Automatically Import Volume Groups:

  Default value is *false*.

– File systems (empty is *all* for the VGs specified):

  If it is not empty, a disk layout issue is probably the cause. Carefully verify the setup.

– File systems Consistency Check:

  Default is fsck.

– File systems Recovery Method:

  Default is “sequential.”

– File systems mounted before IP configured:

  Default for CS, ERS, and AS is *false*.

– […NFS etc related settings]

  Not relevant for SAP Smart Assist, so not further described here.

– Miscellaneous Data:

  - For CS: [ERS02]
    The SAP instance name of the corresponding ERS instance name as indicated in the SAP SCS instance profile.
  - For ERS: [ASCS00,10.17.184.190]
    The SAP instance name of the corresponding SCS instance name as indicated in the SAP ERS instance profile and the SCS IP.
  - For AS: [empty]

2. Select **smitty clsa** → **Change/Show the SAP Instance Attributes**.

   For the following discussion about changing attributes, we use the System Management Interface Tool (SMIT) panel shown in Figure 7-67 on page 203.

> **Note:** At the time of writing this chapter, the SMIT panels were being updated, so they might look slightly different from the SMIT panel shown in Figure 7-67.

```
 Change/Show SAP SCS/ASCS Instance(s) attributes Details

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

[TOP]                                                    [Entry Fields]
* SAP SYSTEM ID                                        HA1
* SAP SCS/ASCS Instance(s) Name(s)                     ASCS00
* Application Name                                     SAP_HA1_ASCS00
  SAP Instance Profile                                [/usr/sap/HA1/SYS/prof>
  SAP Instance Executable Directory                   [/usr/sap/HA1/ASCS00/e>
  Sapcontrol WaitforStarted Timeout                   [60]                     #
  Sapcontrol WaitforStarted Timeout Delay             [0]                      #
  Sapcontrol WaitforStopped Timeout                   [10]                     #
  Sapcontrol WaitforStopped Timeout Delay             [0]                      #
* SAP SYSTEM ID                                        HA1
  ENSA/ERS Sync time                                  [10]                     #
  Is ERS Enabled                                      [1]                     +
  Notification Level                                  [0]                     +
SA SAP XPLATFORM LOGGING                            [0]                     +
  EXIT CODE START sapcontrol Start failed             [1]                     +
  EXIT CODE START sapcontrol StartService failed      [1]                     +
  EXIT_CODE_START_sapcontrol_NFS_failed               [1]                     +
  EXIT_CODE_MONITOR_sapstartsrv_unavailable           [1]                     +
  EXIT_CODE_MONITOR_failover_on_gw_outage             [0]                     +

Is this an NFS mountpoint?                          [1]                     +
  SAPMNT Export Directory                             [/export/sapmnt]
  NFS IP                                              [as0009lx]
Notification Script                                 []
  SAPADMUSER                                          [ha1adm]
  LOGGER LOGFILE                                      [/var/hacmp/log/SAPuti>
  CS OS Connector [1]                     +
  ERS OS Connector                                    [1]                     +
* Application Name                                     SAP_HA1_ASCS00
  AS OS Connector                                     [1]                     +
  SAP Instance Profile                                [/usr/sap/HA1/SYS/prof>
  SAP Instance Executable Directory                   [/usr/sap/HA1/ASCS00/e>
  Sapcontrol WaitforStarted Timeout                   [60]
  Sapcontrol WaitforStarted Timeout Delay             [0]
  Sapcontrol WaitforStopped Timeout                   [10]
Sapcontrol WaitforStopped Timeout Delay             [0]
  ENSA/ERS Sync time                                  [10]

  ************* SAP Globals WARNING ****************
   Changing these values will effect all instances
  **************************************************

  Is NFS                                              [1]                     +
  SAPMNT Export Directory                             [/export/sapmnt]
  NFS IP                                              [as0009lx]
  Notification Script                                 []
```

Figure 7-67 Change/Show SAP instance attributes

Figure 7-67 fields explanation:

– SAP SYSTEM ID, SAP SCS/ASCS Instance(s) Name(s), Application Name: Identifier, static value.

– SAP Instance Profile:

SAP systems can access the instance profiles through different ways:

- `/sapmnt/SID/profile`:

  This directory is available only if the /sapmnt share is available. It is an outage risk for the SAP instance if this directory is not accessible.

- `/usr/sap/SID/SYS/profile` (default):

  This is a link to the SAP global profile. It is an outage risk for the SAP instance if this is not accessible.

- `/usr/sap/SID/INSTANCE/profile`:

  Some instance types, such as ERS, have a local copy of the profile inside the instance directory. The exposure of this location is the currency of the information, because the sapcpe is often not configured to update that profile.

The default is `/usr/sap/SID/SYS/profile`. However, availability can be increased if it is ensured that the instance profile is always kept current by SAP mechanisms in the instance directory. This is not supported for all NetWeaver releases. Please contact your SAP representative for release-specific information.

– SAP Instance Executable Directory:

SAP systems can access the instance executables through different ways (without access, the SAP system cannot be started and will crash over the time):

- `/sapmnt/SID/…/exe`:

  The most unreliable option to access the SAP executables. It is available only as long the `/sapmnt` share. The exact path depends on the SAP Kernel and runtime settings what makes this path unreliable if a change occurs.

- `/usr/sap/SID/SYS/exe/run`:

  A link to the SAP global. The correct linking is ensured by SAP. However, the executables can be accessed only if the share is available.

- `/usr/sap/SID/INSTANCE/exe`:

  The instance directories have a local copy of the executable configured to be copied by sapcpe (see also Chapter 10.7.1, “SAP executable resiliency (sapcpe)” on page 211).This is the most robust access path.

> **Attention:** Setup sapcpe to copy all appropriate information. After each SAP upgrade, re-verify the sapcpe setup as an SAP upgrade might overwrite settings.

– Sapcontrol WaitforStarted Timeout
Sapcontrol WaitforStarted Timeout Delay
Sapcontrol WaitforStopped Timeout
Sapcontrol WaitforStopped Timeout Delay:

These four fields have their main impact on SCS and ERS instances and are merely informational for app server instances. The SCS and ERS instances start or stop scripts that run in foreground, which means that the start script runs until finished and then hands off to the monitors. These are the calls from SAP that are used in the scripts:

`sapcontrol –nr <inst_no> -function WaitforStarted <Sapcontrol WaitforStarted Timeout> <Sapcontrol WaitforStarted Timeout Delay>`” (similar for `WaitforStopped`)

Leave the defaults unless race conditions are seen on the systems.

– ENSA/ERS Sync time:

Time granted on SAP system start to let the ERS sync with its SCS instance.

– Is ERS Enabled:

Some existing systems are running without ERS. All ERS-specific handling can be turned off by setting this value to “0.”

– Notification Level, Notification Script:

For more information, see 7.7.3, “Notification” on page 210.

– SA SAP XPLATFORM LOGGING:

Defines the log level of the script connector between SAP and PowerHA. See 7.6, “OS script connector” on page 206.

– EXIT CODE START sapcontrol Start failed
EXIT CODE START sapcontrol StartService failed
EXIT_CODE_START_sapcontrol_NFS_failed:

These values should not be changed unless explicitly instructed by IBM. It is changing the return code of the Start script under certain conditions.

> **Attention:** PowerHA 7.1.3 has a different return code behavior from previous releases.

– EXIT_CODE_MONITOR_sapstartsrv_unavailable
EXIT_CODE_MONITOR_failover_on_gw_outage:

These two variables define the return code of the PowerHA Application Monitor in case the sapstartsrv process is not available or in case the gateway is not available. Depending on the landscape and your needs, this might be already an issue that requires a failover or an operation that can be continued for other landscapes.

The preceding values are per instance. The following values are effective for all instances:

– Is this an NFS mountpoint?

Set to **1** in case the SAPGlobal is served by an NFS server. It does not matter whether from within this cluster or from outside the cluster.

– SAPMNT Export Directory, NFS IP:

Defines the NFS export directory and IP if “Is this an NFS mount point?” is set to 1.

– SAPADMUSER:

This is the SAP OS <sid>adm user.

> **Attention:** The SAP <sid>adm user will be called using the LANG C environment, in case the “env” output differs between LANG C. Either the environment for the SAP user for LANG C must be updated or a PMR request to change the ODM entry must be opened.

– LOGGER LOGFILE:

Defines the log file where advanced logging information is written.

– CS OS Connector, ERS OS Connector and AS OS Connector:

Online on/off switch for the SAP HA Script connector. Default is 0. As soon as the script connector is enabled, it must be set to 1 manually.

Things to know before adding a Startafter resource group dependency

Resources without *startafter* definitions tend to come online first. If a `startafter` dependency between a SCS instance and NFS is configured, then an ERS instance tries to be fully started before NFS and CS. This will cause the ERS RG to fail.

Not defining any `startafter` dependency might bring the cluster resources online in non-specific order. But if the environment starts properly, this is the preferred way doing it. Please test online both nodes, online single nodes, crash and reintegrate node.

If you configure *startafter* RG dependencies, start with the SCS instances, followed by the ERS instances. This results in a consistent startup. However, clusters with or without `startafter` dependencies can both work.

Important: Parent-child RG dependencies *must not* be used. A parent-child relationship forces the child to stop and restart as the parent does. The SAP architecture can handle reconnects and provide continuous operation if configured in the SAP instance profiles correctly.

7.6 OS script connector

This section describes the steps to enable the OS script connector.

7.6.1 Plan

The planning involves verifying whether the SAP release is capable of supporting this function and which instances should be activated for this (see 7.5.5, “Customize” on page 198 for instance attribute settings for CS OS Connector, ERS OS Connector, and AS OS Connector).

In `/usr/sap/hostcontrol/exe` or in the instance directory, a saphascriptco.so library should exist. This is the minimum requirement. But it is recommended to use the latest patch as documented by your SAP guide. Remember to restart your host agents after the update.

Attention: Enable this function only if the following SAP prerequisites are met:

- Install with a stand-alone enqueue (CS) and enqueue replication (ERS).
- Minimum SAP NetWeaver, kernel, and patch level requirements are met.

7.6.2 Install

This section shows the installation process.

SAP instance profile

In our test environment, as Example 7-25 shows, we configured the variables in the instance profiles that are described in 7.4.1, "Identify the SAP software and SAP manuals" on page 165. The profiles are related to the clustered SAP instances.

Example 7-25 Variables configured in instance profiles

```
service/halib = /usr/sap/<SID>/<instance>/exe/saphascriptco.so
#typically automated by SAP. Can be also a path inside the host control
service/halib_cluster_connector =
/usr/es/sbin/cluster/sa/sap/sbin/sap_powerha_cluster_connector
```

Ensure that you are compliant with SAP Note 897933 (Start and stop sequence for SAP systems).

Debug

The following SAP profile variable can be used to control the debug level of the HAlib:

```
service/halib_debug_level = <value> (value range 0..3)
```

Setting this variable to a value of `2` or higher in the SAP instance profile causes more detailed information to be written to the `sapstartsrv.log`. To activate it, you need to restart *sapstartsrv*.

Detailed SAP logs can be found in the `/usr/sap/<SID>/<INSTANCE>/work/sapstartsrv.log`. You can find log enablement details for the script connector in 7.7.2, "Logging" on page 209.

7.6.3 Verify

Perform start/stop operations for all instances from the SAP Microsoft management console (MMC). Verify that the SAP application behavior is as expected and verify the log files. The log files can be found as configured in `/usr/es/sabin/cluster/sa/sap/SAP/SAPGlobals`.

7.7 Additional preferred practices

This section provides a few notes on preferred practices for SAP and PowerHA.

7.7.1 SAP executable resiliency (sapcpe)

The SAP executable and instance profiles are physically located inside the NFS-mounted SAP global file system. This brings a risk during runtime, start, and stop if the NFS and all of its content becomes unavailable.

The following sections provide the required steps for protection against such outages.

SAP sapcpe copy tool

SAP has a tool called *sapcpe*, which is triggered from within the instance profile at each start of an instance. This provides protection during the runtime and stops of an SAP instance. On initial startup, there is no protection, because all SAP mechanisms that are available today require a fresh copy from the SAP globals.

Plan

The sapcpe tool physically copies the executable from the SAP global directory (/sapmnt/SID/…) into the instance directory, based on “list files,” or directories, as specified in the SAP instance profile. The list files are included with the SAP kernel. Examples are scs.lst and ers.lst.

These list files do not copy all executables and libraries into the instance directories. To get full protection, you must extend the function to copy all executables into the instance directories by either creating your own .lst file, manually copying the entire executable directory, or modifying the existing list files.

Table 7-11 on page 208 shows the sapcpe enablement methods and gives an overview of the deployment effort and a pro-and-con decision aid for the different options. The recommendation is to copy all executables.

Table 7-11 sapcpe enablement methods

Method	Effort	Benefit	Disadvantage
Modify existing list files.	Add executables to the list files.	Easy to initially enable.	A kernel upgrade will typically overwrite these edited files and all extensions will be lost. Manually adding executables includes the risk of missing one.
Add new list files.	Create a list file and enable it inside the instance profile.	List files do not get silently overwritten by a kernel upgrade.	Manually add executables includes the risk of missing one. List can change between kernels.
Copy the entire set of executables.	Change the sapcpe command in the instance profile to copy a full directory.	Do it once.	Required for each instance enabled: 2.5 - 3 GB of space.

Install

In this section, changes in the instance profile for the recommended option to copy all executables are described.

For each SAP instance, the following change in the instance profile (and for older SAP releases the instance Startup profile) must be made, as shown in Example 7-26.

Example 7-26 Changes for the instance profile

```
#vi /sapmnt/<SID>/profile/ HA1_ASCS00_as0010lx
[…]
#----------------------------------------------------------------------
# Copy SAP Executables
#----------------------------------------------------------------------
_CPARGO = list:$(DIR_CT_RUN)/scs.lst
Execute_00 = immediate $(DIR_CT_RUN)/sapcpe$(FT_EXE) pf=$(_PF) $(_CPARGO)
ssl/ssl_lib = $(DIR_EXECUTABLE)$(DIR_SEP)$(FT_DLL_PREFIX)sapcrypto$(FT_DLL)
sec/libsapsecu = $(ssl/ssl_lib)
ssf/ssfapi_lib = $(ssl/ssl_lib)
SETENV_05 = SECUDIR=$(DIR_INSTANCE)/sec
```

```
_CPARG1 = list:$(DIR_CT_RUN)/sapcrypto.lst
#Execute_01 = immediate $(DIR_CT_RUN)/sapcpe$(FT_EXE) pf=$(_PF) $(_CPARG1)
Execute_01 = immediate $(DIR_CT_RUN)/sapcpe source:/sapmnt/HA1/exe/uc/rs6000_64
target:$(DIR_EXECUTABLE) copy all
[…]
```

Paths inside the instance profile

The discovery and addition in Smart Assist is using sapcontrol to get paths to the instance profile, the instance executable, and others. If these paths inside the SAP instance profile are set to the SAP global variable (for example: /sapmnt/SID/profile/SID_ERSxx_ip) or to the SYS directory that has logical links down to the NFS (for example: /usr/sap/SID/SYS/exe/run), there is downtime risk if there is an outage on the NFS. Some instances, such as the ERS, have an instance profile copy inside the instance directory (/usr/sap/SID/ERSxx/profile/<inst-profile>). These differ between SAP releases.

You can remove this risk by either of these two methods:

- After the addition of the Smart Assist resource group, change the paths in the Change/Show SMIT panel.
- Before discovering and adding the instance to the cluster, change the values in the instance profile.

7.7.2 Logging

Smart Assist provides a method of fine-tuning the logging to avoid log flooding. Before handover to production, the appropriate log levels must be defined accordingly to the space and requirements. A full log file directory can result in outages. Therefore, alerts should be implemented to protect from full file systems.

Besides hacmp.out, PowerHA has Smart Assist-specific logs. Of special relevance is the /var/hacmp/log/sapsa.log. Besides the default PowerHA logging, two tunable pairs can be used for advanced logging.

- To log detailed SAP command output, **select smitty clsa** → **Change/Show the SAP Instance Attributes**.

  For each instance, repeat these steps according to the requirements of the business application.

  The first tunable specifies the log level (0 - 3), and the second tunable specifies the location to write the logs to.

  Besides the standard logging, the SAP commands called to start, stop, and monitor will be logged. For your quality assurance tests, it is recommended to set the level to 3. The runtime default is 0. See Example 7-27.

Example 7-27 Change/show SAP ERS Instances (s) attribute details menu

```
         Change/Show SAP ERS Instance(s) attributes Details
[TOP]                                     [Entry Fields]
* SAP SYSTEM ID                           HA2
* SAP ERS Instance(s) Name(s)             ERS22
* Application Name                        SAP_HA2_ERS22
[…]
  SA SAP XPLATFORM LOGGING                [3]                     +
[…]
  LOGGER LOGFILE                          [/var/hacmp/log/SAPutils.log]
```

```
[…]
```

- To log SAP HA script connector output:

  In /usr/es/sbin/cluster/sa/sap/etc/SAPGlobals, the following two parameters can be set to specify the log file location (must be writable on all nodes) and the level (if the log level is below 2, all previously created log files will be deleted and will only show the last operation):

  ```
  OSCON_LOG_FILE=$(echo $SALOGFILEPATH/sap_powerha_script_connector.log")
  OSCON_LogLevel=0  #valid log level 0=no logging 3=max
  ```

  Note: Repeat the setting on all nodes.

7.7.3 Notification

In addition to the standard PowerHA Application Monitor Notification methods, Smart Assist for SAP provides the option to give advanced alerts and optimization information by enabling internal notification methods about events. The start and stop monitor scripts can inform about states where the cluster should continue but should be manually verified if the current situation degrades the productivity of the business application. It also helps to optimize the timeout values and other settings over time.

Create notification script

This script if a free script that can be implemented to send SMS, email, or simply log messages. Example 7-28 shows a small sample script for writing the notifications to a log file.

Example 7-28 Script to write the notifications to a log file

```
#vi notify.sh
      #!/bin/ksh93
      typeset DATE="$(date +"%y%m%d %H:%M:%S") "      print "${DATE} $*" >>
/var/hacmp/log/notify_logger.log
#chmod 755 notify.sh
#touch /var/hacmp/log/notify_logger.log
```

Create the script on all cluster nodes

Enable the notification configuration with **smitty clsa** → **Change/Show the SAP Instance Attributes** SMIT menu, as shown in Example 7-29.

For each instance, repeat the following steps according to the relevance for the business application. Set the notification level and specify the notification script.

Example 7-29 Enabling the notification

```
              Change/Show SAP SCS/ASCS Instance(s) attributes Details [TOP]
[Entry Fields]
* SAP SYSTEM ID                                       HA2
* SAP SCS/ASCS Instance(s) Name(s)                    ASCS00
* Application Name                                    SAP_HA2_ASCS00
  […]
  Notification Level                                 [5]                    +
[…]
  Notification Script                                [/usr/sap/notify.sh]
[…]
```

The notification levels are defined as follows:

Level 0	Disables all notifications.
Level 1 - 3	Sends notifications from the monitor script (1 only for severe issues, 3 for every warning).
Level 4 - 5	Sends notifications from the start script.
Level 6 - 8	Reserved for future purposes.

The script is called with following input parameters, which can be used to define the message inside the notification:

```
<my notification script>.sh "Instance ${INSTANCE} of ${SID} - <description>.\n"
```

7.7.4 Monitor node-specific IP aliases for SAP application servers

Following best practices, each instance is assigned a dedicated virtual IP. The IP can be added to the resource group. This enables monitoring of the availability of the IP.

7.8 Migration

Migrating to PowerHA 7.1.3 in an SAP landscape has two different aspects:

- The first aspect is to migrate the PowerHA base product. This is described in Chapter 4, "Migration" on page 49.
- The second aspect, which requires more planning, is to also ensure that the business application logic survives the migration. In general, there are two approaches:
  - Run the same logic and capabilities with a new cluster base product and ensure that the transition works.
  - Enrich the cluster functionality by using new Smart Assist or base product capabilities.

7.8.1 Migrating from PowerHA 6.1 to 7.1.3

The following steps help with the migration from PowerHA 6.1 to PowerHA 7.1.3:

1. Check whether the application logic can be migrated or, preferably, rediscover the cluster with Smart Assist.
2. Verify whether your SAP application installation fulfills the disk, IP, and SAP release prerequisites for Smart Assist.
3. Plan for MC/UC and CAA.
4. Application migration considerations:
   - Configurations with FDDI, ATM, X.25, and token ring cannot be migrated and must be removed from the configuration.
   - Configurations with heartbeat via alias cannot be migrated and must be removed from the configuration.
   - Non-IP networking is accomplished differently.
   - RS232, TMSCSI, TMSSA, and disk heartbeat are not supported, and the configuration data will not be in the migrated cluster.
   - PowerHA/XD configurations cannot be migrated to version 7.1.1.

– Due to the radically different communication infrastructure and AIX migration, active rolling migration is not outage-free.
– IP address takeover (IPAT) via alias is now the only IPAT option available. Therefore, the following IPAT options cannot be migrated:
  - IPAT via replacement is not possible.
  - IPAT via hostname takeover is not possible.
  - The cluster remote command **clrsh** is not compatible.
– After migration, never mix *startafter* or *stopafter* dependencies with processing orders (acquisition order, release order).

7.8.2 Migrating from PowerHA version 7.1.0 or 7.1.2 to version 7.1.3

You can exchange your base product but not migrate to the new capabilities in PowerHA. Due to the new capabilities, such as the SAP HA API and the increased RG flexibility, a downtime migration is required. The previous RGs need to be removed by using the Smart Assist menu optin to “Remove an Application from the PowerHA SystemMirror Configuration.”

Before considering a Smart Assist migration, verify the capability of your setup to be rediscovered with the new SAP-integrated solution:

- The SCS instances with dual stack must be split into separate resource groups. Verify that a dedicated service IP aliases have been assigned to each of them.
- Verify that each of them has its own VG in case no local file system approach is used.

The ERS instances must follow the same rules as the SCS instances:

- Verify whether a dedicated service IP alias has been assigned to each of them.
- Verify whether each of them has its own VG in case no local file system approach is used.

If the installation is not compliant with these requirements, you have two options:

1. Reinstall the instances to meet the IP alias and file system requirements. This affects only the SCS and ERS instances, which can be fast and easy to reinstall. However, the downtime is a consideration.

   The detailed installation steps and Smart Assist addition are described in this chapter, starting from 7.3, “Installation of SAP NetWeaver with PowerHA Smart Assist for SAP 7.1.3” on page 143.
2. Migrate only the base PowerHA product and stay with the 7.1.2 scripts.

7.9 Administration

This section provides administration information.

7.9.1 Maintenance mode of the cluster

The cluster manages the virtual IPs and the file systems in the case of a moving volume groups approach rather than a local disk approach. Because of this, a cluster shutdown removes vital resources that required for the application to function.

This procedure is occasionally requested by SAP Support in case of problem analysis to separate SAP issues from IBM-related ones, while remaining operative. The first option, *maintenance mode*, is the standard method. The second option is provided if maintenance mode is not appropriate. It is called *suspend application health check*.

> **Note:** After the cluster shutdown and before the instance is moved back under cluster control, ensure that it is in the same operational status (online or offline) and on the same node as it was before the maintenance mode was enabled.

This maintenance can be performed in two ways as described in the following sections.

Maintenance mode

PowerHA has a maintenance mode for bringing the RGs into an unmanaged state and leaves the entire infrastructure up, without any kind of cluster protection.

The effect is that no recovery action, even in the case of a node failure, is triggered. This is in effect for all resources that are controlled in this cluster.

The process starts with the **smitty cl_stop** SMIT menu, as shown in Example 7-30.

Example 7-30 Stop cluster services

```
Stop Cluster Services
                                                       [Entry Fields]
* Stop now, on system restart or both                 now                    +
  Stop Cluster Services on these nodes               [as0007lx,as0008lx]     +
  BROADCAST cluster shutdown?                         true                   +
* Select an Action on Resource Groups                 Unmanage Resource Groups +
```

When “re-managing” the RGs, the cluster manager puts the RGs into the same state as before the unmanage action. This means that if the application was running, it will be put into a running state. To do so, start the nodes with the **smitty cl_start** menu.

Suspend application health check

Often, it is sufficient to just disable the application health monitor. The effect is to ignore the instance for which the monitor has been disabled. The cluster still reacts to hardware outages.

As soon the monitor is reactivated, the instance is reactivated, as well as part of the cluster logic. This is in case the instance was stopped or brought online on a different node than the associated RG.

To suspend the monitoring, select **smitty cl_admin** → **Resource Groups and Applications** → **Suspend/Resume Application Monitoring** → **Suspend Application Monitoring**.

In the following panels, select the PowerHA Application Monitor and the associated resource group to deactivate them.

To enable the monitor again, select **smitty cl_admin** → **Resource Groups and Applications** → **Suspend/Resume Application Monitoring** → **Resume Application Monitoring**.

7.10 Documentation and related information

The following publications provide documentation and other useful information:

- Invincible Supply Chain - SAP APO Hot Standby liveCache on IBM Power Systems:

  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100677

  – Implementation Guide for the liveCache HotStandby
  – Administration and Migration Guide for the liveCache HotStandby
- Smart Assist for SAP liveCache HotStandby, IBM Knowledge Center:

  http://ibm.co/1nsXElC
- High Available Core SAP System with IBM DB2 HADR and Tivoli SA MP (white paper)

  http://ibm.co/1qX8acA
- How to use the SAPControl web service interface, SAP Community Network

  http://bit.ly/1scZC1a
- SAP notes:
  – SAP note 877795: Problems with sapstartsrv as of Release 7.00 and 640 patch 169

    http://bit.ly/1mmVnZv
  – SAP note 927637: Web service authentication in sapstartsrv as of Release 7.00

    http://bit.ly/1mmVsfJ
  – SAP note 1439348: Extended security settings for sapstartsrv

    http://bit.ly/1mmVy6Y
  – SAP note 729945: Auto-restart function for processes in sapstartsrv

    http://bit.ly/1odanie
  – SAP note 768727: Process automatic restart functions in sapstart

    http://bit.ly/1sx4xsn

8

PowerHA HyperSwap updates

This chapter describes new features of the HyperSwap function with IBM PowerHA SystemMirror Enterprise Edition, version 7.1.3. These new features are explained by using a few configuration examples in this chapter. Some configuration best practices are also mentioned in this chapter to provide an easy way to implement and deploy highly available applications that are protected by PowerHA SystemMirror with HyperSwap. This chapter covers the following topics:

- HyperSwap concepts and terminology
  - HyperSwap enhancements in PowerHA SystemMirror 7.1.3
  - HyperSwap reference architecture
- Planning a HyperSwap environment
- HyperSwap environment requirements
- Configuring HyperSwap for PowerHA SystemMirror
  - HyperSwap storage configuration for PowerHA node cluster
  - Configure disks for the HyperSwap environment
- HyperSwap deployment options
- Single-node HyperSwap deployment
  - Oracle single-instance database with Automatic Storage Management in single-node HyperSwap
  - Single-node HyperSwap: Planned HyperSwap
  - Single-node HyperSwap: Unplanned HyperSwap
- Node-level unmanage mode
- Testing HyperSwap
  - Single-node HyperSwap tests
  - Oracle Real Application Clusters in a HyperSwap environment
- Troubleshooting HyperSwap

Scenarios such as protecting applications by HyperSwap in active-passive configuration in linked cluster are not covered, because they are well documented in these other IBM Redbooks publications:

- *Deploying PowerHA Solution with AIX HyperSwap*, REDP-4954

  `http://www.redbooks.ibm.com/redpieces/abstracts/redp4954.html?Open`

- *IBM PowerHA SystemMirror 7.1.2 Enterprise Edition for AIX*, SG24-8106

  `http://www.redbooks.ibm.com/abstracts/sg248106.html?Open`

8.1 HyperSwap concepts and terminology

PowerHA SystemMirror Enterprise Edition 7.1.2 introduced new storage high availability and a disaster recovery feature named *HyperSwap* was released.

The HyperSwap technology concept on IBM Power Systems has its roots on IBM System z mainframe servers, where HyperSwap is managed through the IBM Geographically Dispersed Parallel Sysplex™ (IBM GDPS®). In Power Systems, PowerHA SystemMirror Enterprise Edition is the managing software that provides the capability to handle remote copy and automate recovery procedures for planned or unplanned outages (based on HyperSwap function).

The HyperSwap feature swaps a large number of devices and enhances application availability over storage errors by using the IBM DS8000 Metro Mirror Copy Services.

Currently, the HyperSwap function can handle IBM DS8000 Metro Mirror (formerly Peer-to-Peer Remote Copy, PPRC) relationships (two-site synchronous mirroring configurations). Additional enhancements are being considered for Global Mirror configurations. Therefore, configurations with the IBM DS88xx storage systems can be used for HyperSwap configurations.

The HyperSwap function provides storage swap for application input/output (I/O) if errors occur on the primary storage. It relies on in-band communication with the storage systems by sending control storage management commands through the same communication channel that is used for data I/O.

To benefit from the HyperSwap function, the primary and auxiliary volume groups (LUNs) are reachable on the same node. Traditional Metro Mirror (PPRC) can coexist. In that case, the volume group from the primary storage is visible on one site and the secondary volume group on the secondary site.

8.2 HyperSwap deployment options

HyperSwap can be deployed on a single-node environment and in a multiple-site environment in these situations:

- One PowerHA SystemMirror cluster node is connected to two IBM DS88xx storage systems. The storage systems can be on the same or different sites. Single-node HyperSwap configuration protects against storage failures.
- A cluster can have two or more nodes that are distributed across two sites. The cluster can be a linked or a stretched cluster.

8.2.1 HyperSwap mirror groups

The HyperSwap function relies on mirror group configuration. A *mirror group* in HyperSwap for PowerHA SystemMirror represents a container of disks (logical grouping) that has the following characteristics:

  – Mirror groups contain information about the disk pairs across the sites. This information is used to handle Peer-to-Peer Remote Copy (PPRC) pairs.
  – Mirror groups can consist of IBM AIX volume groups or a set of raw disks that are not managed by the AIX operating system.

  – All disks that are part of a mirror group are configured for replication consistency.

The following types of mirror groups can be configured in HyperSwap:

- *User mirror groups* are used for application shared disks (disks that are managed by PowerHA resource groups). The HyperSwap function is prioritized internally by PowerHA SystemMirror and is considered low-priority.
- *SystemMirror groups* are used for disks that are critical to system operation, such as rootvg disks and paging space disks. This type of mirror group is used for mirroring a copy of data that is not used by any other node or site.
- *Repository mirror groups* represent the cluster repository disks used by Cluster Aware AIX (CAA).

8.3 HyperSwap enhancements in PowerHA SystemMirror 7.1.3

PowerHA SystemMirror 7.1.3 introduced the following HyperSwap enhancements:

- Active-Active sites:
  – Active-Active workload across sites
  – Continuous availability of site-level compute and storage outages.
  – Support for Oracle Real Application Clusters (RAC) extended distance deployment.
  – Single-node HyperSwap.
  – Support storage HyperSwap for AIX LPAR (no need for a second node in the cluster).
  – When protection against storage failures is required for one compute node, the HyperSwap function can be enabled.
- Automatic resynchronization of mirroring.
- Node-level “Unmanage Mode” support:
  – The HyperSwap function is deactivated when resource groups are in an Unmanaged state. The option allows reconfiguration of disks in Mirror Group definition while a resource group is in an Unmanaged state and the application is online.
  – This enables Live Partition Mobility (LPM) of an LPAR that is managed normally by PowerHA with HyperSwap. When LPM is used, PowerHA SystemMirror Clusters events should be evaluated while LPM is in progress. Practically, if PowerHA SystemMirror leaves the resource groups in an Unmanaged state, the HyperSwap function will be also disabled. The HyperSwap function will resume when resource groups are brought online.
- Enhanced repository disk swap management.
  – The administrator can avoid specifying standby disk for repository swap handling.
- Dynamic policy management support:
  – The administrator can modify the HyperSwap policies across the cluster.
  – Example: Expand or delete mirror groups combined with an unmanaged resource.
- Enhanced verification and reliability, availability, and serviceability (RAS).

8.4 HyperSwap reference architecture

To support HyperSwap, there are changes in both the storage and AIX. The change in storage is the implementation of in-band communication, which is described in 8.4.1, "In-band storage management" on page 218.

The AIX operating system changes for HyperSwap support are described in 8.4.2, "AIX support for HyperSwap" on page 220.

8.4.1 In-band storage management

To support a more reliable, more resilient, and lower-latency environment for storage systems that are capable of supporting HyperSwap, IBM developed *in-band* storage management to replace the out-of-band storage management that is used in the traditional SAN storage environment. This in-band storage management infrastructure plays an important role, especially in clusters across sites.

Out-of-band and in-band storage management differences

The data path between the host server and the storage controller is critical to the reliability and performance of a storage system. Therefore, the storage management usually uses a separate path to issue storage commands for storage management and monitoring. This path is usually via a TCP/IP network to a specialized storage subsystem control element or device that performs the storage agent function. This type of storage management, shown in Figure 8-1, is called *out-of-band*.

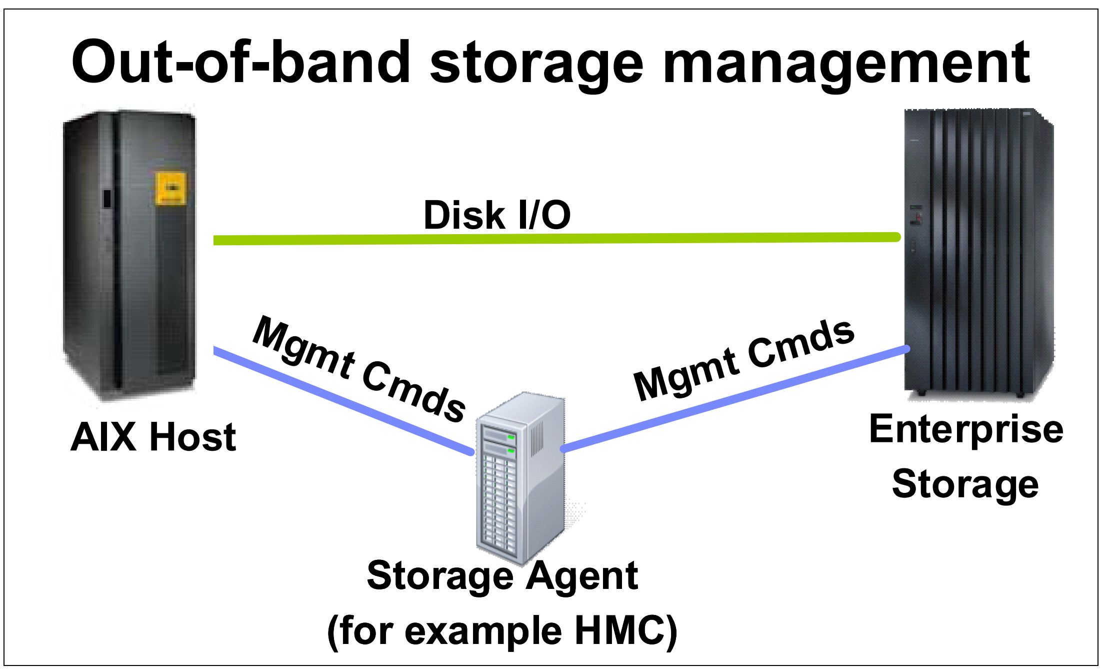


Figure 8-1 Out-of-band storage system

As the storage system evolves in size and complexity, out-of-band architecture becomes inadequate for the following reasons:

- The original consideration was moving the storage management communication out of the data path to eliminate the impact on performance of the critical data throughput. This consideration becomes a lower-priority issue, because the bandwidth of the data path bandwidth grows significantly.

- As the SAN network spans a longer distance, the reliability and latency of the TCP/IP network becomes an issue.

Therefore, it becomes necessary to replace the TCP/IP network for the storage management to support more storage systems. In-band communication is best suited for this purpose. Figure 8-2 shows an example of in-band management of a storage system.

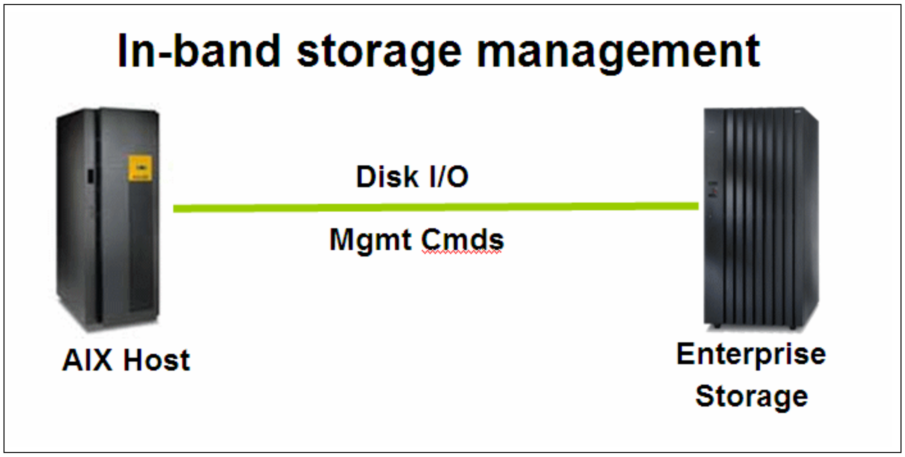


Figure 8-2 In-band storage system

Both data and storage management share the same Fibre Channel (FC) network. This offers two key advantages:

- The FC network is usually faster than a TCP network (lower latency).
- The separate storage agent (for example, the storage Hardware Management Console) that is used in the out-of-band structure is no longer needed. The management communication between host server and storage controller becomes more direct and, as such, more reliable and faster.

8.4.2 AIX support for HyperSwap

Figure 8-3 shows the diagram of the components supporting HyperSwap.

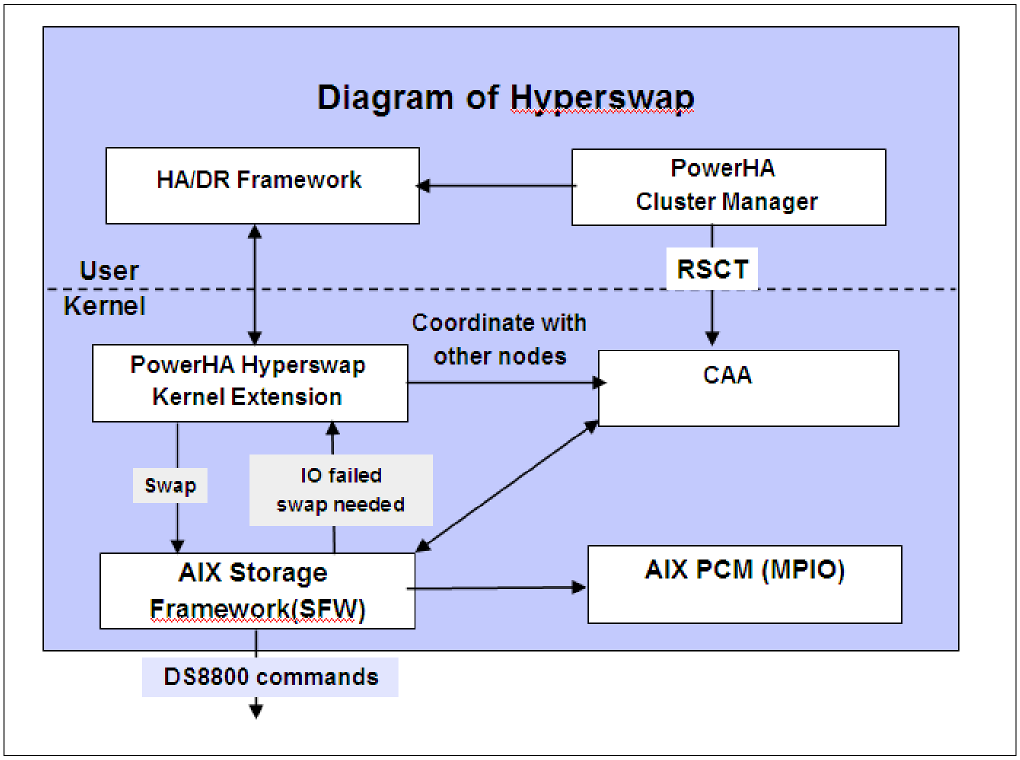


Figure 8-3 Diagram of HyperSwap

Note: Reliable Scalable Cluster Technology (RSCT) is a set of software compnents and tools that provide a comprehensive clutering environment for AIX. It is or has been used by products such as PowerHA and GPFS, among others.

These are the HyperSwap-related components:

- Cluster Aware AIX (CAA), which orchestrates cluster-wide actions.
- PowerHA HyperSwap kernel extension:
  - Works with CAA to coordinate actions with other nodes.
  - Analyzes the messages from the PowerHA and AIX storage frameworks (SFW) and takes proper actions.
  - Determines the swap action.
- AIX storage framework (SFM):
  - Works as the AIX interface to the storage.
  - Works closely with the PowerHA HyperSwap kernel extension.
  - Manages the status of the storage.
  - Informs the PowerHA HyperSwap kernel extension about I/O errors.
  - Receives swap decisions from the PowerHA HyperSwap kernel extension and sends orders to AIX Path Control Module (MPIO).

8.4.3 AIX view of HyperSwap disks

Figure 8-4 shows the AIX view of the HyperSwap disks.

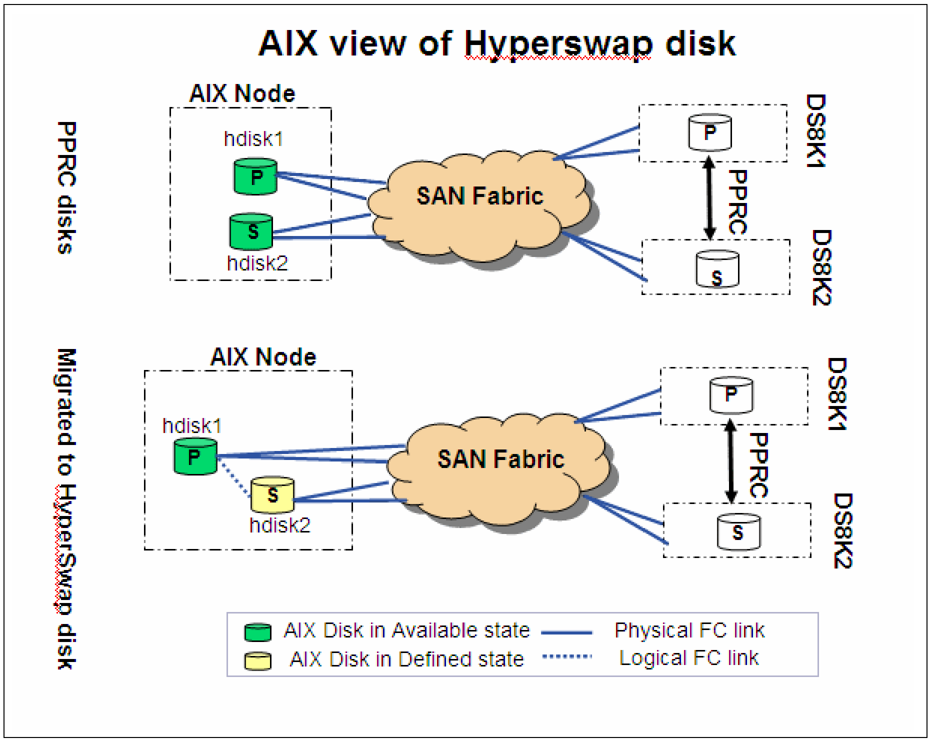


Figure 8-4 AIX view of the HyperSwap disk

Before configuring and enabling HyperSwap, AIX sees PPRC-paired disks hdisk1 and hdisk2, one from each of two storage subsystems, DS8K1 and DS8K2. In our example, hdisk1 is in DS8K1 and hdisk2 is in DS8K2. These two disks are both in *available* state. The AIX node has four FC paths to hdisk1 and four FC paths to hdisk2.

A new disk attribute, *migrate_disk*, has been implemented for HyperSwap. When one of the PPRC paired disks, say hdisk1, has been configured as migrate_disk, its peer-paired disk, hdisk2, is changed to the *defined* state. At that point, AIX can see eight paths to hdisk1, which is in the available state. In case the AIX node cannot access the PPRC source (hdisk1), the disk from DS8K1, the AIX kernel extension changes the path to access to the disk on DS8K2 while still using hdisk1 in AIX. This is called *HyperSwap* and is usually apparent to the application.

8.5 HyperSwap functions on PowerHA SystemMirror 7.1.3, Enterprise Edition

The followingHyperSwap enhancements were introduced in the new version of PowerHA SystemMirror 7.1.3, Enterprise Edition, to increase use options and functions and to offerenhanced flexibility for protected applications:

- Performs coordinated HyperSwap across multiple nodes
- Becomes useful for applications with concurrent I/O spanned over multiple nodes
- Allows planned and unplanned HyperSwap operations

- Provides storage error protection for single-node cluster: single-node HyperSwap
- Provides consistency semantics for data volumes that span multiple storage systems.
- Supports automatic resynchronization of mirroring, when needed
- Provides the flexibility for maintenance work without downtime
- Provides the flexibility for storage migration without downtime
- Provides enhanced repository disk swap management
- Provides dynamic policy management support
- Supports raw disks, volume groups, and logical volumes
- Supports user disks, repository disks, and system disks (such as rootvg, paging space disk, dump disks)

8.6 Limitations and restrictions

The following limitations and restrictions apply to the current HyperSwap version:

- SCSI reservation is not supported for HyperSwap-enabled configurations.
- Automatic resynchronization is not supported for single-node HyperSwap-enabled configuration.

  Users must manually resume replication after a replication is re-established.

  For DS8800 in-band Metro Mirror PPRC resources, automatic resynchronization is done through a SystemMirror join cleanup event.
- LPM requires node-level unmanaged HyperSwap,
- Dedicated logical subsystems (LSSes) are required for the HyperSwap-enabled mirror groups disks.

The previous statements are logical consequences of how mirror groups are managed in a HyperSwap environment. You can swap the disks that belong to a mirror group as a group and mirror groups at their turn, one by one, in case of manual swap, and all together due to an unplanned HyperSwap.

8.7 HyperSwap environment requirements

A HyperSwap environment relates to the AIX operating system, PowerHA System Mirror Enterprise Edition, storage systems and the ways of how applications are deployed and protected.

These are the requirements for AIX and the DS8800 microcode:

- AIX 7.1 TL3 or later, or AIX 6.1 TL9 or later.
- PowerHA SystemMirror 7.1.3 Enterprise Edition.

  If all file sets of PowerHA SystemMirror 7.1.3, Enterprise Edition, are not installed, check that the following HyperSwap-specific file sets are installed:

  – cluster.es.genxd.cmds
  – cluster.es.genxd.rte
  – devices.common.IBM.storfwork.rte
  – devices.common.IBM.mpio.rte

– devices.fcp.disk.rte

For AIX 7.1, the minimum file set level for these files is 7.1.3 and for AIX 6.1, it is 6.1.9.

- DS88xx with microcode 86.30.49.0 or later.
- DS88xx must be attached with N_Port ID Virtualization (NPIV), FC, or Fibre Channel over Ethernet (FCoE).
- DS8K storage systems licensed for Metro Mirror.
- SAN switches for FC, FCoE, NPIV, hosts, and storage systems connectivity.

8.8 Planning a HyperSwap environment

Depending on the number of disk groups that must be protected by HyperSwap and envisioning environment growth, besides the disks that are not used with HyperSwap, a carefully planned storage logical subsystem (LSS) is required. HyperSwap brings in three different types of mirror groups that cannot share or overlap with LSS.

When other LUNs from the same LSSes exist and some disks from the same LSSes exist in one mirror group, you cannot activate HyperSwap. HyperSwap is the application that assures data consistency on the target storage.

Keep these HyperSwap planning considerations in mind:

- Peer-to-Peer Remote Copy (PPRC) paths and relationships must be defined at the storage levels before you configure HyperSwap for PowerHA SystemMirror or in-band PPRC Metro Mirror for PowerHA SystemMirror.
- Disk replication relationships must adhere to a 1-to-1 relationship between the underlying LSSes.
- To maintain consistency group schematics, suspended operations on a storage device must function on the entire logical subsystem (LSS).
- NPIV attached storage configurations using the Virtual I/O Server are supported.
- Concurrent workloads across sites such as Oracle Real Application Clusters (RAC), and concurrent resource groups are supported in stretched clusters and linked clusters that are using HyperSwap enabled mirror groups.
- HyperSwap enables mirror groups for automatic resynchronization when a replication failure occurs. The operations are logged, and log files can be used to identify the cause of failures.
- To add a node to a mirror group, you must perform configuration operations from a node where all disks are accessible.
- Applications using raw disks are expected to open all the disks up front to enable the HyperSwap capability.
- Virtual SCSI (VSCSI) method of disk management is not supported.
- SCSI reservations are not supported for devices that use the HyperSwap function.
- When a mirror group configuration is changed, the mirror group activation requires that you verify and synchronize with the resources in Unmanaged state, using the unmanage mode level.
- If you change the mirror group configuration while the cluster services are active (DARE), these changes might be interpreted as failures, which result in unwanted cluster events. You must disable the HyperSwap function before you change any settings in an active cluster environment.

- LPM can be performed while the cluster nodes are in an Unmanaged state. If an unplanned event occurs while the node is in an Unmanaged state, the node is moved to a halt state.

HyperSwap does not automatically transfer the SCSI reservations (if any) from the primary to the secondary disks.

8.9 Configuring HyperSwap for PowerHA SystemMirror

The HyperSwap function relies on in-band communication of PowerHA SystemMirror cluster nodes with the storage systems and, subsequently, storage-to-storage communication for IBM DS8k Metro Mirror copy services.

Preparing the HyperSwap environment requires the following actions:

- Zoning configurations:
  - Cluster nodes for disk access on corresponding storage systems part of HyperSwap configuration.
  - Configure DS8800 Metro Mirror copy services designated for HyperSwap and for traditional Metro Mirror PPRC.
- DS8K Metro Mirror Copying Services for the replication of HyperSwap disks, based on prior planning:
  - Identify DS8K I/O ports for PPRC operations.
  - Configure PPRC paths.
  - Configure PPRC relationships.
  - Configure host attachments for HyperSwap.
  - Configure disks taking into account further storage migration based on planned disk LSSes.
  - Configure DS8k LUN masking for mirror groups and host connect storage attachment. Be sure to include disks that are not HyperSwap enabled.
- AIX operating system configurations:
  - Set up the Path Control Mode storage driver as the default. If any other storage driver is in place, the SSIC compatibility matrix must be checked.
  - Migrate the disks to be HyperSwap-enabled on the AIX operating system level.
- PowerHA SystemMirror configurations:
  - Configure the sites.
  - Configure the DS8000 Metro Mirror in-band resources (respective storage systems).
  - Configure mirror groups.
  - Configure resource groups.

8.10 HyperSwap storage configuration for PowerHA node cluster

The HyperSwap function requires in-band communication between storage systems and also with all PowerHA SystemMirror nodes in the cluster. The storage systems must be configured with bidirectional replicating paths for Metro Mirror Peer-to-Peer Remote Copy, and Metro Mirror relations must be established for the respective disks. Every PowerHA SystemMirror node in the cluster must be zoned to have the disks assigned from both storage systems as targets.

The host connect for every PowerHA SystemMirror node must be defined in the storage side as having this profile: *IBM pSeries - AIX with Powerswap support*. If the host connection has been defined in the storage system, it can be changed easily by using the **chhostconnect** command at the storage level, as shown in Example 8-1.

Example 8-1 Changing the hostconnect profile

```
dscli> chhostconnect -profile "IBM pSeries - AIX with Powerswap support" 00E3
Date/Time: November 26, 2013 3:06:56 PM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75LY981
CMUC00013I chhostconnect: Host connection 00E3 successfully modified.
```

All available profiles in the DS8800 storage system can be found with the **lsportprof storage_image_id** command. The storage image ID is obtained with the **lssi** command.

The IBM pSeries - AIX with Powerswap support profile does not have a default association for hostconnect HostType. Therefore, modifying the corresponding PowerHA SystemMirror node hostconnect can be done only by using the **chhostconnect -profile "IBM pSeries - AIX with Powerswap support" <host_id>** command.

8.11 HyperSwap Metro Mirror Copy Services configuration

The HyperSwap function relies on the DS88xx Metro Mirror Peer-to-Peer Remote Copy Services. Before configuring HyperSwap mirror groups at the PowerHA SystemMirror level, configure the Metro Mirror on the storage subsystems level.

Follow these steps to configure a Metro Mirror replication on DS88xx:

1. Depending on the number of volumes that are configured for HyperSwap, analyze and plan for LSSes and adapter port adapter allocation.
2. Validate the available PPRC ports for the LSS that you want by using the **lsavailpprcport** command in the storage systems. If you do not have an available port, you must allocate a dedicated port or adapter for the replication function as a next step establishing communication on the ports pair.
3. Given that your desired LSS could not exist on the storage level, define at least one volume in the chosen LSS and add it to corresponding volume group. This operation allows you to define the PPRC path for respective LSS later by using the **mkfbvol** command.
4. Create PPRC paths for the desired replicated LSSes by using the **mkpprcpath** command. Take into account consistency group parameters and how this will be activated at the PPRC path or LSS level Establish the Metro Mirror relation for the desired pairs of disks.
5. Validate the configuration and the replication disk status.

For more information about the DS8000 copy services, see the IBM Redbooks publication titled *IBM System Storage DS8000 Copy Services for Open Systems*, SG24-6788:

http://www.redbooks.ibm.com/redbooks/pdfs/sg246788.pdf

For example, we configure a Metro Mirror relationship for a pair of volumes. Assuming that the LSS planning has been done, we validate for our LSS C9 in storage IBM.2107-75NR571 and LSS EA on storage IBM.2107-75LY981 that we have available the PPRC ports, as shown in Example 8-2. In our example, the dscli version `6.6.0.305` is used.

Example 8-2 List of available PPRC ports between `IBM.2107-75NR571 and IBM.2107-75LY981`

```
dscli> lssi
Date/Time: February 6, 2014 4:49:03 PM CST IBM DSCLI Version: 6.6.0.305 DS: -
Name  ID               Storage Unit     Model WWNN             State  ESSNet
============================================================================
ds8k5 IBM.2107-75NR571 IBM.2107-75NR570 951   5005076309FFC5D5 Online Enabled
dscli>

dscli> lssi
Date/Time: February 6, 2014 4:48:34 PM CST IBM DSCLI Version: 6.6.0.305 DS: -
Name  ID               Storage Unit     Model WWNN             State  ESSNet
============================================================================
ds8k6 IBM.2107-75LY981 IBM.2107-75LY980 951   5005076308FFC6D4 Online Enabled
dscli> lsavailpprcport -remotedev IBM.2107-75NR571 -remotewwnn 5005076309FFC5D5 ea:c9
Date/Time: February 6, 2014 4:17:37 PM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75LY981
Local Port Attached Port Type
=============================
I0000      I0130         FCP
I0000      I0131         FCP
I0000      I0132         FCP
I0000      I0133         FCP
I0001      I0130         FCP
I0001      I0131         FCP
...................<<snippet>>.........
I0202      I0131         FCP
I0202      I0132         FCP
I0202      I0133         FCP
```

The desired LSSes are not yet available on the storage side because they do not own any volume. So we create new volumes, as shown in Example 8-3.

Example 8-3 Create volumes on LSS EA (storage `IBM.2107-75LY980 - C9 on` *IBM.2107-75NR571)*

```
dscli> mkfbvol -cap 1 -name r6r4m51_ea01 -extpool P4 -volgrp V37 ea01
Date/Time: February 6, 2014 4:05:22 PM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75LY981
CMUC00025I mkfbvol: FB volume EA01 successfully created.

dscli> mkfbvol -cap 1 -name r6m451_c901 -extpool P1 -volgrp v12 c901
Date/Time: February 6, 2014 4:02:50 PM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75NR571
CMUC00025I mkfbvol: FB volume C901 successfully created.
```

Now that the LSS is available and shown by using the **`lslss`** command, we create the PPRC paths for the chosen LSSes, as shown in Example 8-4 on page 227.

Example 8-4 mkpprcpath between STG ID IBM.2107-75LY981 and IBM.2107-75NR571

```
dscli> mkpprcpath -remotedev IBM.2107-75LY981 -remotewwnn 5005076308FFC6D4 -srclss
c9 -tgtlss ea -consistgrp I0231:I0130
Date/Time: February 6, 2014 5:04:53 PM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75NR571
CMUC00149I mkpprcpath: Remote Mirror and Copy path c9:ea successfully established.

dscli> mkpprcpath -remotedev IBM.2107-75NR571 -remotewwnn 5005076309FFC5D5 -srclss
ea -tgtlss c9 -consistgrp I0207:I0132
Date/Time: February 6, 2014 5:08:29 PM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75LY981
CMUC00149I mkpprcpath: Remote Mirror and Copy path ea:c9 successfully established.
```

Now, we establish the PPRC relationship, as shown in Example 8-5.

Example 8-5 mkpprc fbvol c901:ea01

```
dscli> mkpprc -remotedev IBM.2107-75LY981 -type mmir c901:ea01
Date/Time: February 6, 2014 5:10:28 PM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75NR571
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship C901:EA01
successfully created.
```

Now the PPRC relationship has been established and the disks can be configured at the operating system level.

8.12 HyperSwap PowerHA SystemMirror cluster node configuration

This section describes configuring a cluster node for HyperSwap requires, at the operating system level, activation of the AIX path control module, and setting up Fibre Channel attributes.

8.12.1 Change the multipath driver

On PowerHA SystemMirror nodes, the AIX Path Control Module (PCM) must be activated as the multipath driver used for the disks in HyperSwap environment. All cluster nodes must be configured to use the AIX_AAPCM driver.

The multipath driver used for specific storage families in the AIX operating system configuration can be found easily and configured by using the **manage_disk_drivers** command, as shown in Example 8-6.

Example 8-6 Multipath driver used in current environment

```
root@r6r4m51:/> manage_disk_drivers -l
Device              Present Driver        Driver Options
2810XIV             AIX_AAPCM             AIX_AAPCM,AIX_non_MPIO
DS4100              AIX_APPCM             AIX_APPCM,AIX_fcparray
DS4200              AIX_APPCM             AIX_APPCM,AIX_fcparray
DS4300              AIX_APPCM             AIX_APPCM,AIX_fcparray
DS4500              AIX_APPCM             AIX_APPCM,AIX_fcparray
DS4700              AIX_APPCM             AIX_APPCM,AIX_fcparray
```

```
DS4800              AIX_APPCM           AIX_APPCM,AIX_fcparray
DS3950              AIX_APPCM           AIX_APPCM
DS5020              AIX_APPCM           AIX_APPCM
DCS3700             AIX_APPCM           AIX_APPCM
DS5100/DS5300       AIX_APPCM           AIX_APPCM
DS3500              AIX_APPCM           AIX_APPCM
XIVCTRL             MPIO_XIVCTRL        MPIO_XIVCTRL,nonMPIO_XIVCTRL
2107DS8K            NO_OVERRIDE         NO_OVERRIDE,AIX_AAPCM
```

Use the same command for activating AIX_AAPCM as the default driver, as shown in Example 8-7. Changing the multipath driver requires a system reboot.

Example 8-7 Configuring multipath driver for DS8k systems to AIX_AAPCM

```
root@r6r4m51:/> manage_disk_drivers -d 2107DS8K -o AIX_AAPCM
 ********************** ATTENTION *************************
  For the change to take effect the system must be rebooted
```

After reboot, verify the present configured driver for the 2107DS8K device that represents the DS8xxxx storage family, as shown in Example 8-8.

Example 8-8 Explicitly set AIX_AAPCM as present driver

```
root@r6r4m51:/> manage_disk_drivers -l
Device              Present Driver      Driver Options
2810XIV             AIX_AAPCM           AIX_AAPCM,AIX_non_MPIO
DS4100              AIX_APPCM           AIX_APPCM,AIX_fcparray
DS4200              AIX_APPCM           AIX_APPCM,AIX_fcparray
DS4300              AIX_APPCM           AIX_APPCM,AIX_fcparray
DS4500              AIX_APPCM           AIX_APPCM,AIX_fcparray
DS4700              AIX_APPCM           AIX_APPCM,AIX_fcparray
DS4800              AIX_APPCM           AIX_APPCM,AIX_fcparray
DS3950              AIX_APPCM           AIX_APPCM
DS5020              AIX_APPCM           AIX_APPCM
DCS3700             AIX_APPCM           AIX_APPCM
DS5100/DS5300       AIX_APPCM           AIX_APPCM
DS3500              AIX_APPCM           AIX_APPCM
XIVCTRL             MPIO_XIVCTRL        MPIO_XIVCTRL,nonMPIO_XIVCTRL
2107DS8K            AIX_AAPCM           NO_OVERRIDE,AIX_AAPCM
```

Note: The DS8800 SDDPCM driver is not supported. If the DS8K Subsystem Device Driver Path Control Module (SDDPCM) driver is installed, it should be removed. By using the NO_OVERRIDE option, you can use SDDPCM to manage DS8000 storage systems families. We do not have SDDPCM installed on our system, so we left the NO_OVERRIDE value unchanged.

Since AIX 7.1 TL2, the ODM unique type field for DS8K managed by AIX Path Control Mode changed from `disk/fcp/mpioosdisk` to `disk/fcp/aixmpiods8k`. This change does not affect software the SDDPCM.

8.12.2 Change Fibre Channel controller protocol device attributes

The attributes shown in Table 8-1 must be changed for each Fibre Channel present in the system. This is to enhance system reaction speed and automatic system reconfiguration when a link event or SAN reconfiguration occurs.

Table 8-1 FC controller protocol device attributes

FC attribute	Value	Description
dyntrk	yes	Dynamic tracking of FC devices
fc_err_recov	fast_fail	FC fabric event error recovery policy

The command for changing the attributes in Table 8-1 is shown in Example 8-9.

Example 8-9 Changing FC protocol device attributes

```
root@r6r4m51:/> chdev -l fscsi0 -a dyntrk=yes -a fc_err_recov=fast_fail
fscsi0 changed
```

Note: By default, dynamic tracking is enabled on all systems that are running AIX 7.1.

8.13 Configure disks for the HyperSwap environment

HyperSwap disk configuration requires changing specific disk attributes after replication and LUN masking configuration in the storage side.

Enabling disks for HyperSwap configuration requires the following actions:

- Validate the actual disk configuration (inspecting disk attributes).
- Change policy reservation to `no_reserve` for all disks that will be activated for HyperSwap.
- Change the `san_rep_cfg` attribute on the disk that is located on the primary site.
- Depending on your test results, modify disk-tunable parameters.

Note: SCSI reservations are not supported for HyperSwap disks.

The command available for verifying and cancelling the disk reservation while PCM is the default driver is **`devrsrv -c query -l hdisk_name`**. The command output for hdisk31 is shown in Example 8-10.

Example 8-10 Querying hdisk SCSI reservation policy

```
root@r6r4m51:/work> devrsrv -c query -l hdisk31
Device Reservation State Information
==================================================
Device Name                     :  hdisk31
Device Open On Current Host?    :  NO
ODM Reservation Policy          :  NO RESERVE
Device Reservation State        :  NO RESERVE
```

Note: The reservation policy can be also changed to `no reserve` by using the **`chdev –a reserve_policy=no_reserve –l hdisk_number`** command.

The `san_rep_device` disk attribute shows the HyperSwap configuration hdisk state and capabilities of the system.

> **Note:** These are the possible attributes for the `san_rep_device`:
>
> | | |
> |---|---|
> | no [DEFAULT] | Does not support PPRC SCSI in-band. |
> | supported | Not a PPRC disk, but it supports PPRC SCSI in-band. |
> | detected | HyperSwap-capable. PPRC disk, which supports PPRC SCSI in-band. |
> | yes | PPRC-configured disk. This does not guarantee that the AIX host has access to both DS8Ks in the PPRC pair. |
>
> The **`lsattr -Rl hdisk_name -a san_rep_device`** command does not provide information regarding expected values.

Transforming a disk marked as a HyperSwap-capable requires changing the `san_rep_cfg` disk attribute to `migrate_disk`. The `san_rep_cfg` attribute disk should be modified on the disk that is the source in the Metro Mirror replication relationship. If the disk migration for HyperSwap is performed on the Metro Mirror target disk, PowerHA will fail to swap the corresponding disk.

> **Note:** When san_rep_device is "yes," the hdisk is configured for PPRC. The unique_id is based on the Copy Relation ID from Inquiry Page 0x83, rather than the LUN ID Descriptor from Inquiry Page 0x80.
>
> The `san_rep_cfg` attribute determines what types of devices are configured as HyperSwap disks:
>
> | | |
> |---|---|
> | `none` = [DEFAULT] | Devices are not to be configured for PPRC. |
> | `revert_disk` | The selected hdisk is to be reverted to not PPRC-configured and keeps its existing name. The **`-U`** switch used along with the **`revert_disk`** parameter in the **`chdev`** command allows you to maintain the hdisk name while is reverted but for the secondary device. |
> | `migrate_disk` | The selected hdisk is to be converted to PPRC-configured and keeps its existing name. |
> | `new` | Newly defined hdisks will be configured as PPRC, if capable. |
> | `new_and_existing` | New and existing hdisks will be configured as PPRC, if capable. Existing hdisks will have a new logical hdisk instance name, and the previous hdisk name will remain in *Defined* state. |

In Example 8-11 on page 231, hdisk31 is replicated by Metro Mirror to hdisk71. This pair of disks is configured for HyperSwap on AIX. The storage disk membership is shown in Table 8-2.

Table 8-2 hdisk31 and hdisk71 storage membership

Storage device	hdisk ID	Volume ID
IBM.2107-75TL771	hdisk31	4404
IBM.2107-75LY981	hdisk71	0004

We first verify the replication status on the storage side as shown in Example 8-11.

Example 8-11 Replication status for hdisk31 and hisk71 on the storage side

```
lspprc -fullid -l 4404
Date/Time: November 25, 2013 4:44:54 PM CST IBM DSCLI Version: 6.6.0.305 DS: IBM.2107-75TL771
ID                                         State       Reason Type         Out Of Sync Tracks
Tgt Read Src Cascade Tgt Cascade Date Suspended SourceLSS          Timeout (secs) Critical Mode
First Pass Status Incremental Resync Tgt Write GMIR CG PPRC CG  isTgtSE DisableAutoResync
================================================================================================
================================================================================================
=================================================================================
IBM.2107-75TL771/4404:IBM.2107-75LY981/0004 Full Duplex -      Metro Mirror 0
Disabled Disabled    Invalid     -             IBM.2107-75TL771/44 60            Disabled
Invalid          Disabled           Disabled  N/A     Disabled Unknown -
```

Next, verify the corresponding disk attributes, as shown in Example 8-12.

Example 8-12 Disk attributes before transforming into HyperSwap capable disks

```
root@r6r4m51:/testhyp> lsattr -El hdisk31 |egrep 'reserve_policy|san_rep_cfg|san_rep_device'
reserve_policy  no_reserve Reserve Policy                             True+
san_rep_cfg     none SAN Replication Device Configuration Policy     True+
san_rep_device  detected SAN Replication Device                      False
root@r6r4m51:/testhyp> lsattr -El hdisk71 |egrep 'reserve_policy|san_rep_cfg|san_rep_device'
reserve_policy  no_reserve Reserve Policy                            True+
san_rep_cfg     none SAN Replication Device Configuration Policy     True+
san_rep_device  detected SAN Replication Device                      False
```

Using the **lspprc** command at the AIX operating system level, verify the disks so that you know exactly what is the current information regarding Peer-to-Peer Remote Copy status, as shown in Example 8-13.

Example 8-13 Display information about PPRC disks path status

```
root@r6r4m51:/testhyp> lspprc -p hdisk31
path       WWNN              LSS   VOL    path
group id                                  group status
=======================================================
0(s)       5005076308ffc6d4  0x00  0x04    PRIMARY
-1         500507630affc16b  0x44  0x04

path      path  path       parent  connection
group id  id    status
=====================================================================
     0    0     Enabled    fscsi0  50050763085046d4,4000400400000000
     0    1     Enabled    fscsi0  50050763085006d4,4000400400000000
     0    2     Enabled    fscsi1  50050763085046d4,4000400400000000
     0    3     Enabled    fscsi1  50050763085006d4,4000400400000000
root@r6r4m51:/testhyp> lspprc -p hdisk71
path       WWNN              LSS   VOL    path
group id                                  group status
=======================================================
0(s)       500507630affc16b  0x44  0x04    SECONDARY
-1         5005076308ffc6d4  0x00  0x04

path      path  path       parent  connection
```

```
group id  id    status
====================================================================
     0    0     Enabled    fscsi0  500507630a03416b,4044400400000000
     0    1     Enabled    fscsi0  500507630a03016b,4044400400000000
     0    2     Enabled    fscsi1  500507630a03416b,4044400400000000
     0    3     Enabled    fscsi1  500507630a03016b,4044400400000000
```

Configure hdisk31for HyperSwap as the principal (source) disk, as shown in Example 8-14.

Example 8-14 Configuring hdisk31 as a HyperSwap disk

```
root@r6r4m51:/testhyp> lspv |grep -E 'hdisk31|hdisk71'
hdisk31         00cdb3117a5b1485                    itsovg          active
hdisk71         none                                None
root@r6r4m51:/testhyp> chdev -l hdisk31 -a san_rep_cfg=migrate_disk -U
hdisk31 changed
root@r6r4m51:/testhyp> lspprc -v hdisk31
HyperSwap lun unique
identifier.............35203735544c3737313434303400525bf2bb07210790003IBMfcp

hdisk31 Primary         MPIO IBM 2107 FC Disk

        Manufacturer..................IBM
        Machine Type and Model........2107900
        ROS Level and ID.............2E313336
        Serial Number.................75LY9810
        Device Specific.(Z7)..........0004
        Device Specific.(Z0)..........000005329F101002
        Device Specific.(Z1)..........004
        Device Specific.(Z2)..........075
        Unique Device Identifier......200B75LY981000407210790003IBMfcp
        Logical Subsystem ID..........0x00
        Volume Identifier.............0x04
        Subsystem Identifier(SS ID)...0xFF00
        Control Unit Sequence Number..00000LY981
        Storage Subsystem WWNN........5005076308ffc6d4
        Logical Unit Number ID........4000400400000000

hdisk31 Secondary       MPIO IBM 2107 FC Disk

        Manufacturer..................IBM
        Machine Type and Model........2107900
        ROS Level and ID.............2E393330
        Serial Number.................75TL7714
        Device Specific.(Z7)..........4404
        Device Specific.(Z0)..........000005329F101002
        Device Specific.(Z1)..........404
        Device Specific.(Z2)..........075
        Unique Device Identifier......200B75TL771440407210790003IBMfcp
        Logical Subsystem ID..........0x44
        Volume Identifier.............0x04
        Subsystem Identifier(SS ID)...0xFF44
        Control Unit Sequence Number..00000TL771
        Storage Subsystem WWNN........500507630affc16b
```

```
        Logical Unit Number ID........4044400400000000
```

After configuring the disk for HyperSwap, hdisk71 has the status of *Defined* and only hdisk31 is available. The **lspprc** command indicates the paths for the HyperSwap disk, as shown Example 8-15.

Example 8-15 HyperSwap disk configuration

```
root@r6r4m51:/testhyp> lspv |grep -E 'hdisk31|hdisk71'
hdisk31         00cdb3117a5b1485                    itsovg          active

root@r6r4m51:/testhyp> lspprc -p hdisk31
path       WWNN              LSS   VOL    path
group id                                  group status
=======================================================
0(s)       5005076308ffc6d4  0x00  0x04    PRIMARY
1          500507630affc16b  0x44  0x04    SECONDARY

path      path  path       parent  connection
group id  id    status
=====================================================================
     0    0     Enabled    fscsi0  50050763085046d4,4000400400000000
     0    1     Enabled    fscsi0  50050763085006d4,4000400400000000
     0    2     Enabled    fscsi1  50050763085046d4,4000400400000000
     0    3     Enabled    fscsi1  50050763085006d4,4000400400000000
     1    4     Enabled    fscsi0  500507630a03416b,4044400400000000
     1    5     Enabled    fscsi0  500507630a03016b,4044400400000000
     1    6     Enabled    fscsi1  500507630a03416b,4044400400000000
     1    7     Enabled    fscsi1  500507630a03016b,4044400400000000
```

Note: At any time, only one of the two path groups is selected for I/O operations to the hdisk. The selected path group is identified in the output by (s).

At this time, the HyperSwap disk configuration has been performed without unmounting file systems or stopping the application.

The migrating disk should be the primary disk when it is migrated to the HyperSwap disk. Otherwise, if the auxiliary disk is chosen instead of the primary, and the primary disk is part of a volume group, the message from Example 8-16 appears.

Example 8-16 Choosing hdisk71 as migrated disk for HyperSwap instead of hdisk31

```
root@r6r4m51:/testhyp> chdev -l hdisk71 -a san_rep_cfg=migrate_disk -U
Method error (/usr/lib/methods/chgdisk):
        0514-062 cannot perform the requested function because the
                 specified device is busy.
```

Important: If the primary disk does not belong to any volume group or the volume group is varied off, the **chdev** command succeeds for the auxiliary disk (PPRC target). In this case, even if the PPRC replication direction is reversed on the storage side, on the AIX operating system, the disk is not seen with the required information. The entire process for migrating disk should be redone.

This is *not* a recommended method for enabling HyperSwap by using a secondary disk.

When a disk or group of disks are swapped to auxiliary storage and the primary storage is lost, HyperSwap reroutes the I/O to secondary storage. If we cannot recover the hdisk configured initially as primary, now lost, we can maintain the disk configuration in terms of hdisk number at operating system level. In other words, the auxiliary disk and primary disk can be reversed as hdisk number at the operating system level, and the hdisk source will be at this time on auxiliary storage system. This configuration is valuable when storage migration is performed or a primary storage reconfiguration is required. We show this behavior with another pair of disks with storage membership as described in Table 8-3.

Table 8-3 hdisk45 and hdisk10 storage membership

Storage device	hdisk	Volume ID
IBM.2107-75TL771	hdisk45	a004
IBM.2107-75LY981	hdisk10	1004

We determine the hdisk45 configuration at the operating system level, as shown in Example 8-17.

Example 8-17 hdisk45 configuration

```
root@r6r4m51:/work> lspprc -p hdisk45
path       WWNN              LSS   VOL    path
group id                                  group status
=======================================================
0          5005076308ffc6d4  0x10  0x04    SECONDARY
1(s)       500507630affc16b  0xa0  0x04    PRIMARY

path      path  path       parent  connection
group id  id    status
=====================================================================
     0    0     Enabled    fscsi0  50050763085046d4,4010400400000000
     0    1     Enabled    fscsi0  50050763085006d4,4010400400000000
     0    2     Enabled    fscsi1  50050763085046d4,4010400400000000
     0    3     Enabled    fscsi1  50050763085006d4,4010400400000000
     1    4     Enabled    fscsi0  500507630a03416b,40a0400400000000
     1    5     Enabled    fscsi0  500507630a03016b,40a0400400000000
     1    6     Enabled    fscsi1  500507630a03416b,40a0400400000000
     1    7     Enabled    fscsi1  500507630a03016b,40a0400400000000

dscli> lspprc -fullid -l a004
Date/Time: December 19, 2013 2:42:03 AM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75TL771
ID                                         State       Reason Type         Out Of
Sync Tracks Tgt Read Src Cascade Tgt Cascade Date Suspended SourceLSS
Timeout (secs) Critical Mode First Pass Status Incremental Resync Tgt Write GMIR
CG PPRC CG  isTgtSE DisableAutoResync
==================================================================================
==================================================================================
==================================================================================
===================================
IBM.2107-75TL771/A004:IBM.2107-75LY981/1004 Full Duplex -    Metro Mirror 0
Disabled Disabled    Invalid     -             IBM.2107-75TL771/A0 60
Disabled      Invalid           Disabled           Disabled  N/A     Disabled
Unknown
```

The disk with volume ID 1004 is hdisk10 is in the *Defined* state, as shown in Example 8-18.

Example 8-18 hdisk10 configuration

```
root@r6r4m51:/work> lsdev -Cc disk |grep hdisk10
hdisk10 Defined   07-08-02      MPIO IBM 2107 FC Disk
root@r6r4m51:/work> lscfg -vpl hdisk10
  hdisk10        U7311.D20.10135EC-P1-C02-T1-W50050763085046D4-L4010400400000000
MPIO IBM 2107 FC Disk

        Manufacturer................IBM
        Machine Type and Model......2107900
        Part Number.................
        ROS Level and ID............2E313336
        Serial Number...............75LY9811
...........................<<snipped text>>.............................
        Device Specific.(Z7)........1004
```

We swap hdisk45 to auxiliary storage, and unconfigure it as HperSwap disk by using the **chdev** command, as shown in Example 8-19. At the end, the hdisk is configured as hdisk45 and reverted disk but on auxiliary storage.

Example 8-19 Swapping hdisk

Hdisk45 is swapped

```
dscli> lspprc -fullid -l 1004
Date/Time: December 19, 2013 2:54:13 AM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75LY981
ID                                          State             Reason Type
Out Of Sync Tracks Tgt Read Src Cascade Tgt Cascade Date Suspended SourceLSS
Timeout (secs) Critical Mode First Pass Status Incremental Resync Tgt Write GMIR
CG PPRC CG isTgtSE DisableAutoResync
==================================================================================
==================================================================================
==================================================================================
=========================================
IBM.2107-75TL771/A004:IBM.2107-75LY981/1004 Target Full Duplex -      Metro Mirror
0                  Disabled Invalid     Disabled    -
IBM.2107-75TL771/A0 unknown       Disabled     Invalid          Disabled
Disabled  N/A     N/A     Unknown -
dscli> failoverpprc -remotedev  IBM.2107-75TL771 -type mmir 1004:a004
Date/Time: December 19, 2013 2:54:46 AM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75LY981
CMUC00196I failoverpprc: Remote Mirror and Copy pair 1004:A004 successfully
reversed.
dscli> failbackpprc -remotedev  IBM.2107-75TL771 -type mmir 1004:a004
Date/Time: December 19, 2013 2:55:06 AM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75LY981
CMUC00197I failbackpprc: Remote Mirror and Copy pair 1004:A004 successfully failed
back.
```

Hdisk45 has the source on the auxiliary storage.

```
root@r6r4m51:/work> lspprc -p hdisk45
path       WWNN               LSS   VOL    path
group id                                   group status
```

```
====================================================
0(s)       5005076308ffc6d4  0x10  0x04    PRIMARY
1          500507630affc16b  0xa0  0x04    SECONDARY

path      path  path       parent  connection
group id  id    status
======================================================================
     0    0     Enabled    fscsi0  50050763085046d4,4010400400000000
     0    1     Enabled    fscsi0  50050763085006d4,4010400400000000
     0    2     Enabled    fscsi1  50050763085046d4,4010400400000000
     0    3     Enabled    fscsi1  50050763085006d4,4010400400000000
     1    4     Enabled    fscsi0  500507630a03416b,40a0400400000000
     1    5     Enabled    fscsi0  500507630a03016b,40a0400400000000
     1    6     Enabled    fscsi1  500507630a03416b,40a0400400000000
     1    7     Enabled    fscsi1  500507630a03016b,40a0400400000000
```

We revert the disk hdisk45 using -U switch.

```
root@r6r4m51:/work> chdev -l hdisk45 -a san_rep_cfg=revert_disk -U
hdisk45 changed
root@r6r4m51:/work> lspprc -v hdisk45

HyperSwap lun unique identifier.............200B75LY981100407210790003IBMfcp

hdisk45 Primary         MPIO IBM 2107 FC Disk

        Manufacturer..................IBM
        Machine Type and Model........2107900
        ROS Level and ID..............2E313336
        Serial Number.................75LY9811
        Device Specific.(Z7)..........1004
        Device Specific.(Z0)..........000005329F101002
        Device Specific.(Z1)..........004
        Device Specific.(Z2)..........075
        Unique Device Identifier......200B75LY981100407210790003IBMfcp
        Logical Subsystem ID..........0x10
        Volume Identifier.............0x04
        Subsystem Identifier(SS ID)...0xFF10
        Control Unit Sequence Number..00000LY981
        Storage Subsystem WWNN........5005076308ffc6d4
        Logical Unit Number ID........4010400400000000
root@r6r4m51:/work> lspprc -v hdisk10
Invalid device name hdisk10

root@r6r4m51:/work> cfgmgr

Now the hdisk10 is on the primary storage.
root@r6r4m51:/work> lspprc -v hdisk10

HyperSwap lun unique identifier.............200B75TL771A00407210790003IBMfcp

hdisk10 Secondary       MPIO IBM 2107 FC Disk

        Manufacturer..................IBM
```

```
Machine Type and Model........2107900
ROS Level and ID.............2E393330
Serial Number.................75TL771A
Device Specific.(Z7)..........A004
Device Specific.(Z0)..........000005329F101002
Device Specific.(Z1)..........004
Device Specific.(Z2)..........075
Unique Device Identifier......200B75TL771A00407210790003IBMfcp
Logical Subsystem ID.........0xa0
Volume Identifier.............0x04
Subsystem Identifier(SS ID)...0xFFA0
Control Unit Sequence Number..00000TL771
Storage Subsystem WWNN........500507630affc16b
Logical Unit Number ID........40a0400400000000
```

In some cases, if the HyperSwap is enabled on the disk that is the target on Metro Mirror replication, the disk is not usable for HyperSwap. To be HyperSwap functional, you must set up its `revert_disk` attribute and then follow the procedure for activating HyperSwap on the primary disk again.

8.14 Node-level unmanage mode

Starting with PowerHA SystemMirror 7.1.3, HyperSwap mirror groups can be reconfigured while the resource groups are in *Unmanaged state*.

This is an important feature, because you can reconfigure HyperSwap mirror groups by adding or removing disks in the mirror group configuration. It is no longer necessary to bring the resources offline while the HyperSwap mirror group is configured, as in the previous version of PowerHA SystemMirror 7.1.2 Enterprise Edition.

Node-level unmanage mode is the main feature used when mirror group reconfiguration is required.

Follow these steps for adding or replacing new disks in a resource group protected by HyperSwap:

1. Configure new disks for HyperSwap to have the same Metro Mirror replication direction.
2. Stop PowerHA system services, and leave resource groups in an Unmanaged state.
3. Modify the mirror groups configuration by adding or removing new disks.
4. Configure the corresponding resource groups to reflect the configuration of the new mirror group definition.
5. Start PowerHA SystemMirror services, leaving the resource groups in *Unmanaged state*.
6. Verify and synchronize the cluster configuration.
7. Bring resource groups online.
8. Validate the configuration.

Adding and removing disks is shown in the Oracle Node HyperSwap and Oracle RAC active-active configuration in 8.20.2, “Adding new disks to the ASM configuration: Oracle RAC HyperSwap” on page 294.

8.15 Single-node HyperSwap deployment

Single-node HyperSwap deployment consists of one PowerHA SystemMirror configuration with only a single PowerHA SystemMirror cluster node. Single-node HyperSwap offers storage errors protection for just one cluster node, using PowerHA SystemMirror for HyperSwap events handling.

Single-node HyperSwap configuration is available starting with PowerHA SystemMirror Enterprise Edition 7.1.3.

The requirements for single-node HyperSwap deployment are the same as the requirements for single node in multi-node cluster deployment:

- DS88xx and higher with minimum microcode level 86.30.49.0 or higher
- AIX version 6.1 TL9 or AIX 71 TL3
- PowerHA SystemMirror 7.1.3
- FC, FCoE, NPIV are only supported host attachment connection

Functions:

- It offers storage protection in case of a primary storage failure.
- It is configured when delivered, so it does not require a second node to form a cluster.

Limitations:

- Extending single-node HyperSwap cluster by adding other nodes in configuration is not possible, because since the sites cannot be added in single-node HyperSwap mode. The entire cluster configuration should be performed after the cluster-wide policies are set to disabled. In this case, the single-node HyperSwap configuration is lost.
- While a node is configured for a single-node HyperSwap, it cannot be added as a node into another stretched or linked cluster.
- This does not provide protection for node failure, because there is only one node.

Figure 8-5 shows a logical diagram of a single-node HyperSwap configuration.

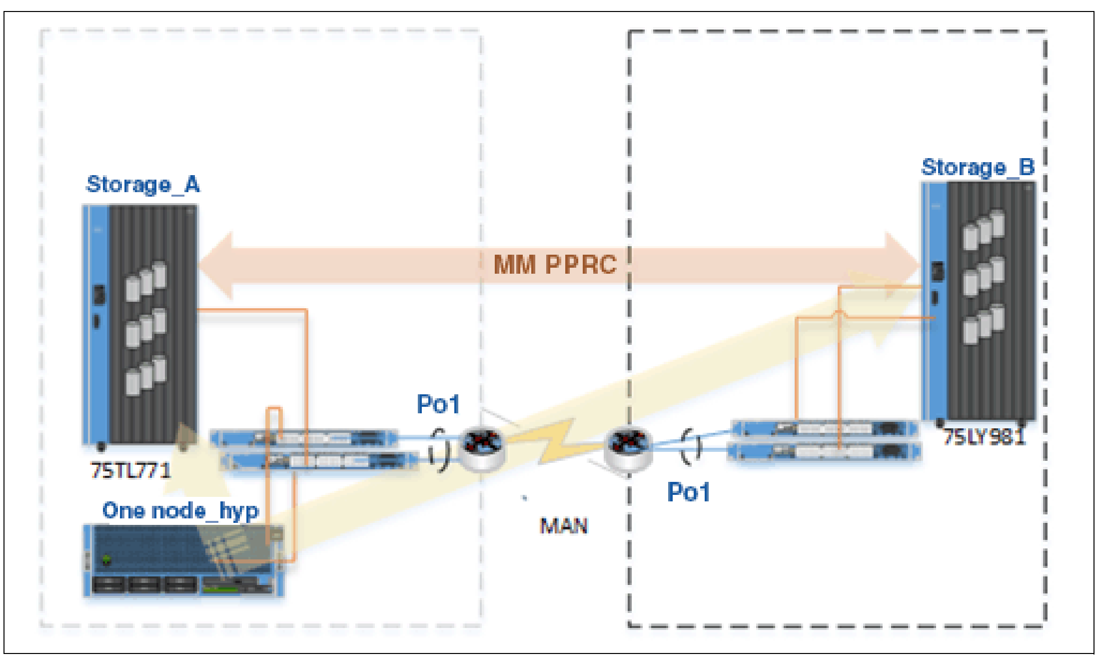


Figure 8-5 Single-node HyperSwap

8.15.1 Single-node HyperSwap configuration steps

These are the steps required for a single-node HyperSwap configuration:

1. Configure the host for HyperSwap function as described in 8.10, "HyperSwap storage configuration for PowerHA node cluster" on page 225.
2. Validate the HyperSwap configuration for the disks that will be protected in case of storage failure.
3. Set up the cluster, node, and networks on the target node.
4. Configure Cluster-Wide Policies to be `ENABLED` (the default option is set to *enabled*) in the configure DS8000 Metro Mirror in-band resources section.
5. Associate storage systems with the corresponding sites (based on Cluster-Wide Policies activation, the sites are created but are not shown in sites menu).
6. Define mirror groups and associate the desired raw disks or volume groups with them.
7. Define resource groups, including mirror group relationships, in their respective configurations.
8. Add configurations for the application controllers.
9. Verify and synchronize the cluster configuration.
10. Start services and bring the resources online.
11. Validate the configuration.

8.15.2 Oracle single-instance database with Automatic Storage Management in single-node HyperSwap

In this example, we configure a single-node HyperSwap cluster to protect an Oracle single-instance database that has database files on disks that are managed by automatic storage management. The examples described here cover the following operations and scenarios:

- Adding and replacing new disks in an Automatic Storage Management (ASM) configuration
- Planned HyperSwap for user mirror group disks
- Storage migration on single-node HyperSwap
- Unplanned HyperSwap for user mirror group disks
- Migration of a repository disk to a HyperSwap-enabled disk
- Migration of the rootvg disk to a HyperSwap-enabled disk

The Oracle single-node database installation has a grid infrastructure and a database home directory on the file systems that are created on a volume group, with disks that are configured for HyperSwap.

The destination of the database's data files are the raw disks that are managed by the ASM configuration and HyperSwap.

Note: Swap operations are performed based on mirror group configuration. The disks that have the same LSS on the storage system must be configured on the same mirror group.

In a single-node HyperSwap deployment, an active-active storage system configuration is possible by using multiple mirror groups. But it is mostly used for storage load balancing for read operations, because there is not a second node or site for server high availability and disaster recovery.

The disks were configured in the PowerHA SystemMirror node. Their designated roles are shown in Table 8-4. The procedure for configuring a disk for the HyperSwap environment is described in 8.13, “Configure disks for the HyperSwap environment” on page 229.

Table 8-4 Disk configurations in one HyperSwap node r6r4m51

ORACLE_BASE, User MG (mirror group)	ASM, User MG	System MG (rootvg)	Cluster repo MG
oravg (hdisk31)	hdisk41 hdisk61 hdisk63	hdisk30	hdisk4 (00cdb3119ad0e49a)

In this example, we use a single files system for GRID HOME and Database HOME created on the *oravg* volume group with the `/u01` and `ORACLE_BASE /u01/app/oracle` mount point. ASM uses the hdisk41, hdisk61, and hdisk63 disks.

Configuring disks for HyperSwap requires careful planning regarding the LSSes used, the replication direction for the HyperSwap disks, and which disk on the system is configured as the primary disk when `migrate_disk` and `no_reserve_policy` attributes are set.

In Example 8-20, the HyperSwap-configured disks are shown using the `disks_hyp.sh` script for a quick view of the HyperSwap pair disks.

Example 8-20 Configure HyperSwap disks used on the r6r4m51node

```
root@r6r4m51:/work> ./disks_hyp.sh HYP |egrep 'hdisk31|hdisk61|hdisk63|hdisk41'
Disk_Nr StgPRSN  StgSCSN PRID SECID
hdisk31 75TL771  75LY981 4404 0004
hdisk41 75TL771  75LY981 5204 0E04
hdisk61 75TL771  75LY981 B207 E204
hdisk63 75TL771  75LY981 B406 E700
root@r6r4m51:/work> lspprc -Ao |egrep 'hdisk31|hdisk61|hdisk63|hdisk41'
hdisk31   Active   0(s)        1            500507630affc16b  5005076308ffc6d4
hdisk41   Active   0(s)        1            500507630affc16b  5005076308ffc6d4
hdisk61   Active   0(s)        1            500507630affc16b  5005076308ffc6d4
hdisk63   Active   0(s)        1            500507630affc16b  5005076308ffc6d4
```

The storage device ID in Site A is `75TL771`, and the storage device ID on Site B is `75LY981`. The replication direction is from Storage A to Storage B for all Metro Mirror replicated disks.

We define the cluster as shown in Example 8-21.

Example 8-21 Cluster configuration for single-node HyperSwap

```
Set up a Cluster, Nodes and Networks

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

```
                                                    [Entry Fields]
* Cluster Name                                     [one_node_hyperswap]
  New Nodes (via selected communication paths)      []                            +
  Currently Configured Node(s)                      r6r4m51.austin.ibm.com


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

The cluster has been created, and the output is shown in Example 8-22.

Example 8-22 Single-node HyperSwap cluster

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

Before command completion, additional instructions may appear below.

[TOP]
Cluster Name:    one_node_hyperswap
Cluster Type:    Stretched
Heartbeat Type:  Unicast
Repository Disk: None
Cluster IP Address: None

There are 1 node(s) and 1 network(s) defined

NODE r6r4m51:
        Network net_ether_01
                r6r4m51 9.3.207.109

No resource groups defined
Initializing..
Gathering cluster information, which may take a few minutes...
Processing...
Storing the following information in file
/usr/es/sbin/cluster/etc/config/clvg_config

r6r4m51:
Hdisk:          hdisk0
PVID:           00cdb3119e416dc6
[MORE...428]
F1=Help                 F2=Refresh              F3=Cancel
F6=Command
F8=Image                F9=Shell                F10=Exit
/=Find
n=Find Next
```

We create the repository disk on hdisk4, which is not a HyperSwap disk in the first phase. The disk attributes values are shown in Example 8-23 on page 242. The `reserve_policy` attribute is also set as `no_reserve`.

Example 8-23 hdisk 4 disk attributes

```
root@r6r4m51:/> lsattr -El hdisk4
DIF_prot_type   none                             T10 protection type                          False
DIF_protection  no                               T10 protection support                       True
FC3_REC         false                            Use FC Class 3 Error Recovery                True
PCM             PCM/friend/aixmpiods8k           Path Control Module                          False
PR_key_value    none                             Persistent Reserve Key Value                 True+
algorithm       fail_over                        Algorithm                                    True+
clr_q           no                               Device CLEARS its Queue on error             True
dist_err_pcnt   0                                Distributed Error Percentage                 True
dist_tw_width   50                               Distributed Error Sample Time                True
hcheck_cmd      test_unit_rdy                    Health Check Command                         True+
hcheck_interval 60                               Health Check Interval                        True+
hcheck_mode     nonactive                        Health Check Mode                            True+
location                                         Location Label                               True+
lun_id          0x4004400200000000               Logical Unit Number ID                       False
lun_reset_spt   yes                              LUN Reset Supported                          True
max_coalesce    0x40000                          Maximum Coalesce Size                        True
max_retry_delay 60                               Maximum Quiesce Time                         True
max_transfer    0x80000                          Maximum TRANSFER Size                        True
node_name       0x5005076308ffc6d4               FC Node Name                                 False
pvid            00cdb3119ad0e49a0000000000000000 Physical volume identifier                   False
q_err           yes                              Use QERR bit                                 True
q_type          simple                           Queuing TYPE                                 True
queue_depth     20                               Queue DEPTH                                  True
reassign_to     120                              REASSIGN time out value                      True
reserve_policy  no_reserve                       Reserve Policy                               True+
rw_timeout      30                               READ/WRITE time out value                    True
san_rep_cfg     none                             SAN Replication Device Configuration Policy True+
san_rep_device  supported                        SAN Replication Device                       False
scsi_id         0xa0600                          SCSI ID                                      False
start_timeout   60                               START unit time out value                    True
timeout_policy  fail_path                        Timeout Policy                               True+
unique_id       200B75LY981040207210790003IBMfcp Unique device identifier                     False
ww_name         0x50050763085046d4               FC World Wide Name                           False
```

We proceed with the repository disk definition as shown in Example 8-24.

Example 8-24 Define a repository disk and cluster IP address for a single-node HyperSwap

```
                          Define Repository Disk and Cluster IP Address

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

                                                           [Entry Fields]
* Cluster Name                                            one_node_hyperswap
* Heartbeat Mechanism                                     Unicast                 +
* Repository Disk                                        [(00cdb3119ad0e49a)]     +
  Cluster Multicast Address                              []
      (Used only for multicast heartbeat)


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

The cluster repository has been configured and verified, as shown in Example 8-25 on page 243.

Example 8-25 Repository disk details

```
root@r6r4m51:/> odmget HACMPsircol

HACMPsircol:
        name = "one_node_hyperswap_sircol"
        id = 0
        uuid = "0"
        ip_address = ""
        repository = "00cdb3119ad0e49a"
        backup_repository = ""
```

The single-node HyperSwap has the key configuration point and the activation of the cluster-wide HyperSwap policies. This activation operation brings two sites into the configuration that are defined internally and not visible in the site configuration menu.

The sites are configured automatically, using cluster-wide HyperSwap policies. The cluster definition and node selection are preliminary steps in the configuration.

Cluster-wide HyperSwap policies on a single-node HyperSwap are configured by using smitty fast path as shown in Example 8-26: **smitty cm_cfg_clstr_hypswp_polc** or **smitty sysmirror** → **Cluster Applications and Resources** → **Resources** → **Configure DS8000 Metro Mirror (In-Band) Resources** → **Cluster Wide HyperSwap Policies**.

Example 8-26 Define cluster-wide HyperSwap policies for single-node HyperSwap activation

```
                                    Define Cluster wide HyperSwap Policies

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

                                                           [Entry Fields]
  Single node HyperSwap                                    Enabled                  +



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

The field for a single-node HyperSwap is shown enabled by default. We choose to leave that default setting.

The activation procedure requires not to have sites already configured in PowerHA SystemMirror since enabling the single-node HyperSwap feature automatically adds the sites in the cluster configuration.

Activating the cluster-wide HyperSwap policies provides only the completion status of the running command. Behind, two sites only for storage site association are configured. The HACMsite ODM class is not populated and the sites are not shown as site definitions. The sites *Site1_primary* and *Site2_secondary* are used internally by the clxd daemon.

In the next steps, we configure the DS8000 Metro Mirror (in-band) resources by using this smitty fast_path to define and configure both storages systems, as shown in Example 8-27 on page 244: **smitty cm_cfg_strg_systems** or **smitty sysmirror** → **Cluster Applications and**

Resources → Resources → Configure DS8000 Metro Mirror (In-Band) Resources → Configure Storage Systems → Add a Storage System.

Example 8-27 Adding DS8K storages as Metro Mirror in-band resources

```
Add a Storage System

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

                                                          [Entry Fields]
* Storage System Name                                  [STG_A]
* Site Association                                                         +
* Vendor Specific Identifier                            IBM.2107-00000LY981       +
* WWNN                                                  500507630AFFC16B          +


    lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
    x                            Site Association                            x
    x                                                                        x
    x Move cursor to desired item and press Enter.                           x
    x                                                                        x
    x   Site1_primary                                                        x
    x   Site2_secondary                                                      x
    x                                                                        x
    x F1=Help                F2=Refresh             F3=Cancel                x
F1=Hx F8=Image               F10=Exit               Enter=Do                 x
Esc+x /=Find                 n=Find Next                                     x
F9=Smqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

In the same way, the secondary storage is added and configured, with the site association as `Site2_secondary`. Both storage systems are defined as Metro Mirror resources, as shown in Example 8-28.

Example 8-28 Query defined storage systems

```
clmgr -v query storage_system
NAME="STG_A"
TYPE="ds8k_inband_mm"
VENDOR_ID="IBM.2107-00000LY981"
WWNN="5005076308FFC6D4"
SITE="Site1_primary"
ATTRIBUTES=""

NAME="STG_B"
TYPE="ds8k_inband_mm"
VENDOR_ID="IBM.2107-00000TL771"
WWNN="500507630AFFC16B"
SITE="Site2_secondary"
ATTRIBUTES=""
root@r6r4m51:/usr/es/sbin/cluster/utilities>
```

The associated data for the storage configurations can be obtained by using the **`odmget HACMPxd_storage_system`** command, as shown Example 8-29 on page 245.

Example 8-29 Storage system configurations

```
root@r6r4m51:/etc/objrepos> odmget HACMPxd_storage_system

HACMPxd_storage_system:
        xd_storage_tech_id = 5
        xd_storage_system_id = 7
        xd_storage_system_name = "STG_A"
        xd_storage_vendor_unique_id = "IBM.2107-00000LY981"
        xd_storage_system_site_affiliation = "Site1_primary"
        xd_storage_system_wwnn = "5005076308FFC6D4"

HACMPxd_storage_system:
        xd_storage_tech_id = 5
        xd_storage_system_id = 8
        xd_storage_system_name = "STG_B"
        xd_storage_vendor_unique_id = "IBM.2107-00000TL771"
        xd_storage_system_site_affiliation = "Site2_secondary"
        xd_storage_system_wwnn = "500507630AFFC16B”
```

The next configuration step is the mirror group definition. The definition specifies the logical collection of volumes that must be mirrored to a storage system on the remote site.

The *recovery action* parameter must be set to Manual in a single-node HyperSwap environment. Also, the HyperSwap function must be enabled andthe *re-sync automatically* parameter set to *auto*.

Note: In case of volume replication or path recovery, the HyperSwap function makes sure to perform a re-sync automatically for *auto*. For *manual*, a user- recommended action would be displayed in **errpt** for a HyperSwap-enabled mirror group and in `hacmp.out` for the HyperSwap-disabled mirror group. It is best to configure a split and merge policy when using *auto*.

In this step, we add in User Mirror Group configuration to the volume group oravg and as RAW disks were selected hdisk41, hdisk61 and hdisk63.

The fast_path for accessing the configuration menu is **smitty cm_add_mirr_gps_select** choosing the desired type of mirror group. For this example, the chosen Mirror Group type is *User,* as shown in Example 8-30.

Example 8-30 User Mirror Group definition

```
                          Add a User Mirror Group

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

                                                        [Entry Fields]
* Mirror Group Name                                  [ORA_MG]
  Volume Group(s)                                     oravg
  Raw Disk(s)                                         hdisk61:3be20bb3-2aa1> +
  HyperSwap                                           Enabled                +
  Consistency Group                                   Enabled                +
  Unplanned HyperSwap Timeout (in sec)               [60]                     #
  HyperSwap Priority                                  Medium
  Recovery Action                                     Manual                 +
```

```
  Re-sync Action                                      Automatic          +

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

Unplanned HyperSwap timeout remains momentary unchanged. This value represents how long a connection remains unavailable before an unplanned HyperSwap site failover occurs.

After the Mirror Group definition, based on Metro Mirror replication direction, associated storage systems are automatically added to the Mirror Group definition as shown in Example 8-31.

Example 8-31 Mirror Group attributes

```
                    Change/Show a User Mirror Group

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

                                                        [Entry Fields]
  Mirror Group Name                                   ORA_MG
  New Mirror Group Name                              []
  Volume Group(s)                                     oravg              +
  Raw Disk(s)                                        [3be20bb3-2aa1-e421-ef> +
  Associated Storage System(s)                        STG_A STG_B        +
  HyperSwap                                           Enabled            +
  Consistency Group                                   Enabled            +
  Unplanned HyperSwap Timeout (in sec)               [60]                 #
  HyperSwap Priority                                  Medium
  Recovery Action                                     Manual             +
  Re-sync Action                                      Automatic          +

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

The resource group protected by PowerHA SystemMirror is defined as shown in Example 8-32. Since we have only single node, resource policies do not have direct implications to the single-node HyperSwap as the cluster node is the same.

Example 8-32 Resource group definition in single-node HyperSwap scenario

```
                             Add a Resource Group

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

                                                        [Entry Fields]
* Resource Group Name                                [ORARG]
* Participating Nodes (Default Node Priority)        [r6r4m51]               +

  Startup Policy                                      Online On Home Node Only   +
  Fallover Policy                                     Fallover To Next Priority> +
  Fallback Policy                                     Never Fallback             +
```

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

Further on, the resource group configuration is performed and the previous mirror group defined is also indicated in **DS8000-Metro Mirror (In-band) Resources** entry field, as shown in Example 8-33. In this step, as we proceeded for User Mirror Group definition, we select the volume group oravg, and also the raw disks hdisk41, hdisk63, and hdisk61, based on each disk's UUID.

Example 8-33 Resource group attributes for using HyperSwap Mirror Group

```
        Change/Show All Resources and 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                                  ORARG
  Participating Nodes (Default Node Priority)          r6r4m51

  Startup Policy                                       Online On First Available>
  Fallover Policy                                      Fallover To Next Priority>
  Fallback Policy                                      Never Fallback

  Service IP Labels/Addresses                         []                      +
  Application Controllers                             []                      +

  Volume Groups                                       [oravg]                +
  Use forced varyon of volume groups, if necessary     false                  +
...........................<<..snipped text..>>...................................
* Tape Resources                                      []                  +
  Raw Disk PVIDs                                      []                  +
  Raw Disk UUIDs/hdisks                               [3be20bb3-2aa1-e421> +

  DS8000 Global Mirror Replicated Resources                                   +
  XIV Replicated Resources                                                    +
  TRUECOPY Replicated Resources                       []                      +
  DS8000-Metro Mirror (In-band) Resources              ORA_MG                 +
...........................<<..snipped text..>>...................................

[BOTTOM]

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

Specifying the DS8000-Metro Mirror (In-band) Resources is required for all resource groups that have disks protected by the HyperSwap function.

Note: The raw disks are identified by UUID. Disk UUID can be obtained using `lspv -u`. Adding raw disks by UUID does not require a PVID.

If a new volume group or disks are required to be configured for HyperSwap, the definitions of Mirror Group and resource groups must be updated to reflect the new configuration, as shown in 8.16, “Dynamically adding new disk in ASM” on page 250.

The configuration of resource group ORARG is shown in Example 8-34.

Example 8-34 ORARG resource group configuration

```
root@r6r4m51:/> clshowres -g ORARG

Resource Group Name                                    ORARG
Participating Node Name(s)                             r6r4m51
Startup Policy                                         Online On First Available Node
Fallover Policy                                        Fallover To Next Priority Node In The List
Fallback Policy                                        Never Fallback
Site Relationship                                      ignore
...........................................<<snipped text>>.............................
Volume Groups                                          oravg
Concurrent Volume Groups
Use forced varyon for volume groups, if necessary      false
Disks
Raw Disks
                      3be20bb3-2aa1-e421-ef06-fc9877cf486f 6a56cac1-2d
4a-912a-0d17-d702e32ca52a 7d9ddb03-4c8c-9219-a461-0aa2fac14388
Disk Error Management?
...........................................<<snipped text>>.............................

GENERIC XD Replicated Resources                        ORA_MG
Node Name                                              r6r4m51
Debug Level                                            high
Format for hacmp.out                                   Standard
```

The DS8000-Metro Mirror (In-band) Resources field is not automatically populated, even if the volume groups are already part of a resource group while you indicate the disks or volume groups that are protected by HyperSwap.

We proceed with verifying and cluster synchronization, and then we start the PowerHA SystemMirror services.

> **Note:** During the verify and synchronize step, this message appears:
>
> Mirror Group “ORA_MG” has the Recovery Action set to “manual.” In case of a site outage, this resource will not be automatically failed-over, and a manual intervention will be required to resolve the situation and bring the RG online on the secondary site.
>
> In a single-node HyperSwap configuration, the warning message can be ignored.

Because we want to have more flexibility for disk management and not be dependent of an exact disk location on the system, the ASM disks are configured to use a special file that is created with the **mknod** command, as shown in Example 8-35 on page 249. Also, the required permissions were added on the corresponding raw disks.

Example 8-35 Configuring RAW devices wit mknod command

```
root@r6r4m51:/>  lspv -u |egrep 'hdisk31|hdisk41|hdisk61|hdisk63'
hdisk31         00cdb3117a5b1485                    oravg           concurrent
35203735544c3737313434303400525bf2bb07210790003IBMfcp
bbccfcb9-2466-ba9b-071b-882418eefc84
hdisk41         00cdb3117a24e5e0                    None
35203735544c373731353230340051c9408407210790003IBMfcp
7d9ddb03-4c8c-9219-a461-0aa2fac14388
hdisk61         00cdb3117a5aa6e7                    None
35203735544c373731423230370051d6c22507210790003IBMfcp
3be20bb3-2aa1-e421-ef06-fc9877cf486f
hdisk63         00cdb3117a5a9f21                    None
35203735544c373731423430360051cd25f307210790003IBMfcp
6a56cac1-2d4a-912a-0d17-d702e32ca52a

root@r6r4m51:/> ls -l /dev/ |grep hdisk41
brw-------    1 root     system       21, 39 Nov 22 12:18 hdisk41
crw-------    1 root     system       21, 39 Nov 22 12:18 rhdisk41
root@r6r4m51:/> ls -l /dev/ |grep hdisk61
brw-------    1 root     system       21, 59 Nov 22 12:18 hdisk61
crw-------    1 root     system       21, 59 Nov 22 12:18 rhdisk61
root@r6r4m51:/> ls -l /dev/ |grep hdisk63
brw-------    1 root     system       21, 60 Nov 22 12:18 hdisk63
crw-------    1 root     system       21, 60 Nov 22 12:18 rhdisk63
root@r6r4m51:/> mknod /dev/asm_disk1  c 21 39
root@r6r4m51:/> mknod /dev/asm_disk2  c 21 59
root@r6r4m51:/> mknod /dev/asm_disk3  c 21 60
root@r6r4m51:/> chown oracle:oinstall /dev/rhdisk41
root@r6r4m51:/> chown oracle:oinstall /dev/rhdisk61
root@r6r4m51:/> chown oracle:oinstall /dev/rhdisk63
root@r6r4m51:/> chown oracle:oinstall /dev/asm_disk*
root@r6r4m51:/> chmod 660 /dev/asm_disk*
root@r6r4m51:/> chmod 660 /dev/rhdisk41
root@r6r4m51:/> chmod 660 /dev/rhdisk61
root@r6r4m51:/> chmod 660 /dev/rhdisk63
```

Now, it is time to install the Oracle single-instance database with *rmed* database data files configured to use AMS disk group +DATA. The database resource is shown configured in Example 8-36.

See “Oracle Grid Infrastructure for a Standalone Server” on the Oracle Database Installation Guide web page for details about installation and configuration of Oracle single-instance database:

http://docs.oracle.com/cd/E11882_01/install.112/e24321/oraclerestart.htm

Example 8-36 ASM Disk Group

```
$ . oraenv
ORACLE_SID = [grid] ? +ASM
The Oracle base has been set to /u01/app/grid
$ asmcmd
ASMCMD> lsdsk
Path
/dev/asm_disk1
/dev/asm_disk2
```

```
/dev/asm_disk3
ASMCMD> lsdg
State    Type    Rebal  Sector  Block       AU  Total_MB  Free_MB  Req_mir_free_MB
Usable_file_MB  Offline_disks  Voting_files  Name
MOUNTED  NORMAL  N         512   4096  1048576     15360    15181           5120
5030             0             N  DATA/

$ crs_stat -t
Name           Type           Target    State     Host
------------------------------------------------------------
ora.DATA.dg    ora....up.type ONLINE    ONLINE    r6r4m51
ora....ER.lsnr ora....er.type ONLINE    ONLINE    r6r4m51
ora.asm        ora.asm.type   ONLINE    ONLINE    r6r4m51
ora.cssd       ora.cssd.type  ONLINE    ONLINE    r6r4m51
ora.diskmon    ora....on.type OFFLINE   OFFLINE
ora.evmd       ora.evm.type   ONLINE    ONLINE    r6r4m51
ora.ons        ora.ons.type   OFFLINE   OFFLINE
```

We also create and configure the itsodb database, which has database files on ASM disk group +DATA. Available resources are shown in Example 8-37.

Example 8-37 Status of resources

```
crs_stat -t
Name           Type           Target    State     Host
------------------------------------------------------------
ora.DATA.dg    ora....up.type ONLINE    ONLINE    r6r4m51
ora....ER.lsnr ora....er.type ONLINE    ONLINE    r6r4m51
ora.asm        ora.asm.type   ONLINE    ONLINE    r6r4m51
ora.itsodb.db  ora....se.type ONLINE    ONLINE    r6r4m51
ora.cssd       ora.cssd.type  ONLINE    ONLINE    r6r4m51
ora.diskmon    ora....on.type OFFLINE   OFFLINE
ora.evmd       ora.evm.type   ONLINE    ONLINE    r6r4m51
ora.ons        ora.ons.type   OFFLINE   OFFLINE
```

8.16 Dynamically adding new disk in ASM

Adding a new ASM configuration requires following this procedure:

1. Allocate disks from both storage repositories to cluster nodes by following LSS planning.
2. Replicate the disks by using Metro Mirror Peer-to- Peer Copy Services.
3. Configure the disks at the AIX level to be HyperSwap-enabled disks.
4. Use the HyperSwap Unmanage mode function, and stop cluster services, leaving the resource group (RG) in Unmanaged mode.
5. Start PowerHA services, leaving the RG in Unmanaged state.
6. Add the disks to the existing mirror group.
7. Add new disks to the existing RG.
8. Verify and synchronize.
9. Bring the RGs online.

10. Verify logs and the cluster resources status.

Before the new ASM disk addition, we start the database loading by using Swingbench and execute the procedure for inserting data into the table. The load runs continuously until the reconfiguration is finished.

The Swingbench workload starts at 14:24, continuing until the new disk is added. The entire load is shown in Figure 8-6.

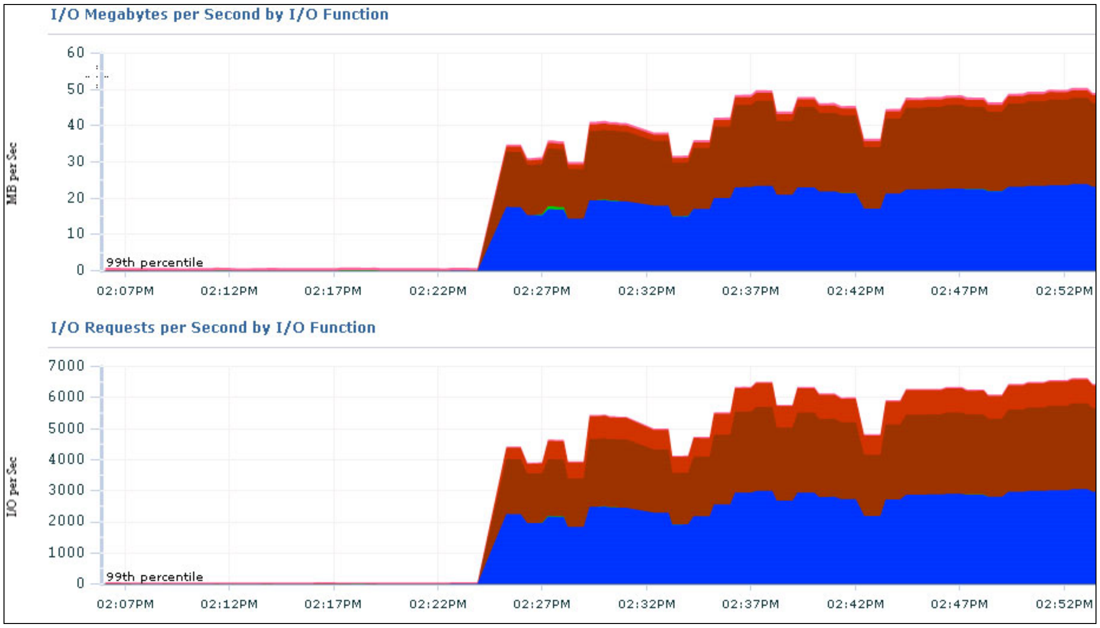


Figure 8-6 I/O Megabytes and I/O Requests per second during Swingbench load

We introduce to the existing configuration a new disk, hdisk42, which has the same Metro Mirror replication direction, as shown in Example 8-38. First, the disk is configured with the same permissions required by ASM. We also create the special pseudo device file by using the **mknod** command.

Example 8-38 Configuring a new disk for ASM

```
root@r6r4m51:/> lspprc -p hdisk42
path       WWNN              LSS   VOL    path
group id                                  group status
=======================================================
0(s)       500507630affc16b  0x52  0x05    PRIMARY
1          5005076308ffc6d4  0x0e  0x05    SECONDARY

path       path  path       parent  connection
group id   id    status
=====================================================================
     0     0     Enabled    fscsi0  500507630a03416b,4052400500000000
     0     1     Enabled    fscsi0  500507630a03016b,4052400500000000
     0     2     Enabled    fscsi1  500507630a03416b,4052400500000000
     0     3     Enabled    fscsi1  500507630a03016b,4052400500000000
     1     4     Enabled    fscsi0  50050763085046d4,400e400500000000
     1     5     Enabled    fscsi0  50050763085006d4,400e400500000000
     1     6     Enabled    fscsi1  50050763085046d4,400e400500000000
     1     7     Enabled    fscsi1  50050763085006d4,400e400500000000
```

```
root@r6r4m51:/> lspprc -Ao |egrep 'hdisk31|hdisk61|hdisk63|hdisk41|hdisk42'
hdisk31   Active   0(s)        1          500507630affc16b  5005076308ffc6d4
hdisk41   Active   0(s)        1          500507630affc16b  5005076308ffc6d4
hdisk42   Active   0(s)        1          500507630affc16b  5005076308ffc6d4
hdisk61   Active   0(s)        1          500507630affc16b  5005076308ffc6d4
hdisk63   Active   0(s)        1          500507630affc16b  5005076308ffc6d4

root@r6r4m51:/> chown oracle:oinstall /dev/rhdisk42
root@r6r4m51:/> mknod /dev/asm_disk4 c 21 40
root@r6r4m51:/> chown oracle:oinstall /dev/asm_disk4
root@r6r4m51:/> chmod 660 /dev/asm_disk4
root@r6r4m51:/> chmod 660 /dev/rhdisk42
```

We monitor the database alert log file and ASM log file to make sure that they are visible during the disk addition, as shown in Example 8-39.

Example 8-39 Monitoring the logs

```
$ tail -f /u01/app/oracle/diag/rdbms/itsodb/itsodb/trace/alert_itsodb.log
database for recovery-related files, and does not reflect the amount of
space available in the underlying filesystem or ASM diskgroup.
Wed Dec 18 13:11:47 2013
Starting background process CJQ0
Wed Dec 18 13:11:47 2013
CJQ0 started with pid=30, OS id=13369580
Wed Dec 18 13:21:46 2013
Starting background process SMCO
Wed Dec 18 13:21:46 2013
SMCO started with pid=21, OS id=15794212
```

We stop PowerHA services and leave the resource group in Unmanaged state, as shown in Example 8-40.

Example 8-40 Leave RG in Unmanaged state

```
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                    +
  Stop Cluster Services on these nodes               [r6r4m51]               +
  BROADCAST cluster shutdown?                         false                  +
* Select an Action on Resource Groups                 Unmanage Resource Gro> +


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

root@r6r4m51:/> clRGinfo -p

Cluster Name: one_node_hyperswap
```

```
Resource Group Name: ORARG
Node                             State
---------------------------- ---------------
r6r4m51                          UNMANAGED
$ id
uid=208(oracle) gid=2000(oinstall) groups=2001(dba),212(hagsuser)

$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.3.0 Production on Wed Dec 18 14:21:49 2013

Copyright (c) 1982, 2011, Oracle.  All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options

SQL> select status from v$instance;

STATUS
------------
OPEN
```

We start the PowerHA SystemMirror services, as shown in Example 8-41, without bringing the resource group online.

Example 8-41 Starting PowerHA SystemMirror, manually managing RGs

```
Start Cluster Services

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

                                                       [Entry Fields]
* Start now, on system restart or both                 now                  +
  Start Cluster Services on these nodes               [r6r4m51]             +
* Manage Resource Groups                               Manually             +
  BROADCAST message at startup?                        true                 +
  Startup Cluster Information Daemon?                  true                 +
  Ignore verification errors?                          false                +
  Automatically correct errors found during            Interactively        +
  cluster start?


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

COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

Adding any necessary PowerHA SystemMirror entries to /etc/inittab and
```

```
/etc/rc.net for IPAT on node r6r4m51.

Starting Cluster Services on node: r6r4m51
This may take a few minutes.  Please wait...
r6r4m51: start_cluster: Starting PowerHA SystemMirror
r6r4m51: Dec 18 2013 14:31:14 Starting execution of /usr/es/sbin/cluster/etc/rc.
cluster
r6r4m51: with parameters: -boot -N -M -b -i -C interactive -P cl_rc_cluster
r6r4m51:
r6r4m51: Dec 18 2013 14:31:14 Checking for srcmstr active...
r6r4m51: Dec 18 2013 14:31:14 complete.

F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
```

We add the Mirror Group configuration and add the new disk to the resource group. Notice that all of the disks should be picked up again from the list.

Next, we verify and synchronize the cluster configuration.

This time, the database is not affected. During the synchronization, in the `clxd.log`, we see that the mirror group is reconfiguration for new disks, as shown in Example 8-42.

Example 8-42 Mirror Group change seen in clxd.log

```
tail -f /var/hacmp/xd/log/clxd.log
.......................<<snipped text>>..............................
INFO      |2013-12-18T14:38:53.666828|Unplanned HyperSwap timeout = 60
INFO      |2013-12-18T14:38:53.666848|Volume group = oravg
INFO      |2013-12-18T14:38:53.666868|Raw Disks = 7d9ddb03-4c8c-9219-a461-0aa2fac14388
INFO      |2013-12-18T14:38:53.666888|Raw Disks = c444aae0-02f2-11a2-d0f0-a3615e926c85
INFO      |2013-12-18T14:38:53.666907|Raw Disks = 3be20bb3-2aa1-e421-ef06-fc9877cf486f
INFO      |2013-12-18T14:38:53.666927|Raw Disks = 6a56cac1-2d4a-912a-0d17-d702e32ca52a
.......................<<snipped text>>..............................
INFO      |2013-12-18T14:40:11.088056|Calling ADD_MIRROR_GROUP
INFO      |2013-12-18T14:40:11.088268|Calling CHANGE_MIRROR_GROUP
INFO      |2013-12-18T14:40:11.088835|ADD_MIRROR_GROUP completed
INFO      |2013-12-18T14:40:11.144843|Received XD CLI request = 'List Storage Modules' (0x6)
```

Now, we bring the resource group online, as shown in Example 8-43. We also monitor the database activity.

Example 8-43 Bring the resource group online

```
Resource Group and Applications

Move cursor to desired item and press Enter.

  Show the Current State of Applications and Resource Groups
  Bring a Resource Group Online
  Bring a Resource Group Offline
  Move Resource Groups to Another Node
  Move Resource Groups to Another Site

  Suspend/Resume Application Monitoring
  Application Availability Analysis
```

```
  ??????????????????????????????????????????????????????????????????????????????
  ?                          Select a Resource Group                           ?
  ?                                                                            ?
  ? Move cursor to desired item and press Enter.                               ?
  ?                                                                            ?
  ?   ORARG                              UNMANAGED                             ?
  ?                                                                            ?
  ? F1=Help                  F2=Refresh              F3=Cancel                 ?
  ? F8=Image                 F10=Exit                Enter=Do                  ?
F1? /=Find                   n=Find Next
```

The resource group remains online, as shown in Example 8-44.

Example 8-44 Resource group remains in Unmanaged state

```
COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

[TOP]
Attempting to bring group ORARG online on node r6r4m51.

Waiting for the cluster to process the resource group movement request....

Waiting for the cluster to stabilize......

Resource group movement successful.
Resource group ORARG is online on node r6r4m51.

Cluster Name: one_node_hyperswap
[MORE...6]

F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next
```

We validate that there were no error messages and verify the clxd.log, the hacmp.out log, and the database alert log, as shown in Example 8-45.

Example 8-45 Logs verification

```
$ tail -f /u01/app/oracle/diag/rdbms/itsodb/itsodb/trace/alert_itsodb.log
database for recovery-related files, and does not reflect the amount of
space available in the underlying filesystem or ASM diskgroup.
Wed Dec 18 13:11:47 2013
Starting background process CJQ0
Wed Dec 18 13:11:47 2013
CJQ0 started with pid=30, OS id=13369580
Wed Dec 18 13:21:46 2013
Starting background process SMCO
Wed Dec 18 13:21:46 2013
SMCO started with pid=21, OS id=1579421
```

```
clxd.log
INFO      |2013-12-18T14:46:35.402540|Calling sfwGetRepGroupInfo()
INFO      |2013-12-18T14:46:35.402578|sfwGetRepGroupInfo() completed
INFO      |2013-12-18T14:46:35.402600|Calling sfwGetRepDiskInfo()
INFO      |2013-12-18T14:46:35.402633|sfwGetRepDiskInfo() completed
INFO      |2013-12-18T14:46:35.402655|Volume state PRI=DS8K_VSTATE_ PRI,
SEL[0x0000000000010001] SEC=DS8K_VSTATE_SECONDARY[0x0000000000000002] for
RDG=pha_9654581787rdg2
INFO      |2013-12-18T14:46:35.402732|Calling sfwGetRepGroupInfo()
INFO      |2013-12-18T14:46:35.402766|sfwGetRepGroupInfo() completed
INFO      |2013-12-18T14:46:35.402789|Calling sfwGetRepDiskInfo()
INFO      |2013-12-18T14:46:35.402823|sfwGetRepDiskInfo() completed
INFO      |2013-12-18T14:46:35.402844|Volume state PRI=DS8K_VSTATE_ PRI,
SEL[0x0000000000010001] SEC=DS8K_VSTATE_SECONDARY[0x0000000000000002] for
RDG=pha_9654601806rdg3
INFO      |2013-12-18T14:46:35.402955|Calling START_MG
INFO      |2013-12-18T14:46:35.403306|Start Mirror Group 'ORA_MG' completed.
```

We configure the new added disk and verify the addition in the ASM configuration, as shown in Example 8-46.

Example 8-46 Verify ASM candidate disk

```
SQL> SELECT HEADER_STATUS,MOUNT_STATUS,MODE_STATUS,NAME,PATH,LABEL from
v$ASM_DISK;

HEADER_STATUMOUNT_SMODE_STNAME     PATH
LABEL
------------ ------- ------- ------------------------------
---------------------------------------------- -------------------------------
CANDIDATE    CLOSED  ONLINE                                 /dev/asm_disk4
MEMBER       CACHED  ONLINE  DATA_0000                      /dev/asm_disk1
MEMBER       CACHED  ONLINE  DATA_0001                      /dev/asm_disk2
MEMBER       CACHED  ONLINE  DATA_0002                      /dev/asm_disk3
```

Adding the new ASM disk to the DATA disk group is shown in Example 8-47.

Example 8-47 Adding new ASM disk in configuration

```
SQL> alter diskgroup data add disk '/dev/asm_disk4' name DATA_0004 ;

Diskgroup altered.

SQL> SELECT HEADER_STATUS,MOUNT_STATUS,MODE_STATUS,NAME,PATH,LABEL from
v$ASM_DISK;

HEADER_STATU MOUNT_S MODE_ST NAME                           PATH LABEL
------------ ------- ------- ------------------------------
---------------------------------------------------------------------------------
MEMBER       CACHED  ONLINE  DATA_0000                      /dev/asm_disk1
MEMBER       CACHED  ONLINE  DATA_0001                      /dev/asm_disk2
MEMBER       CACHED  ONLINE  DATA_0002                      /dev/asm_disk3
```

```
MEMBER       CACHED  ONLINE  DATA_0004                      /dev/asm_disk4
```

We verify that disks in the configuration have the same data replication direction. As we expected, all disk sources are on the storage with wwpn `500507630affc16b` and their replicated targets are on the auxiliary storage with wwpn `5005076308ffc6d4`, as shown in Example 8-48.

Example 8-48 Display information about PPRC

```
root@r6r4m51:/> lspprc -Ao |egrep 'hdisk31|hdisk61|hdisk63|hdisk41|hdisk42'
hdisk31   Active   0(s)       1          500507630affc16b  5005076308ffc6d4
hdisk41   Active   0(s)       1          500507630affc16b  5005076308ffc6d4
hdisk42   Active   0(s)       1          500507630affc16b  5005076308ffc6d4
hdisk61   Active   0(s)       1          500507630affc16b  5005076308ffc6d4
hdisk63   Active   0(s)       1          500507630affc16b  5005076308ffc6d4
```

8.17 Testing HyperSwap

PowerHA SystemMirror with the HyperSwap function offers protection against storage errors for applications that are configured to use disks that are protected by HyperSwap.

In case of storage failure in one site, the I/O is transparently routed to the remaining site as a function of HyperSwap. One cluster node in the configuration must remain active to monitor and keep the application running.

To provide a workload on the configured database that has data files on ASM, besides writing directly on disks during tests when ACFS is configured, we use Swingbench as a load data generator and as a benchmark. See the Swingbench website for installation and configuration details.

In our tests, we use the benchmark order entry, which provides a PL/SQL stress test model. This test is based on static PL/SQL with a small set of tables that are heavily queried and updated.

Also see the Oracle white paper titled “Evaluating and Comparing Oracle Database Appliance Performance:”

http://www.oracle.com/technetwork/server-storage/engineered-systems/database-appliance/documentation/oda-eval-comparing-performance-1895230.pdf

In parallel with Swingbench, we use a PL/SQL procedure to insert data into the database. This data is composed of a generated sequence, corresponding system timestamp, and the name of the instance when the insert is performed. The procedure is shown in Example 8-49.

Example 8-49 PL/SQL procedure for data generation

```
create or replace procedure insert_data is
 v_inst varchar2(20);
begin
  for i in 1..10000000000 loop
    begin
        select instance_name
        into v_inst
        from v_$instance;
```

```
        insert into performance(id,data,instance)
        values(performance_seq.nextval,systimestamp,v_inst);
        commit;
    exception
      when others then
        raise_application_error(-20101,'Error: '||sqlerrm);
    end;
  end loop;
end insert_data;
```

8.18 Single-node HyperSwap tests

The goal of our tests of this single-node HyperSwap is the storage migration from one site to another with the application online and, at the end, performing an unplanned HyperSwap.

These operations are performed, and each is described further in this section:

1. Planned HyperSwap from Storage A to Storage B
2. Storage migration: New storage is added in PowerHA configuration, and the HyperSwap configuration is used for migration between Storage B and Storage C
3. Unplanned HyperSwap is performed after the migration

8.18.1 Single-node HyperSwap: Planned HyperSwap

A planned HyperSwap operation can be used in these situations:

- When storage maintenance is performed on the primary site
- When a storage migration is required
- When a workload distribution on the node must be on different storage for some reason

As stated previously, because it is only a one-cluster node, there is no flexibility to move the applications to another site. Only the storage disks can be swapped.

In this scenario, we perform a planned HyperSwap operation for the ORA_MG mirror group while the Swingbench OE benchmark runs. To do that, you can use smitty fast path `cm_user_mirr_gp` or this path: **smitty cspoc** → **Storage** → **Manage Mirror Groups** → **Manage User Mirror Group(s).**

We select the desired mirror group for which we perform the planned swap as shown in Example 8-50.

Example 8-50 Performing planned swap for the ORA_MG user mirror group

```
                       Manage User Mirror Group(s)

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

                                                        [Entry Fields]
* Mirror Group(s)                                      ORA_MG             +
* Operation                                            Swap               +


???????????????????????????????????????????????????????????????????????????????
```

```
?                              SHOW COMMAND STRING                            ?
?                                                                             ?
? Press Enter or Cancel to return to the                                      ?
? application.                                                                ?
?                                                                             ?
?   /usr/es/sbin/cluster/xd_generic/xd_cli/cl_clxd_manage_mg_smit -t 'user   ?
?   '  -m 'ORA_MG' -o 'swap'                                                  ?
?                                                                             ?
F1? F1=Help                F2=Refresh             F3=Cancel                   ?
Es? F8=Image               F10=Exit               Enter=Do                    ?
F9?????????????????????????????????????????????????????????????????????????????
```

The HyperSwap operation is triggered, and we verify the disk status, as shown in Example 8-51.

Example 8-51 Result of planned swap operation

```
root@r6r4m51:/> lspprc -Ao |egrep 'hdisk31|hdisk61|hdisk63|hdisk41|hdisk42'
hdisk31   Active   1(s)        0            5005076308ffc6d4  500507630affc16b
hdisk41   Active   1(s)        0            5005076308ffc6d4  500507630affc16b
hdisk42   Active   1(s)        0            5005076308ffc6d4  500507630affc16b
hdisk61   Active   1(s)        0            5005076308ffc6d4  500507630affc16b
hdisk63   Active   1(s)        0            5005076308ffc6d4  500507630affc16b
```

Note: The single-node HyperSwap recovery action is set to Manual. As such, you must swap back the mirror group before the cluster is restarted. If manual recovery is not performed and you restart the cluster, you will be able to bring up the resource groups but the mirror group relation will not be started.

All process events are logged in the clxd.log as shown in Example 8-52.

Example 8-52 Planned HyperSwap logged on clxd.log

```
INFO      |2013-12-18T15:15:38.306237|Received XD CLI request = 'List Mirror Group' (0xc)
INFO      |2013-12-18T15:15:38.306391|MG Name='ORA_MG'
INFO      |2013-12-18T15:15:38.306412|MG Mode='Synchronous'
INFO      |2013-12-18T15:15:38.306432|CG Enabled = 'Yes'
INFO      |2013-12-18T15:15:38.306452|Recovery Action = 'Manual'
INFO      |2013-12-18T15:15:38.306490|Re-Sync Action = 'Automatic'
INFO      |2013-12-18T15:15:38.306511|Vendor's unique ID =
INFO      |2013-12-18T15:15:38.306531|Printing Storage System Set @(0x2005c680)
INFO      |2013-12-18T15:15:38.306555|Num Storage System: '2'
INFO      |2013-12-18T15:15:38.306576|Storage System Name = 'STG_A'
INFO      |2013-12-18T15:15:38.306595|Storage System Name = 'STG_B'
INFO      |2013-12-18T15:15:38.306615|Printing Opaque Attribute Value Set ... @(0x2018e54c)
INFO      |2013-12-18T15:15:38.306640|Number of Opaque Attributes Values = '0'
INFO      |2013-12-18T15:15:38.306661|HyperSwap Policy = Enabled
INFO      |2013-12-18T15:15:38.306685|MG Type = user
INFO      |2013-12-18T15:15:38.306705|HyperSwap Priority = medium
INFO      |2013-12-18T15:15:38.306726|Unplanned HyperSwap timeout = 60
INFO      |2013-12-18T15:15:38.306750|Volume group = oravg
INFO      |2013-12-18T15:15:38.306770|Raw Disks = 7d9ddb03-4c8c-9219-a461-0aa2fac14388
INFO      |2013-12-18T15:15:38.306790|Raw Disks = c444aae0-02f2-11a2-d0f0-a3615e926c85
INFO      |2013-12-18T15:15:38.306810|Raw Disks = 3be20bb3-2aa1-e421-ef06-fc9877cf486f
INFO      |2013-12-18T15:15:38.306830|Raw Disks = 6a56cac1-2d4a-912a-0d17-d702e32ca52a
INFO      |2013-12-18T15:15:39.885080|Received XD CLI request = '' (0x1d)
```

```
INFO      |2013-12-18T15:15:40.885310|Received XD CLI request = 'Swap Mirror Group' (0x1c)
INFO      |2013-12-18T15:15:40.885340|Request to Swap Mirror Group 'ORA_MG', Direction
'Site2_secondary', Outfile ''
..........................<<snippet>>.......................
INFO      |2013-12-18T15:15:41.070814|Calling DO_SWAP
INFO      |2013-12-18T15:15:41.144146|DO_SWAP completed
INFO      |2013-12-18T15:15:41.144361|Swap Mirror Group 'ORA_MG' completed.
```

8.18.2 Single-node HyperSwap: Storage migration

The HyperSwap feature with PowerHA SystemMirror offers multiple advantages when storage migration, relocation or maintenance are required to be performed on a critical environment where an outage is not possible.

When a physically storage relocation is required, maintaining the imposed replication limit of maximum 100KM, HyperSwap can be used to achieve the business continuity storage related without a scheduled outage. In this case, all the disks are swapped to the storage that will remain in place. If the entire site is relocated, a HyperSwap PowerHA SystemMirror active-active configuration could be put in place.

Storage migration steps:

- Validate disks location to be on remaining storage. If this conditions is not satisfied, a planned swap is required to bring all disks on the remaining storage.
- PowerHA services must be stopped with Unmanaged groups options (on all the nodes where the resource groups and mirror groups are online).
- **chdev -l hdisk# -a san_rep_cfg=revert_disk -U** (for all the disks part of MG).
- **rmpprc** for the HyperSwap disks to the existing auxiliary storage.
- **rmmpprcpath** to the existing auxiliary storage (optional).
- **rmdev -dl hdisk**# (for all LUNs from the auxiliary storage).
- Create the PPRC paths using the **mkpprcpath** command with the new auxiliary storage.
- **mkpprc** for existing disks to the new auxiliary storage.
- Perform zoning configuration for every host with the new auxiliary storage.
- Configure the hostconnect HyperSwap profile for every host attached to the new auxiliary storage.
- **cfgmgr**, **chdev** to no_reserve, **chdev -l hdisk# -a san_rep_cfg=migrate_disk -U** (for all the disks from new auxiliary storage).
- Create a new storage subsystem.
- Perform a change mirror group and freshly add raw disks and VGs again.
- Perform a verify and synchronize.
- Start PowerHA services in all nodes with unmanage resources.
- Inspect the clxd.log for error free.
- Bring the resource groups online.

In this scenario, we use a single-node HyperSwap cluster with the disks configured for Metro Mirror replication using storage subsystems STG _A and STG_B. The goal is to migrate online the storage STG_A to STG_C having the application up and running as shown in Figure 8-7.

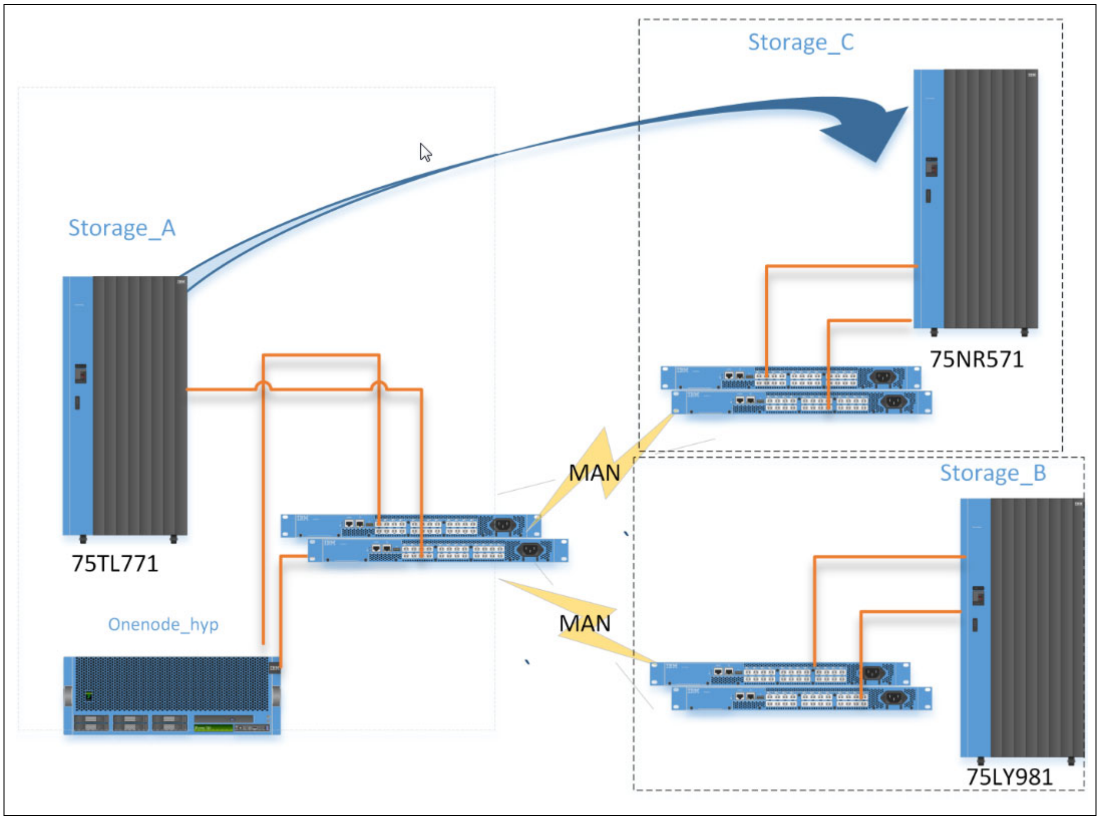


Figure 8-7 Storage migration schema

Node r6r4m51is connected to Storage_A, Storage_B, Storage_C and configured as node cluster in single-node HyperSwap cluster.

The storages and disk configurations are shown in Table 8-5 and Table 8-6 on page 262.

Table 8-5 Storage systems DS8800 FW

STORAGE	DEV ID/LMC	WWPN
STORAGE_A	75TL771/7.6.31.930	500507630AFFC16B
STORAGE_B	75LY981/7.6.31.136	5005076308FFC6D4
STORAGE_C	75NR571/7.6.31.930	5005076309FFC5D5

The storage migration is performed having configured a database on the node and an appropriate workload using Swingbench load generator.

The disks provided from each storage are shown in Table 8-6 on page 262.

Table 8-6 Storage disks

NODE r6r4m51	hdisk31	hdisk41	hdisk42	hdisk61
Storage_A	**44**04	**52**04	**52**05	**B2**07
Storage_B	**00**04	**0E**04	**0E**05	**E2**04
Storage_C	**C2**04	**C3**04	**C3**05	C404

On the Storage_C, the r6r4m51 node is not configured for HyperSwap, as shown in Example 8-53.

Example 8-53 r6r4m51 hostconnect profile definition on Storage _C

```
dscli> lssi
Date/Time: December 15, 2013 4:13:07 PM CST IBM DSCLI Version: 6.6.0.305 DS: -
Name  ID               Storage Unit     Model WWNN             State  ESSNet
============================================================================
ds8k5 IBM.2107-75NR571 IBM.2107-75NR570 951   5005076309FFC5D5 Online Enabled

lshostconnect -dev IBM.2107-75NR571 -l|egrep '10000000C96C387A|10000000C96C387B'
r6r4m51_0         0058 10000000C96C387A pSeries  512 reportLUN     IBM pSeries -
AIX                         58 V12      -        all       Unknown
r6r4m51_1         0059 10000000C96C387B pSeries  512 reportLUN     IBM pSeries -
AIX                         58 V12      -        all       Unknown
```

Configuring the node for in-band communication and HyperSwap capable requires the host profile for the corresponding hostconnect to be *IBM pSeries - AIX with Powerswap support* as shown in Example 8-54.

Example 8-54 Find available host profiles in the storage system

```
dscli> lsportprof IBM.2107-75NR571
Date/Time: December 19, 2013 6:51:26 PM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75NR571
Profile                                  AddrDiscovery LBS
==========================================================
..............<<snipped>>....................
IBM pSeries - AIX                        reportLUN     512
IBM pSeries - AIX/SanFS                  reportLUN     512
IBM pSeries - AIX with Powerswap support -             -
...............<<snipped>>....................
```

We change the hostconnect with above profile (Example 8-54) as shown in Example 8-55.

Example 8-55 Changing hostconnect profile to "IBM pSeries - AIX with Powerswap support"

```
dscli> chhostconnect -profile "IBM pSeries - AIX with Powerswap support" 0058
Date/Time: December 19, 2013 6:57:18 PM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75NR571
CMUC00013I chhostconnect: Host connection 0058 successfully modified.
dscli> chhostconnect -profile "IBM pSeries - AIX with Powerswap support" 0059
Date/Time: December 19, 2013 6:57:30 PM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75NR571
CMUC00013I chhostconnect: Host connection 0059 successfully modified.
```

In our scenario, we migrate the storage from the primary site. In the following migration steps, we validate first that all disks are sources in the remaining storage as shown in Example 8-56.

Example 8-56 Validating disk locations

```
root@r6r4m51:/> lspprc -Ao |egrep 'hdisk31|hdisk61|hdisk63|hdisk41|hdisk42'
hdisk31   Active   0(s)       1           500507630affc16b  5005076308ffc6d4
hdisk41   Active   0(s)       1           500507630affc16b  5005076308ffc6d4
hdisk42   Active   0(s)       1           500507630affc16b  5005076308ffc6d4
hdisk61   Active   0(s)       1           500507630affc16b  5005076308ffc6d4
hdisk63   Active   0(s)       1           500507630affc16b  5005076308ffc6d4
```

The disks are located in Storage_A. We swap the disks to storage_B since the storage_A is being migrated.The disks are swapped to site SITE_B and their configuration is shown in Example 8-57.

Example 8-57 Validating disk location on remaining storage after swap operation

```
root@r6r4m51:/> lspprc -Ao |egrep 'hdisk31|hdisk61|hdisk63|hdisk41|hdisk42'
hdisk31   Active   1(s)       0           5005076308ffc6d4  500507630affc16b
hdisk41   Active   1(s)       0           5005076308ffc6d4  500507630affc16b
hdisk42   Active   1(s)       0           5005076308ffc6d4  500507630affc16b
hdisk61   Active   1(s)       0           5005076308ffc6d4  500507630affc16b
hdisk63   Active   1(s)       0           5005076308ffc6d4  500507630affc16b
```

At this time, we stop the cluster services and leave the resource group in Unmanaged state as shown in Example 8-58.

Example 8-58 RG remains on Unmanaged state and verify the database

```
root@r6r4m51:/> clRGinfo -p

Cluster Name: one_node_hyperswap
Resource Group Name: ORARG
Node                         State
---------------------------- ---------------
r6r4m51                      UNMANAGED

SQL> TO_CHAR(SYSDATE, 'dd-mm-yy hh24:mi:ss') as “DATE” from dual;

DATE
-----------------
19-12-13 19:12:14
```

Using the disk reverting procedure, we revert the disk to have the same number when the primary storage is removed as shown in Example 8-59.

Example 8-59 Reverting disks

```
root@r6r4m51:/> for i in 31 41 42 61 63; do chdev -l hdisk$i -a
san_rep_cfg=revert_disk -U;done
hdisk31 changed
hdisk41 changed
hdisk42 changed
hdisk61 changed
hdisk63 changed
```

The disks are no more HyperSwap enabled, and we verify this as shown in Example 8-60.

Example 8-60 Verifying disk status after reconfiguration

```
root@r6r4m51:/> lspprc -Ao |egrep 'hdisk31|hdisk61|hdisk63|hdisk41|hdisk42'
hdisk31   Active   1(s)        -1          5005076308ffc6d4  500507630affc16b
hdisk41   Active   1(s)        -1          5005076308ffc6d4  500507630affc16b
hdisk42   Active   1(s)        -1          5005076308ffc6d4  500507630affc16b
hdisk61   Active   1(s)        -1          5005076308ffc6d4  500507630affc16b
hdisk63   Active   1(s)        -1          5005076308ffc6d4  500507630affc16b
root@r6r4m51:/>

root@r6r4m51:/> for i in 31 41 42 61 63; do lspprc -v hdisk$i ;done

HyperSwap lun unique identifier.............200B75LY981000407210790003IBMfcp

hdisk31 Primary         MPIO IBM 2107 FC Disk

        Manufacturer..................IBM
        Machine Type and Model........2107900
        ROS Level and ID..............2E313336
        Serial Number.................75LY9810
        Device Specific.(Z7)..........0004
        Device Specific.(Z0)..........000005329F101002
        Device Specific.(Z1)..........004
        Device Specific.(Z2)..........075
        Unique Device Identifier......200B75LY981000407210790003IBMfcp
        Logical Subsystem ID..........0x00
        Volume Identifier.............0x04
        Subsystem Identifier(SS ID)...0xFF00
        Control Unit Sequence Number..00000LY981
        Storage Subsystem WWNN........5005076308ffc6d4
        Logical Unit Number ID........4000400400000000

HyperSwap lun unique identifier.............200B75LY9810E0407210790003IBMfcp

hdisk41 Primary         MPIO IBM 2107 FC Disk

        Manufacturer..................IBM
        Machine Type and Model........2107900
        ROS Level and ID..............2E313336
        Serial Number.................75LY9810
        Device Specific.(Z7)..........0E04
        Device Specific.(Z0)..........000005329F101002
        Device Specific.(Z1)..........E04
        Device Specific.(Z2)..........075
        Unique Device Identifier......200B75LY9810E0407210790003IBMfcp
        Logical Subsystem ID..........0x0e
        Volume Identifier.............0x04
        Subsystem Identifier(SS ID)...0xFF0E
        Control Unit Sequence Number..00000LY981
        Storage Subsystem WWNN........5005076308ffc6d4
        Logical Unit Number ID........400e400400000000

HyperSwap lun unique identifier.............200B75LY9810E0507210790003IBMfcp
```

```
hdisk42 Primary         MPIO IBM 2107 FC Disk

        Manufacturer..................IBM
        Machine Type and Model........2107900
        ROS Level and ID..............2E313336
        Serial Number.................75LY9810
        Device Specific.(Z7)..........0E05
        Device Specific.(Z0)..........000005329F101002
        Device Specific.(Z1)..........E05
        Device Specific.(Z2)..........075
        Unique Device Identifier......200B75LY9810E0507210790003IBMfcp
        Logical Subsystem ID..........0x0e
        Volume Identifier.............0x05
        Subsystem Identifier(SS ID)...0xFF0E
        Control Unit Sequence Number..00000LY981
        Storage Subsystem WWNN........5005076308ffc6d4
        Logical Unit Number ID........400e400500000000

HyperSwap lun unique identifier.............200B75LY981E20407210790003IBMfcp

hdisk61 Primary         MPIO IBM 2107 FC Disk

        Manufacturer..................IBM
        Machine Type and Model........2107900
        ROS Level and ID..............2E313336
        Serial Number.................75LY981E
        Device Specific.(Z7)..........E204
        Device Specific.(Z0)..........000005329F101002
        Device Specific.(Z1)..........204
        Device Specific.(Z2)..........075
        Unique Device Identifier......200B75LY981E20407210790003IBMfcp
        Logical Subsystem ID..........0xe2
        Volume Identifier.............0x04
        Subsystem Identifier(SS ID)...0xFFE2
        Control Unit Sequence Number..00000LY981
        Storage Subsystem WWNN........5005076308ffc6d4
        Logical Unit Number ID........40e2400400000000

HyperSwap lun unique identifier.............200B75LY981E70007210790003IBMfcp

hdisk63 Primary         MPIO IBM 2107 FC Disk

        Manufacturer..................IBM
        Machine Type and Model........2107900
        ROS Level and ID..............2E313336
        Serial Number.................75LY981E
        Device Specific.(Z7)..........E700
        Device Specific.(Z0)..........000005329F101002
        Device Specific.(Z1)..........700
        Device Specific.(Z2)..........075
        Unique Device Identifier......200B75LY981E70007210790003IBMfcp
        Logical Subsystem ID..........0xe7
        Volume Identifier.............0x00
        Subsystem Identifier(SS ID)...0xFFE7
```

```
        Control Unit Sequence Number..00000LY981
        Storage Subsystem WWNN........5005076308ffc6d4
        Logical Unit Number ID........40e7400000000000
```

On the Storage_B, we validate the PPRC relationships as shown in Example 8-61.

Example 8-61 Validating PPRC relationships

```
dscli> lspprc -fullid -l 0004 0e04 0e05 e204 e700
Date/Time: December 19, 2013 7:34:51 PM CST IBM DSCLI Version: 6.6.0.305 DS: IBM.2107-75LY981
ID                                         State      Reason Type        Out Of Sync Tracks
Tgt Read Src Cascade Tgt Cascade Date Suspended SourceLSS        Timeout (secs) Critical Mode
First Pass Status Incremental Resync Tgt Write GMIR CG PPRC CG  isTgtSE DisableAutoResync
===============================================================================================
===============================================================================================
=========================================================================================
IBM.2107-75LY981/0004:IBM.2107-75TL771/4404 Full Duplex -     Metro Mirror 0
Disabled Disabled    Invalid     -             IBM.2107-75LY981/00 60           Disabled
Invalid          Disabled          Disabled  N/A    Disabled Unknown -
IBM.2107-75LY981/0E04:IBM.2107-75TL771/5204 Full Duplex -     Metro Mirror 0
Disabled Disabled    Invalid     -             IBM.2107-75LY981/0E 60           Disabled
Invalid          Disabled          Disabled  N/A    Disabled Unknown -
IBM.2107-75LY981/0E05:IBM.2107-75TL771/5205 Full Duplex -     Metro Mirror 0
Disabled Disabled    Invalid     -             IBM.2107-75LY981/0E 60           Disabled
Invalid          Disabled          Disabled  N/A    Disabled Unknown -
IBM.2107-75LY981/E204:IBM.2107-75TL771/B207 Full Duplex -     Metro Mirror 0
Disabled Disabled    Invalid     -             IBM.2107-75LY981/E2 60           Disabled
Invalid          Disabled          Disabled  N/A    Disabled Unknown -
IBM.2107-75LY981/E700:IBM.2107-75TL771/B406 Full Duplex -     Metro Mirror 0
Disabled Disabled    Invalid     -             IBM.2107-75LY981/E7 60           Disabled
Invalid          Disabled          Disabled  N/A    Disabled Unknown -
```

We remove the PPRC relationships for the corresponding disks as shown in Example 8-62. Depending on the disk load, it is recommended to use the **pausepprc** command to pause all mirrored disks before removing the relationships with the **rmpprc** command operation.

Example 8-62 Removing PPRC relationships

```
dscli> rmpprc -remotedev IBM.2107-75TL771 0004:4404 0E04:5204 0E05:5205 E204:B207
E700:B406
Date/Time: December 19, 2013 7:38:34 PM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75LY981
CMUC00160W rmpprc: Are you sure you want to delete the Remote Mirror and Copy
volume pair relationship 0004:4404:? [y/n]:y
CMUC00155I rmpprc: Remote Mirror and Copy volume pair 0004:4404 relationship
successfully withdrawn.
CMUC00160W rmpprc: Are you sure you want to delete the Remote Mirror and Copy
volume pair relationship 0E04:5204:? [y/n]:y
CMUC00155I rmpprc: Remote Mirror and Copy volume pair 0E04:5204 relationship
successfully withdrawn.
CMUC00160W rmpprc: Are you sure you want to delete the Remote Mirror and Copy
volume pair relationship 0E05:5205:? [y/n]:y
CMUC00155I rmpprc: Remote Mirror and Copy volume pair 0E05:5205 relationship
successfully withdrawn.
```

```
CMUC00160W rmpprc: Are you sure you want to delete the Remote Mirror and Copy
volume pair relationship E204:B207:? [y/n]:y
CMUC00155I rmpprc: Remote Mirror and Copy volume pair E204:B207 relationship
successfully withdrawn.
CMUC00160W rmpprc: Are you sure you want to delete the Remote Mirror and Copy
volume pair relationship E700:B406:? [y/n]:y
CMUC00155I rmpprc: Remote Mirror and Copy volume pair E700:B406 relationship
successfully withdrawn.
```

Note: **rmpprc** with **-quiet** switch can be used to eliminate the operation confirmation.

We check if the disks are visible on the system as shown in Example 8-63.

Example 8-63 Validate the disk volume ID

```
root@r6r4m51:/work> lshostvol.sh |egrep '4404|5204|5205|B207|B406'
 hdisk6           IBM.2107-75TL771/4404
 hdisk7           IBM.2107-75TL771/5204
 hdisk22          IBM.2107-75TL771/5205
 hdisk27          IBM.2107-75TL771/B207
 hdisk31          IBM.2107-75TL771/4404
 hdisk41          IBM.2107-75TL771/5204
 hdisk42          IBM.2107-75TL771/5205
 hdisk61          IBM.2107-75TL771/B207
 hdisk63          IBM.2107-75TL771/B406
 hdisk71          IBM.2107-75TL771/B406
```

The configured disks are in bold in Example 8-63. The rest of the disks have the ID shown in Example 8-64.

Example 8-64 Matching disk ID for disks required to be removed

```
root@r6r4m51:/work> for i in hdisk6 hdisk7 hdisk22 hdisk27 hdisk71; do lscfg -vpl
$i|grep Z7;done
        Device Specific.(Z7)........4404
        Device Specific.(Z7)........5204
        Device Specific.(Z7)........5205
        Device Specific.(Z7)........B207
        Device Specific.(Z7)........B406
```

We remove the disks as shown in Example 8-65. Considering the storage is removed from our configuration, we do not have to change the LUN masking for corresponding disks on the storage side.

Example 8-65 Removing all disks provided of former main storage

```
root@r6r4m51:/work> for i in hdisk6 hdisk7 hdisk22 hdisk27 hdisk71; do rmdev -dRl
$i;done
hdisk6 deleted
hdisk7 deleted
hdisk22 deleted
hdisk27 deleted
hdisk71 deleted
```

> **Note:** It is indicated to remove the volume host mapping for all volumes taken out from configuration since running a **cfgmgr** after their removal displays the volumes again but with VG information.
>
> Only after **chdev -l hdisk# -a revert_disk -U** the disks that are taken out and have **none** for VG.
>
> In case you have Volume Groups on these hdisks, they will be seen by the system and you must **exportvg** and **importvg** these disks after their removal.

We create at this time the volumes, the PPRC paths and we start the Metro Mirror replication for the corresponding volumes on the new attached storage subsystems as shown in Example 8-66.

Example 8-66 Create pprcpaths between remaining storage and the new auxiliary storage

```
              On the storage Storage_B

dscli> lssi
Date/Time: December 19, 2013 8:21:45 PM CST IBM DSCLI Version: 6.6.0.305 DS: -
Name  ID               Storage Unit     Model WWNN             State  ESSNet
============================================================================
ds8k6 IBM.2107-75LY981 IBM.2107-75LY980 951   5005076308FFC6D4 Online Enabled

dscli>  mkpprcpath -remotedev IBM.2107-75NR571 -remotewwnn 5005076309FFC5D5 -srclss 00 -tgtlss
c2 -consistgrp I0207:I0132
Date/Time: December 19, 2013 8:04:21 PM CST IBM DSCLI Version: 6.6.0.305 DS: IBM.2107-75LY981
CMUC00149I mkpprcpath: Remote Mirror and Copy path 00:c2 successfully established.
dscli> mkpprcpath -remotedev IBM.2107-75NR571 -remotewwnn 5005076309FFC5D5 -srclss 0e -tgtlss c3
-consistgrp I0207:I0132
Date/Time: December 19, 2013 8:07:37 PM CST IBM DSCLI Version: 6.6.0.305 DS: IBM.2107-75LY981
CMUC00149I mkpprcpath: Remote Mirror and Copy path 0e:c3 successfully established.
dscli> mkpprcpath -remotedev IBM.2107-75NR571 -remotewwnn 5005076309FFC5D5 -srclss e2 -tgtlss c4
-consistgrp I0207:I0132
Date/Time: December 19, 2013 8:09:25 PM CST IBM DSCLI Version: 6.6.0.305 DS: IBM.2107-75LY981
CMUC00149I mkpprcpath: Remote Mirror and Copy path e2:c4 successfully established.
dscli>  mkpprcpath -remotedev IBM.2107-75NR571 -remotewwnn 5005076309FFC5D5 -srclss e7 -tgtlss
c5 -consistgrp I0207:I0132
Date/Time: December 19, 2013 8:10:26 PM CST IBM DSCLI Version: 6.6.0.305 DS: IBM.2107-75LY981
CMUC00149I mkpprcpath: Remote Mirror and Copy path e7:c5 successfully established.

              On the storage Storage_C

dscli> lssi
Date/Time: December 19, 2013 8:31:46 PM CST IBM DSCLI Version: 6.6.0.305 DS: -
Name  ID               Storage Unit     Model WWNN             State  ESSNet
============================================================================
ds8k5 IBM.2107-75NR571 IBM.2107-75NR570 951   5005076309FFC5D5 Online Enabled

dscli>mkpprcpath -remotewwnn 5005076308FFC6D4 -srclss c2 -tgtlss 00 -consistgrp I023
1:I0130
Date/Time: December 19, 2013 8:15:21 PM CST IBM DSCLI Version: 6.6.0.305 DS: IBM.2107
-75NR571
CMUC00149I mkpprcpath: Remote Mirror and Copy path c2:00 successfully established.
dscli> mkpprcpath -remotewwnn 5005076308FFC6D4 -srclss c3 -tgtlss 0e -consistgrp I0231:I0130
```

```
Date/Time: December 19, 2013 8:16:13 PM CST IBM DSCLI Version: 6.6.0.305 DS: IBM.2107-75NR571
CMUC00149I mkpprcpath: Remote Mirror and Copy path c3:0e successfully established.
dscli>  mkpprcpath -remotewwnn 5005076308FFC6D4 -srclss c4 -tgtlss e2 I0231:I0130
Date/Time: December 19, 2013 8:16:44 PM CST IBM DSCLI Version: 6.6.0.305 DS: IBM.2107-75NR571
CMUC00149I mkpprcpath: Remote Mirror and Copy path c4:e2 successfully established.
dscli>  mkpprcpath -remotewwnn 5005076308FFC6D4 -srclss c5 -tgtlss e7  I0231:I0130
Date/Time: December 19, 2013 8:17:28 PM CST IBM DSCLI Version: 6.6.0.305 DS: IBM.2107-75NR571
CMUC00149I mkpprcpath: Remote Mirror and Copy path c5:e7 successfully established.
```

We establish the Metro Mirror relationships for the corresponding disks as shown in Example 8-67.

Example 8-67 mkpprc on the storage STORAGE_B for Storage_C disks

```
dscli> mkpprc -remotedev IBM.2107-75NR571  -type mmir 0004:c204 0E04:c304
e204:c404 e700:c500
Date/Time: December 19, 2013 8:22:17 PM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75LY981
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 0004:C204
successfully created.
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 0E04:C304
successfully created.
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship E204:C404
successfully created.
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship E700:C500
successfully created.
dscli> mkpprc -remotedev IBM.2107-75NR571  -type mmir 0e05:c305
Date/Time: December 19, 2013 8:24:23 PM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75LY981
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 0E05:C305
successfully created.
```

On the node, we observe that the disks were added in number of 5 as shown in Example 8-68.

Example 8-68 Validate disks added to the system

```
root@r6r4m51:/work> lshostvol.sh |egrep 'C204|C304|C305|C404|C500'
 hdisk72          IBM.2107-75NR571/C204
 hdisk73          IBM.2107-75NR571/C304
 hdisk74          IBM.2107-75NR571/C305
 hdisk75          IBM.2107-75NR571/C404
 hdisk76          IBM.2107-75NR571/C500
```

Setting up disk attributes is a required task for newly added disks from the storage STORAGE_ C before migrating them to the HyperSwap configuration as shown in Example 8-69 on page 270.

Example 8-69 Preliminary configuration for new added disks

```
root@r6r4m51:/work> for i in hdisk72 hdisk73 hdisk74 hdisk75 hdisk76; do chdev -l
$i -a reserve_policy=no_reserve;done
hdisk72 changed
hdisk73 changed
hdisk74 changed
hdisk75 changed
hdisk76 changed
```

Now we configure the previous reverted disks to be migrate_disk as shown in Example 8-70.

Example 8-70 Migrate to HyperSwap disks

```
for i in 31 41 42 61 63; do chdev -l hdisk$i -a san_rep_cfg=migrate_disk -U;done
hdisk31 changed
hdisk41 changed
hdisk42 changed
hdisk61 changed
hdisk63 changed
root@r6r4m51:/work> lspprc -Ao |egrep 'hdisk31|hdisk61|hdisk63|hdisk41|hdisk42'
hdisk31   Active   1(s)        0            5005076308ffc6d4  5005076309ffc5d5
hdisk41   Active   1(s)        0            5005076308ffc6d4  5005076309ffc5d5
hdisk42   Active   1(s)        0            5005076308ffc6d4  5005076309ffc5d5
hdisk61   Active   1(s)        0            5005076308ffc6d4  5005076309ffc5d5
hdisk63   Active   1(s)        0            5005076308ffc6d4  5005076309ffc5d5
```

Since the storage has been changed, we must reconfigure the configure DS8000 Metro Mirror (In-Band) Resources to reflect the new changes.The storages remain as they are defined and the new storage STORAGE_C is added on the primary site as shown in Example 8-71.

Example 8-71 Adding the new storage system on the primary site

```
Add a Storage System

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

                                                             [Entry Fields]
* Storage System Name                                      [STG_C]
* Site Association                                          Site1_primary        +
* Vendor Specific Identifier                                IBM.2107-00000NR571   +
* WWNN                                                      5005076309FFC5D5     +


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

We configure again the mirror group adding the disks in the configuration as shown in Example 8-72 on page 271.

Example 8-72 Reconfiguring the mirror groups

```
Change/Show a User Mirror Group

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

                                                          [Entry Fields]
  Mirror Group Name                                     ORA_MG
  New Mirror Group Name                                []
  Volume Group(s)                                       oravg               +
  Raw Disk(s)                                          [f64bde11-9356-53fe-68> +
  Associated Storage System(s)                          STG_C STG_B         +
  HyperSwap                                             Enabled             +
  Consistency Group                                     Enabled             +
  Unplanned HyperSwap Timeout (in sec)                 [60]                  #
  HyperSwap Priority                                    Medium
  Recovery Action                                       Manual              +
  Re-sync Action                                        Automatic           +


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

The storage systems appear with the new relationship after re-adding the disks.

Note: You do not have to reconfigure the Associated Storage System(s) field. Even if this field is modified, it shows the relationship from the primary site to the secondary site.

We also configure the resource group disks to reflect the new changes, and verify and synchronize the cluster at this point.

We bring the resource group online and validate that the database is still open and functional as shown in Example 8-73.

Example 8-73 Bring the ORARG online

```
Resource Group and Applications

Move cursor to desired item and press Enter.

  Show the Current State of Applications and Resource Groups
  Bring a Resource Group Online
  Bring a Resource Group Offline
  Move Resource Groups to Another Node
  Move Resource Groups to Another Site

  Suspend/Resume Application Monitoring
  Application Availability Analysis

  ?????????????????????????????????????????????????????????????????????????????
  ?                         Select a Resource Group                           ?
```

```
  ?                                                                          ?
  ? Move cursor to desired item and press Enter.                             ?
  ?                                                                          ?
  ?   ORARG                              UNMANAGED                           ?
  ?                                                                          ?
  ? F1=Help                F2=Refresh              F3=Cancel                 ?
  ? F8=Image               F10=Exit                Enter=Do                  ?
F1? /=Find                 n=Find Next                                       ?
F9???????????????????????????????????????????????????????????????????????????
```

Since we have only one cluster node, a message as shown in Example 8-74 appears in the clxd.log.

Example 8-74 Message error when RHG is brought online

```
Failed to Start Mirror Group  'ORA_MG' . rc=2 retval=2 errno=10 err_str=Auto
recovery is not allowed here for start MG as swap is needed here and no sibling
node exists
```

As such, we stop again the services, put the resource group in Unmanaged mode and reverse the PPRC relationship to be from STG_C to STG_B. The process is shown in the Example 8-75.

Example 8-75 Matching required mirror group configuration

```
Bring the RG in unmanaged state.
Croot@r6r4m51:/work> clRGinfo -p

Cluster Name: one_node_hyperswap

Resource Group Name: ORARG
Node                         State
---------------------------- ---------------
r6r4m51                      UNMANAGED
root@r6r4m51:/work>
```

On the storage side we reverse the Metro Mirror relation to be from STG_C to STG_B as shown below:

```
dscli> failoverpprc -remotedev IBM.2107-75LY981 -type mmir c204:0004 c304:0e04 c305:0e05
c404:e204 c500:e700
Date/Time: December 19, 2013 9:27:42 PM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75NR571
CMUC00196I failoverpprc: Remote Mirror and Copy pair C204:0004 successfully reversed.
CMUC00196I failoverpprc: Remote Mirror and Copy pair C404:E204 successfully reversed.
CMUC00196I failoverpprc: Remote Mirror and Copy pair C304:0E04 successfully reversed.
CMUC00196I failoverpprc: Remote Mirror and Copy pair C305:0E05 successfully reversed.
CMUC00196I failoverpprc: Remote Mirror and Copy pair C500:E700 successfully reversed.

dscli> failbackpprc -remotedev IBM.2107-75LY981 -type mmir c204:0004 c304:0e04 c305:0e05
c404:e204 c500:e700
Date/Time: December 19, 2013 9:28:36 PM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75NR571
CMUC00197I failbackpprc: Remote Mirror and Copy pair C204:0004 successfully failed back.
CMUC00197I failbackpprc: Remote Mirror and Copy pair C404:E204 successfully failed back.
CMUC00197I failbackpprc: Remote Mirror and Copy pair C304:0E04 successfully failed back.
```

```
CMUC00197I failbackpprc: Remote Mirror and Copy pair C305:0E05 successfully failed back.
CMUC00197I failbackpprc: Remote Mirror and Copy pair C500:E700 successfully failed back.

root@r6r4m51:/work> lspprc -Ao |egrep 'hdisk31|hdisk61|hdisk63|hdisk41|hdisk42'
hdisk31   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk41   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk42   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk61   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk63   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
root@r6r4m51:/work>
```

At this time, we start the PowerHA services and bring online the resource group. The operation status is displayed in Example 8-76.

Example 8-76 Starting PowerHA services

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

Before command completion, additional instructions may appear below.

Adding any necessary PowerHA SystemMirror entries to /etc/inittab and /etc/rc.net
for IPAT on node r6r4m51.

.............................<<snipet>>.......................................
Starting Cluster Services on node: r6r4m51
This may take a few minutes.  Please wait...
r6r4m51: start_cluster: Starting PowerHA SystemMirror
r6r4m51: Dec 19 2013 21:31:26 Starting execution of
/usr/es/sbin/cluster/etc/rc.cluster
r6r4m51: with parameters: -boot -N -A -b -i -C interactive -P cl_rc_cluster
r6r4m51:
r6r4m51: Dec 19 2013 21:31:26 Checking for srcmstr active...
r6r4m51: Dec 19 2013 21:31:26 complete.


Using tail -f /var/hacmp/xd/log/clxd.log
.............................<<snipet>>.......................................
INFO      |2013-12-19T21:31:41.289647|Volume state PRI=DS8K_VSTATE_ PRI,
SEL[0x0000000000010001] SEC=DS8K_VSTATE_SECONDARY[0x0000000000000002] for
RDG=pha_9655062264rdg2
INFO      |2013-12-19T21:31:41.289714|Calling sfwGetRepGroupInfo()
INFO      |2013-12-19T21:31:41.289758|sfwGetRepGroupInfo() completed
INFO      |2013-12-19T21:31:41.289781|Calling sfwGetRepDiskInfo()
INFO      |2013-12-19T21:31:41.289815|sfwGetRepDiskInfo() completed
INFO      |2013-12-19T21:31:41.289837|Volume state PRI=DS8K_VSTATE_ PRI,
SEL[0x0000000000010001] SEC=DS8K_VSTATE_SECONDARY[0x0000000000000002] for
RDG=pha_9655112310rdg3
INFO      |2013-12-19T21:31:41.289946|Calling START_MG
INFO      |2013-12-19T21:31:41.290295|Start Mirror Group 'ORA_MG' completed.
```

We validate the database is up as shown Example 8-77 on page 274.

Example 8-77 Validating database connection

```
SQL> SELECT TO_CHAR (startup_time, 'dd-mon-yyyy hh24:mi:ss') start_time from
v$instance;

START_TIME
-----------------------------
19-dec-2013 15:54:06
```

8.18.3 Single-node HyperSwap: Unplanned HyperSwap

Having the configuration as shown in Example 8-78 where all disks are located on the storage DS8k5, we simulate the loss of the SAN paths between node r6r4m51 and the storage DS8K5.

This scenario simulates an unplanned HyperSwap by deactivating the zones on the SAN switches. In Example 8-78, we present the existing zoning configuration for the cluster node r6r4m51.

Example 8-78 Two zones defined for r6r4m51 per attached storage

```
zone:  r6r4m51_fcs0_ds8k5
                DS8K5_I0130; DS8K5_I0131; DS8K5_I0132; r6r4m51_fcs0
zone:  r6r4m51_fcs0_ds8k6
                r6r4m51_fcs0; DS8K6_I0204; DS8K6_I0205
zone:  r6r4m51_fcs1_ds8k5
                DS8K5_I0130; DS8K5_I0131; DS8K5_I0132; r6r4m51_fcs1
 zone:  r6r4m51_fcs1_ds8k6
                r6r4m51_fcs1; DS8K6_I0204; DS8K6_I0205
```

We validate the replication direction for all disks which are swapped from Storage_C to Storage_B. The disk configurations is shown in Example 8-79.

Example 8-79 Disk configuration before swap

```
root@r6r4m51:/work> lspprc -Ao |egrep 'hdisk31|hdisk61|hdisk63|hdisk41|hdisk42'
hdisk31   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk41   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk42   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk61   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk63   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
```

Before the SAN zones deactivation, we generate some traffic to load the database and also in parallel, generate activity on the disks used for applications binaries. In Example 8-80 we observe disks activity using **iostat** monitoring tool at the ASM level.

Example 8-80 iostat output

```
ASMCMD> iostat -G DATA -et
Group_Name  Dsk_Name   Reads         Writes       Read_Err  Write_Err  Read_Time     Write_Time
DATA        DATA_0000  288908775936  57926936064  0         0          16224.238968  26981.640699
DATA        DATA_0001  963327660544  57950595584  0         0          28085.370581  26051.343673
DATA        DATA_0002  323655799296  58444321280  0         0          17990.936824  26781.932068
DATA        DATA_0003  260227339264  57085800448  0         0          15928.859814  25395.34287
```

We deactivate the zones between node r6r4m51 and the active storage DS8k5, as shown in Example 8-81 on page 275.

Example 8-81 Deactivating zones

```
hastk5-12:admin> cfgremove  "stk5_cfg", "r6r4m51_fcs0_ds8k5;r6r4m51_fcs1_ds8k5"
hastk5-12:admin> cfgenable stk5_cfg
You are about to enable a new zoning configuration.
This action will replace the old zoning configuration with the
current configuration selected.
Do you want to enable 'stk5_cfg' configuration  (yes, y, no, n): [no] y
zone config "stk5_cfg" is in effect
Updating flash ...
```

Using the Enterprise Manager, we observe how the database behaves while the disks are swapped in the auxiliary storage. The graphic is shown in Figure 8-8.

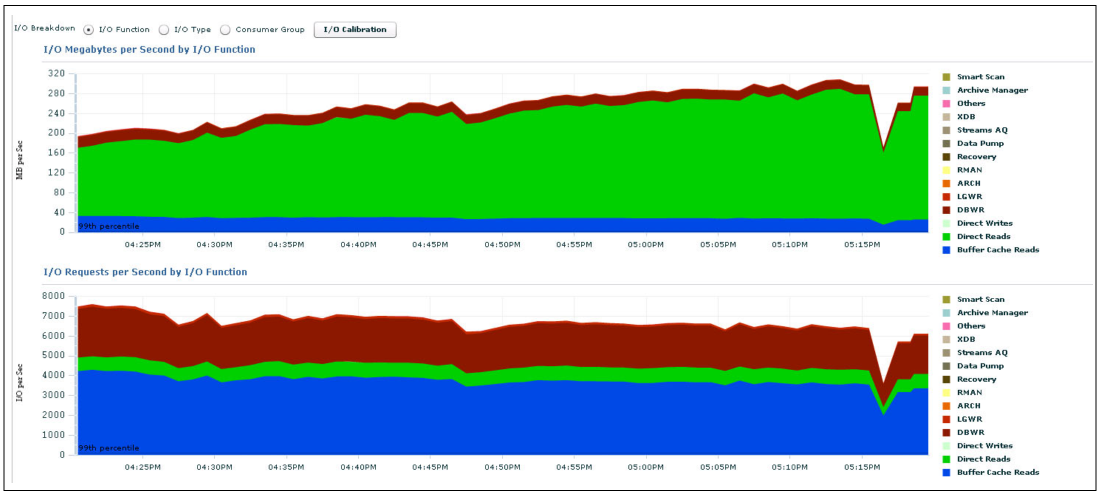


Figure 8-8 Database load and behavior when unplanned swap is performed

We observe what logs were produced after the unplanned swap operation has been triggered. Example 8-82shows the swap operation events logged by syslog in the `/var/hacmp/xd/log/syslog.phake` file.

Example 8-82 Events logged in /var/hacmp/xd/log/syslog.phake

```
Jan 12 17:17:51 r6r4m51 kern:debug unix:     phake_mt.c:  45678617:            xd_lock_release():
Thread '0x2B90019' RELEASED interrupt capable lock. lockP='0xF1000A03E10F3A00' old_intr=11
lockP->saved_intr_level=11
Jan 12 17:17:51 r6r4m51 kern:debug unix:  phake_event.c:  45678617:         process_sfw_event():
Processing SFW Event '0x40000' for MG[ORA_MG 9] @ '0xF1000A03E10F3800'. RDG
E365B2F6-4DEB-E8F1-CDC9-88C3EA77381C
Jan 12 17:17:51 r6r4m51 kern:debug unix:  phake_event.c:  45678617:            post_sfw_action():
Posting Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082B0F0' MG[ORA_MG 9]
RDG[pha_965294144rdg1]
Jan 12 17:17:51 r6r4m51 kern:debug unix: phake_sfwapi.c:  45678617:           sfwpAckPPRCEvent():
Request to Ack a SFW Event with action='PPRC_ACT_DO_NOTHING' for event_handle='0xF1000A05D082B0F0'.
Jan 12 17:17:51 r6r4m51 kern:debug unix:  phake_event.c:  45678617:    get_sfw_function_handle():
Request for returning function handle for methodId = '100'
Jan 12 17:17:51 r6r4m51 kern:debug unix:  phake_event.c:  45678617:    get_sfw_function_handle():
Returning function handle for methodId = '100' func='0x4703A20'
....................<<snippet>..................................
Jan 12 17:17:51 r6r4m51 kern:debug unix:  phake_event.c:  45678617:            post_sfw_action():
Posting of Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082B0F0' MG[ORA_MG 9]
RDG[pha_965294144rdg1] completed with rc=22
Jan 12 17:17:51 r6r4m51 kern:debug unix:  phake_event.c:  45678617:         process_sfw_event():
Processing of SFW Event '0x40000' for MG[ORA_MG 9] @ '0xF1000A03E10F3800' completed with rc=0.
```

```
Jan 12 17:17:51 r6r4m51 kern:debug unix:  phake_event.c:  45678617:         free_xd_event_buf():
Freeing xd_event_t struct. eventP='0xF1000A
03E0641600'
Jan 12 17:17:51 r6r4m51 kern:debug unix:     phake_mt.c:  45678617:           xd_lock_acquire():
Thread '0x2B90019' ACQUIRING interrupt capable lock. lockP='0xF1000000C0785178'
Jan 12 17:17:51 r6r4m51 kern:debug unix:     phake_mt.c:  45678617:           xd_lock_release():
Thread '0x2B90019' RELEASED interrupt capable lock. lockP='0xF1000000C0785178' old_intr=11
lockP->saved_intr_level=11
Jan 12 17:17:51 r6r4m51 kern:debug unix:  phake_event.c:  45678617:         free_xd_event_buf():
Completed freeing xd_event_t struct. num_free_bufs=16
Jan 12 17:17:51 r6r4m51 kern:debug unix:     phake_mt.c:  45678617:           xd_lock_acquire():
Thread '0x2B90019' ACQUIRING interrupt capable lock. lockP='0xF1000A03E10F3A00
```

We verify also the paths of the disks as shown in Example 8-83.

Example 8-83 Disk paths after unplanned HyperSwap

```
root@r6r4m51:/kit> for i in hdisk31 hdisk61 hdisk63 hdisk41 hdisk42;do lspprc -p
$i;done
path       WWNN              LSS   VOL    path
group id                                  group status
=======================================================
0          5005076309ffc5d5  0xc2  0x04    SECONDARY
1(s)       5005076308ffc6d4  0x00  0x04    PRIMARY

path      path  path       parent  connection
group id  id    status
=====================================================================
     0    0     Failed     fscsi0  50050763090b05d5,40c2400400000000
     0    1     Failed     fscsi0  50050763090b45d5,40c2400400000000
     0    2     Failed     fscsi0  50050763090b85d5,40c2400400000000
     0    3     Failed     fscsi1  50050763090b05d5,40c2400400000000
     0    8     Failed     fscsi1  50050763090b45d5,40c2400400000000
     0    9     Failed     fscsi1  50050763090b85d5,40c2400400000000
     1    4     Enabled    fscsi0  50050763085046d4,4000400400000000
     1    5     Enabled    fscsi0  50050763085006d4,4000400400000000
     1    6     Enabled    fscsi1  50050763085046d4,4000400400000000
     1    7     Enabled    fscsi1  50050763085006d4,4000400400000000
path       WWNN              LSS   VOL    path
group id                                  group status
=======================================================
0          5005076309ffc5d5  0xc4  0x04    SECONDARY
1(s)       5005076308ffc6d4  0xe2  0x04    PRIMARY

path      path  path       parent  connection
group id  id    status
=====================================================================
     0    0     Failed     fscsi0  50050763090b05d5,40c4400400000000
     0    1     Failed     fscsi0  50050763090b45d5,40c4400400000000
     0    2     Failed     fscsi0  50050763090b85d5,40c4400400000000
     0    3     Failed     fscsi1  50050763090b05d5,40c4400400000000
     0    8     Failed     fscsi1  50050763090b45d5,40c4400400000000
     0    9     Failed     fscsi1  50050763090b85d5,40c4400400000000
     1    4     Enabled    fscsi0  50050763085046d4,40e2400400000000
     1    5     Enabled    fscsi0  50050763085006d4,40e2400400000000
     1    6     Enabled    fscsi1  50050763085046d4,40e2400400000000
     1    7     Enabled    fscsi1  50050763085006d4,40e2400400000000
path       WWNN              LSS   VOL    path
```

```
group id                                 group status
======================================================
0          5005076309ffc5d5  0xc5  0x00    SECONDARY
1(s)       5005076308ffc6d4  0xe7  0x00    PRIMARY

path      path  path       parent  connection
group id  id    status
======================================================================
     0    0     Failed     fscsi0  50050763090b05d5,40c5400000000000
     0    1     Failed     fscsi0  50050763090b45d5,40c5400000000000
     0    2     Failed     fscsi0  50050763090b85d5,40c5400000000000
     0    3     Failed     fscsi1  50050763090b05d5,40c5400000000000
     0    8     Failed     fscsi1  50050763090b45d5,40c5400000000000
     0    9     Failed     fscsi1  50050763090b85d5,40c5400000000000
     1    4     Enabled    fscsi0  50050763085046d4,40e7400000000000
     1    5     Enabled    fscsi0  50050763085006d4,40e7400000000000
     1    6     Enabled    fscsi1  50050763085046d4,40e7400000000000
     1    7     Enabled    fscsi1  50050763085006d4,40e7400000000000
path       WWNN               LSS   VOL    path
group id                                 group status
======================================================
0          5005076309ffc5d5  0xc3  0x04    SECONDARY
1(s)       5005076308ffc6d4  0x0e  0x04    PRIMARY

path      path  path       parent  connection
group id  id    status
======================================================================
     0    0     Failed     fscsi0  50050763090b05d5,40c3400400000000
     0    1     Failed     fscsi0  50050763090b45d5,40c3400400000000
     0    2     Failed     fscsi0  50050763090b85d5,40c3400400000000
     0    3     Failed     fscsi1  50050763090b05d5,40c3400400000000
     0    8     Failed     fscsi1  50050763090b45d5,40c3400400000000
     0    9     Failed     fscsi1  50050763090b85d5,40c3400400000000
     1    4     Enabled    fscsi0  50050763085046d4,400e400400000000
     1    5     Enabled    fscsi0  50050763085006d4,400e400400000000
     1    6     Enabled    fscsi1  50050763085046d4,400e400400000000
     1    7     Enabled    fscsi1  50050763085006d4,400e400400000000
path       WWNN               LSS   VOL    path
group id                                 group status
======================================================
0          5005076309ffc5d5  0xc3  0x05    SECONDARY
1(s)       5005076308ffc6d4  0x0e  0x05    PRIMARY

path      path  path       parent  connection
group id  id    status
======================================================================
     0    0     Failed     fscsi0  50050763090b05d5,40c3400500000000
     0    1     Failed     fscsi0  50050763090b45d5,40c3400500000000
     0    2     Failed     fscsi0  50050763090b85d5,40c3400500000000
     0    3     Failed     fscsi1  50050763090b05d5,40c3400500000000
     0    8     Failed     fscsi1  50050763090b45d5,40c3400500000000
     0    9     Failed     fscsi1  50050763090b85d5,40c3400500000000
     1    4     Enabled    fscsi0  50050763085046d4,400e400500000000
     1    5     Enabled    fscsi0  50050763085006d4,400e400500000000
     1    6     Enabled    fscsi1  50050763085046d4,400e400500000000
```

```
    1    7      Enabled     fscsi1  50050763085006d4,400e400500000000
```

We find also errors in errpt where the paths are mentioned to failed and also PPRC LUNs failed as shown in Example 8-84.

Example 8-84 Errpt logs

```
DE3B8540  0112171714 P H hdisk63        PATH HAS FAILED
D250CE8D   0112171714 T H hdisk63        PPRC Secondary LUN Failed
DE3B8540   0112171714 P H hdisk63        PATH HAS FAILED
DE3B8540   0112171714 P H hdisk63        PATH HAS FAILED
DE3B8540   0112171714 P H hdisk63        PATH HAS FAILED
DE3B8540   0112171714 P H hdisk41        PATH HAS FAILED
......................<<snippet>..................
```

The recovery procedure for an unplanned swap scenario considering that the root cause for the failed storage has been solved should take into account the disk replication direction since the automatic recovery is only manual.

If the cluster has been restarted, you must manually reverse the disk replication direction, start the cluster and check the clxd.log for completion of the mirror group start. Otherwise, if the cluster was not restarted, you can use the CSPOC menu to swap the disks after all prerequisites for starting mirror group are met (the disks are seen correctly HyperSwap enabled on the system, the disk paths are not failed, disk status is not suspended, etc.).

8.19 System mirror group: Single-node HyperSwap

In this section, we configure a single-node HyperSwap rootvg to be protected from a storage failure by using the system mirror group. The configuration steps are as follows:

- Configure the new disk or disks to be configured for HyperSwap.
- Clone existing rootvg by using the **alt_disk_install** command.
- Reboot the system with the new disk.
- Define the system mirror group and indicate the rootvg volume group which is further protected by HyperSwap. Also indicated is the name of the owner node (here is only single node).
- Verify and synchronize.
- Start PowerHA services.
- Check the clxd.log,hacmp.out.

If the LSSes for the disks configured in the system mirror group overlap with other disks' LSS, disks configured either on another mirror group or disks with the same LSS on your system, then you get the error RC=22 when verification and synchronization is performed.

In our system, we use the pair of disks shown in Example 8-85 on page 279.

Example 8-85 Configuring hdisk84 as for HyperSwap

```
hdisk81          IBM.2107-75LY981/E802
hdisk84          IBM.2107-75NR571/C600
root@r6r4m51:/work>  chdev -l hdisk84 -a san_rep_cfg=migrate_disk -U
hdisk84 changed
```

We clone the existing rootvg using the **alt_disk_install** command as shown in Example 8-86.

Example 8-86 Cloning existing rootvg to hdisk84

```
root@r6r4m51:/> alt_disk_install -C hdisk84
+-----------------------------------------------------------------------------+
ATTENTION: calling new module /usr/sbin/alt_disk_copy. Please see the
alt_disk_copy man page
and documentation for more details.
Executing command: {/usr/sbin/alt_disk_copy -d "hdisk84"}
+-----------------------------------------------------------------------------+
Calling mkszfile to create new /image.data file.
..............................<<snippet>>............................

root@r6r4m51:/> bootlist -om normal
hdisk84 blv=hd5 pathid=0
```

At this time, at the reboot, the system boots up using hdisk84.

We create a new mirror group of type System as shown in Example 8-87.

Example 8-87 Adding a system mirror group

```
                          Add System Mirror Group

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

                                                        [Entry Fields]
* Mirror Group Name                                  [rvg_mg]
  Volume Group(s)                                     rootvg

* HyperSwap                                           Enabled
  Consistency Group                                   Enabled
  Unplanned HyperSwap Timeout (in sec)               [60]
                                                      #
  HyperSwap Priority                                  High
........................<<snippet>>...........................
```

We identified the active paths as shown in Example 8-88.

Example 8-88 Identifying the active paths

```
Manage System Mirror Group

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

```
                                                              [Entry Fields]
* Mirror Group(s)                                           rvg_mg +
* Node Name                                                 r6r4m51 +
* Operation                                                 Show active path +
...........................<<snippet>>..................................

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

r6r4m51: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
r6r4m51: rvg_mg:Site1_primary:Site2_secondary:STG_C
```

We validate the PPRC paths at the AIX level, as shown in Example 8-89.

Example 8-89 Display information about PPRC hdisk84

```
root@r6r4m51:/> lspprc -Ao |egrep 'hdisk84'
hdisk84   Active   0(s)         1           5005076309ffc5d5  5005076308ffc6d4
```

8.19.1 Planned swap system mirror group

We perform a planned swap operation of a system mirror group. We start by writing on the disks and performing the planned swap operation, as shown in Example 8-90.

Example 8-90 Start writing on rootvg hdisk

```
dd if=/dev/zero of=/tmp/4G bs=1M count=4096
```

Then, we use **iostat to** start monitoring the activity of hdisk84, as shown in Example 8-91.

Example 8-91 iostat hdisk84

```
iostat -d 1 |grep hdisk84
Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn
hdisk84         28.0     48772.0     382.0          0     48772
hdisk84        100.0     181888.0     1421.0          0    181888
hdisk84        100.0     175616.0     1372.0          0    175616
hdisk84        100.0     171392.0     1339.0          0    171392
hdisk84         99.0     165632.0     1294.0          0    165632
hdisk84        100.0     168192.0     1314.0          0    168192
hdisk84         25.0     44928.0     351.0          0     44928
hdisk84          0.0       0.0       0.0          0         0
hdisk84         75.0     140220.0     920.0          0    140220
hdisk84         99.0     180000.0     1420.0          4    179996
hdisk84        100.0     171168.0     1345.0          0    171168
hdisk84        100.0     169216.0     1322.0          0    169216
hdisk84        100.0     160256.0     1252.0          0    160256
hdisk84        100.0     186496.0     1457.0          0    186496
hdisk84        100.0     166400.0     1300.0          0    166400
hdisk84         99.0     146744.0     1174.0         40    146704
```

Meanwhile, we perform a swap operation for the system mirror group and verify the `clxd.log`, as shown in Example 8-92 on page 281.

Example 8-92 Swap completed

```
NFO      |2014-01-15T15:26:09.957179|Received XD CLI request = '' (0x1d)
INFO      |2014-01-15T15:26:10.957492|Received XD CLI request = 'Swap Mirror
Group' (0x1c)
INFO      |2014-01-15T15:26:10.957524|Request to Swap Mirror Group 'rvg_mg',
Direction 'Site2_secondary', Outfile ''
ERROR     |2014-01-15T15:26:10.958607|!! Failed to get RG name record from ODM
'HACMPresource'. odmerrno=0 for MG rvg_mg
INFO      |2014-01-15T15:26:10.958630|Not able to find any RG for MG rvg_mg
INFO      |2014-01-15T15:26:10.958888|Not able to find any RAW disks for MG=rvg_mg
INFO      |2014-01-15T15:26:11.091338|Calling sfwGetRepGroupInfo()
INFO      |2014-01-15T15:26:11.091411|sfwGetRepGroupInfo() completed
INFO      |2014-01-15T15:26:11.091606|Calling DO_SWAP
INFO      |2014-01-15T15:26:11.105944|DO_SWAP completed
INFO      |2014-01-15T15:26:11.106095|Swap Mirror Group 'rvg_mg' completed.
```

Therefore, we observe during the swap that, only for one second, the disk was not available, as shown in Example 8-91 on page 280.

8.19.2 Unplanned swap of a system mirror group

In this scenario, we write intensively on the rootvg hdisk and, meanwhile, deactivate the zones between the host and the primary storage.

We start writing on the rootvg hdisk by using the **dd** command, as shown in Example 8-93.

Example 8-93 The dd command writing in /tmp

```
dd if=/dev/zero of=/tmp/15G bs=1M count=15360
```

We deactivate the zone’s communication between host r6r4m51 and the storage DS8K5, as shown in Example 8-94.

Example 8-94 Deactivating the zones communication between the host and storage

```
hastk5-12:admin> zoneremove "r6r4m51__ds8k5", "DS8K5_I0130;
DS8K5_I0131;DS8K5_I0132"
hastk5-12:admin> cfgsave
You are about to save the Defined zoning configuration. This
action will only save the changes on Defined configuration.
Any changes made on the Effective configuration will not
take effect until it is re-enabled.
Do you want to save Defined zoning configuration only?  (yes, y, no, n): [no] y
Updating flash ...
```

We observe the transition status at the writing rate of 240 MB/s, as shown in Example 8-95, and we note that the swap time took place in 25 seconds.

Example 8-95 dd writing on rootvg hdisk

```
hdisk84        75.0     174508.0     349.0          40    174468
hdisk84        99.0     176924.0     375.0          28    176896
hdisk84       100.0     259456.0     537.0           0    259456
hdisk84       100.0     254336.0     510.0           0    254336
hdisk84       100.0     254592.0     516.0           0    254592
```

```
hdisk84        100.0     253824.0     504.0          0    253824
hdisk84        100.0     248832.0     495.0          0    248832
hdisk84        100.0     247816.0     492.0          0    247816
hdisk84        100.0     244736.0     484.0          0    244736
hdisk84        100.0     240128.0     477.0          0    240128
hdisk84        100.0     242540.0     491.0        272    242268
hdisk84        100.0     239616.0     473.0          0    239616
hdisk84        100.0     243200.0     480.0          0    243200
hdisk84        100.0     174128.0     356.0         40    174088
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84         94.0       0.0       0.0          0         0
hdisk84        106.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84         94.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0       0.0       0.0          0         0
hdisk84        100.0     73280.0     155.0        568     72712
hdisk84        100.0     276992.0     544.0          0    276992
hdisk84        100.0     277588.0     555.0        588    277000
hdisk84        100.0     277504.0     546.0          0    277504
hdisk84        100.0     275968.0     542.0          0    275968
hdisk84        100.0     276604.0     560.0         52    276552
hdisk84        100.0     275968.0     543.0          0    275968
hdisk84        100.0     277120.0     544.0          0    277120
hdisk84         64.0     136816.0     606.0       3260    133556
hdisk84         55.0     107856.0     290.0         36    107820
```

We also validate the log, as shown in Example 8-96.

Example 8-96 Log /var/hacmp/xd/log/syslog.phake

```
Jan 16 15:27:20 r6r4m51 kern:debug unix:   phake_swap.c:  23855199:
initiate_swap_mg(): Attempting to initiate a 'Unplanned Swap' operation on
MG[rvg_mg 9] fromSiteId=0 toSiteId=0 mgControllerSetP='0xFFFFFFFF4047CD8'
Jan 16 15:27:20 r6r4m51 kern:debug unix:   phake_swap.c:
23855199:get_swap_here_for_RDG(): Request to get SWAP_HERE flag for
RDG[uuid=9E37585E-55B9-DECC-01FF-483F1EF922E6].

.........................<<snippet>>......................................
```

```
Jan 16 15:27:20 r6r4m51 kern:debug unix: phake_sfwapi.c:  23855199:
sfwpSetRepGroupState(): Request to set State for
RDG[uuid=9E37585E-55B9-DECC-01FF-483F1EF922E6 name=pha_9654781980rdg0] completed
with rc=0
Jan 16 15:27:20 r6r4m51 kern:debug unix:   phake_swap.c:  23855199:
set_state_for_rdg_set(): Attempt to set the state for '1' RDGs included in
MG[rvg_mg 9] to 'PPRC_GS_QUIESCE' completed with rc=0.
```

Also, we validate the paths at the AIX operating system level, as shown in Example 8-97.

Example 8-97 hdisk84 PPRC disk seen at AIX level

```
root@r6r4m51:/> lspprc -p hdisk84
path      WWNN              LSS   VOL    path
group id                                 group status
=======================================================
0         5005076309ffc5d5  0xc6  0x00   SECONDARY
1(s)      5005076308ffc6d4  0xe8  0x02   PRIMARY

path      path  path       parent  connection
group id  id    status
=====================================================================
     0    0     Failed     fscsi0  50050763090b05d5,40c6400000000000
     0    1     Failed     fscsi0  50050763090b45d5,40c6400000000000
     0    2     Failed     fscsi0  50050763090b85d5,40c6400000000000
     0    3     Failed     fscsi1  50050763090b05d5,40c6400000000000
     0    4     Failed     fscsi1  50050763090b45d5,40c6400000000000
     0    5     Failed     fscsi1  50050763090b85d5,40c6400000000000
     1    6     Enabled    fscsi0  50050763085046d4,40e8400200000000
     1    7     Enabled    fscsi0  50050763085006d4,40e8400200000000
     1    8     Enabled    fscsi1  50050763085046d4,40e8400200000000
     1    9     Enabled    fscsi1  50050763085006d4,40e8400200000000
```

8.20 Oracle Real Application Clusters in a HyperSwap environment

Starting with PowerHA SystemMirror 7.1.3, HyperSwap active-active configurations are supported in the Enterprise Edition.

Concurrent workloads across sites, such as Oracle Real Application Clusters (RAC), are supported. Concurrent resource groups are also supported in stretched clusters and linked clusters that are using HyperSwap-enabled mirror groups.

Implementing active-active solutions over an extended distance requires a deep analysis of how the application works and which are the tools, methods, distance, hardware, and software requirements to deliver services without interruption. From hardware and network perspectives, redundancy should be provided for every element that represents a single point of failure, at all levels.

Inter-site communication is critical due to carrying network and storage-replicated data. When inter-site communication is lost, a split-brain situation occurs. To avoid this, be sure to define a decision mechanism that decides where the activity must continue.

Configuring Oracle RAC with PowerHA Enterprise Edition, with disks protected by the HyperSwap function, offers a higher level of protection when stretched clusters are implemented.

HyperSwap relies on the IBM DS8K Metro Mirror Copy Services. A synchronous replication data mechanism is recommended to be used for a *maximum* distance of 100KM. The distance is imposed by the speed light in fibre, which is about 66% of the speed light in a vacuum. Also, many network equipment components can be between the sites. These hops can also add packet processing time, which increases the communication latency. Therefore, when you plan to deploy a stretched cluster, take all network communication parameters into account in terms of latency, bandwidth, and specific equipment configuration, such as buffer credits at the SAN switch level.

PowerHA SystemMirror 7.1.3 Enterprise Edition added unicast heartbeat support. This provides an alternative to the existing multicast heartbeat method. Either heartbeat option can be used within a site, but only unicast is used across sites.

There are many applications that require a specific network setup for deployment, especially when they are meant to be configured in a stretched cluster. To prevent and align a specific network configuration when an application takes advantage of PowerHA SystemMirror HyperSwap protection in a stretched cluster configuration, technologies such Multiprotocol Label Switching (MPLS), Overlay Transport Virtualization, and QFabric (minimizing distance at 80KM between sites) should be taken into account.

In our test configuration, we deploy an Oracle Real Application Cluster (RAC) on a PowerHA SystemMirror Enterprise Edition stretched cluster, with the HyperSwap function enabled, with two sites with two nodes per site. Details and information about Oracle RAC can be found on the Oracle Real Application Clusters page on the Oracle website:

http://www.oracle.com/technetwork/database/options/clustering/overview/index.html

Also see the Oracle white paper titled “Oracle RAC and Oracle RAC One Node on Extended Distance (Stretched) Clusters:”

http://www.oracle.com/technetwork/products/clustering/overview/extendedracversion11-435972.pdf

The requirements for the AIX operating system and the DS88xx microcode level are mentioned in 8.7, “HyperSwap environment requirements” on page 222.

The Oracle RAC version used in our tests is 11.2.0.3 with patch 6 applied (Patch 16083653). It is highly recommended to apply all grid infrastructure and database patches at the latest available and recommended versions.

In this section, we describe various tests of HyperSwap functionality with Oracle RAC in the following scenarios:

- Planned HyperSwap
- Unplanned HyperSwap: Sstorage failure for Site_A but not for Site_B
- Unplanned HyperSwap: Storage from Site_A unavailable for both sites
- Unplanned HyperSwap: Site A failure
- Tie breaker disk consideration in a HyperSwap environment
- CAA dynamic disk addition in a HyperSwap environment
- Online storage migration in ORACLE RAC: HyperSwap

Figure 8-9 shows an example of an Oracle RAC configuration in a HyperSwap environment with two nodes per site.

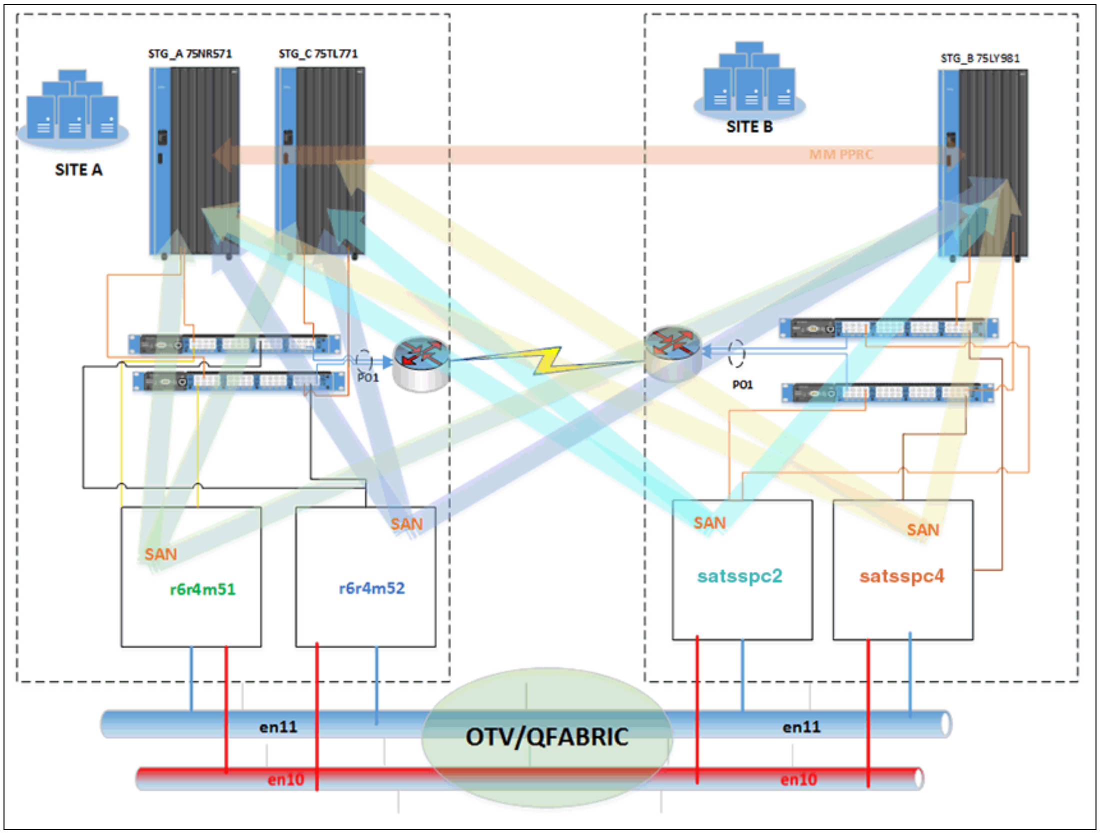


Figure 8-9 Oracle RAC, two sites and two nodes per site

The lines between the LPARs and the storage systems are the SAN zones. The required networks for Oracle RAC are stretched across the sites by using Overlay Transport Virtualization, MPLS, QFabric, and so on. Also, the storage subsystems are configured for IBM Metro Mirror PPRC.

8.20.1 Oracle Real Application Clusters: PowerHA Enterprise Edition stretched cluster configuration

The HyperSwap disks used in our stretched cluster were configured as follows:

- At the Oracle RAC installation, hdisk41, hdisk42, and hdisk61
- Adding hdisk80 and hidks89 as shown in 8.20.2, “Adding new disks to the ASM configuration: Oracle RAC HyperSwap” on page 294.
- The hdisk100 configuration that is described in 8.20.8, “CAA dynamic disk addition in a HyperSwap environment” on page 317

The disk configuration is shown in Example 8-98.

Example 8-98 ASM DISKs and CAA disks active paths

```
root@r6r4m51:/> dsh /work/status_disk_ID.sh |dshbak -c
HOSTS -------------------------------------------------------------------------
r6r4m51.austin.ibm.com
-------------------------------------------------------------------------------
```

```
 hdisk41          IBM.2107-75NR571/C304 ASM_DISK1
 hdisk42          IBM.2107-75NR571/C305 ASM_DISK2
 hdisk61          IBM.2107-75NR571/C404 ASM_DISK3
 hdisk80          IBM.2107-75NR571/C501 ASM_DISK4
 hdisk89          IBM.2107-75NR571/C502 ASM_DISK5
 hdisk100         IBM.2107-75NR571/C901 CAA
HOSTS -------------------------------------------------------------------------
r6r4m52.austin.ibm.com
-------------------------------------------------------------------------------
 hdisk81          IBM.2107-75NR571/C304
 hdisk82          IBM.2107-75NR571/C305
 hdisk83          IBM.2107-75NR571/C404
 hdisk85          IBM.2107-75NR571/C501
 hdisk86          IBM.2107-75NR571/C502
 hdisk94          IBM.2107-75NR571/C901

HOSTS -------------------------------------------------------------------------
satsspc2.austin.ibm.com
-------------------------------------------------------------------------------
 hdisk84          IBM.2107-75NR571/C304
 hdisk85          IBM.2107-75NR571/C305
 hdisk86          IBM.2107-75NR571/C404
 hdisk88          IBM.2107-75NR571/C501
 hdisk89          IBM.2107-75NR571/C502
 hdisk97          IBM.2107-75NR571/C901

HOSTS -------------------------------------------------------------------------
satsspc4.austin.ibm.com
-------------------------------------------------------------------------------
 hdisk83          IBM.2107-75NR571/C304
 hdisk84          IBM.2107-75NR571/C305
 hdisk85          IBM.2107-75NR571/C404
 hdisk87          IBM.2107-75NR571/C501
 hdisk88          IBM.2107-75NR571/C502
 hdisk99          IBM.2107-75NR571/C901

root@r6r4m51:/> cat /work/status_disks.sh
#!/bin/ksh
for i in `/work/lshostvol.sh |egrep 'C304|C305|C404|C501|C502|C901'|awk '{print
$1}'`; do lspprc -Ao|grep $i;done

root@r6r4m51:/> dsh "/work/status_disks.sh" |dshbak -c
HOSTS -------------------------------------------------------------------------
r6r4m51.austin.ibm.com
-------------------------------------------------------------------------------
hdisk41   Active   0(s)        1           5005076309ffc5d5  5005076308ffc6d4
hdisk42   Active   0(s)        1           5005076309ffc5d5  5005076308ffc6d4
hdisk61   Active   0(s)        1           5005076309ffc5d5  5005076308ffc6d4
hdisk80   Active   0(s)        1           5005076309ffc5d5  5005076308ffc6d4
hdisk89   Active   0(s)        1           5005076309ffc5d5  5005076308ffc6d4
hdisk100  Active   0(s)        1           5005076309ffc5d5  5005076308ffc6d4

HOSTS -------------------------------------------------------------------------
r6r4m52.austin.ibm.com
-------------------------------------------------------------------------------
```

```
hdisk81   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk82   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk83   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk85   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk86   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk94   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4

HOSTS -------------------------------------------------------------------------
satsspc2.austin.ibm.com
------------------------------------------------------------------------------
hdisk84   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk85   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk86   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk88   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk89   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk97   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4

HOSTS -------------------------------------------------------------------------
satsspc4.austin.ibm.com
------------------------------------------------------------------------------
hdisk83   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk84   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk85   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk87   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk88   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
hdisk99   Active   0(s)       1          5005076309ffc5d5  5005076308ffc6d4
```

Configuring a stretched cluster for Oracle RAC requires the following configuration steps:

1. Populate `/etc/cluster/rhosts` with corresponding IP addresses.
2. Verify that the clcomd service status is active.
3. Set up cluster, sites, nodes, and networks.
4. Define the repository disk.
5. Define the storage subsystems for each site.
6. Define the mirror groups, taking into account the required fields for HyperSwap enablement and behavior.
7. Define the resource groups and the desired startup policies
8. Modifying resource groups and adding the corresponding Mirror Group relationship
9. Verify and synchronize
10. Start services and bring online resource groups
11. Verifying cluster status and logs

After following these configuration steps, we create the cluster, choosing the appropriate nodes on each site and the type of cluster, as shown in Example 8-99. Then, we start configuring the cluster on the r6r4m51 node.

Example 8-99 Set up cluster, sites, and nodes

```
Setup Cluster, Sites, Nodes and Networks

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

                                                        [Entry Fields]
* Cluster Name                                       [orahyp1]
```

```
* Site 1 Name                                       [SITE_A]
* New Nodes (via selected communication paths)      [r6r4m51.austin.ibm.com
r6r4m52.austin.ibm.com] +

* Site 2 Name                                       [SITE_B]
* New Nodes (via selected communication paths)      [satsspc2.austin.ibm.com
satsspc4.austin.ibm.com]+


  Cluster Type                                      [Stretched Cluster] +
```

We define the repository disk as shown in Example 8-100.

Example 8-100 Defining cluster repository disk

```
            Define Repository Disk and Cluster IP Address

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

                                                       [Entry Fields]
* Cluster Name                                       orahyp1
* Heartbeat Mechanism                                Unicast              +
  Repository Disk                                    00ce123feacbbf49
  Cluster Multicast Address
      (Used only for multicast heartbeat)

Note: Once a cluster has been defined to AIX, all
  that can be modified is the Heartbeat Mechanism

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

We define the storage subsystems, which are attached at our hosts, for both sites (Example 8-101), using fast path **smitty cm_add_strg_system**.

Example 8-101 Adding storage subsystems for Site A and for Site B

```
Site A


   Add a Storage System

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

                                                       [Entry Fields]
* Storage System Name                               [STG_A]
* Site Association                                   SITE_A +
* Vendor Specific Identifier                         IBM.2107-00000NR571    +
* WWNN                                               5005076309FFC5D5       +

..................................<snippet>............................
```

```
Site B

Add a Storage System

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

                                                    [Entry Fields]
* Storage System Name                              [STG_B]
* Site Association                                 SITE_B
+
* Vendor Specific Identifier                        IBM.2107-00000LY981 +
* WWNN                                              5005076308FFC6D4 +
..................................<snippet>...........................
```

We also configure the mirror group, activating the HyperSwap function for the group of disks that is designated for the ASM configuration. Using smitty fast path **smitty cm_cfg_mirr_gps**, we configure the ORAMG user mirror group, as shown in Example 8-102.

Example 8-102 Defining a user mirror group

```
                             Add a User Mirror Group

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

                                                   [Entry Fields]
* Mirror Group Name                                [ORA_MG]
  Volume Group(s)                                                                       +
  Raw Disk(s)       hdisk41:f64bde11-9356-53fe-68bb-6a2aebc647a1 hdisk42:2198648b-a136-2416-d66f-9aa04> +
  HyperSwap                                         Enabled                             +
  Consistency Group                                 Enabled                             +
  Unplanned HyperSwap Timeout (in sec)                      [ 60]                       #
  HyperSwap Priority                               Medium
  Recovery Action                                                  Automatic            +
  Re-sync Action                                   Automatic
```

We maintain the Unplanned HyperSwap Timeout value at the default of 60 seconds. The value represents how long a connection remains unavailable before an unplanned HyperSwap site failover occurs.

Depending on the desired results, the parameter can be lowered to accommodate the environment requirements. For databases, a value of 30 seconds for HyperSwap Timeout is acceptable, taking into account the maximum time allotted for queue full operation.

When multiple disks are configured to be protected by the mirror group, a consistency group parameter should be enabled. Based on the consistency group parameter, HyperSwap with PowerHA SystemMirror reacts as a consistency group-aware application assuring data consistency on the target storage within extend long busy state window.

By default, for Fixed Block extended, a long busy timeout is 60 seconds when the consistency group parameter is enabled at the PPRC path level. Because it is a good practice not to overlap mirror group LSSes, we can also minimize the extended long busy state window on the storage side to 30 seconds, modifying it at the LSS level on both storage repositories (by using **xtndlbztimout**), as shown in Example 8-103 on page 290.

Example 8-103 Changing xtndbztimout for LSS

```
dscli> chlss -pprcconsistgrp enable -extlongbusy 30 C5
dscli> showlss c5
Date/Time: February 1, 2014 9:57:47 AM CST IBM DSCLI Version: 6.6.0.305 DS:
IBM.2107-75NR571
ID             C5
Group          1
addrgrp        C
stgtype        fb
confgvols      6
subsys         0xFFC5
pprcconsistgrp Enabled
xtndlbztimout  30 secs
```

For more information about data consistency in the DS8xxx Metro Mirror Peer-to-Peer Remote Copy, see the IBM Redbooks publication titled *IBM System Storage DS8000 Copy Services for Open Systems*, SG24-6788:

http://www.redbooks.ibm.com/redbooks/pdfs/sg246788.pdf

The next step is to configure the resource group that, practically, will be brought online on all nodes and across the sites as part of the startup policy. The failover policy and fallback policy are shown in Example 8-104.

Example 8-104 Defining the resource group

```
                       Add a Resource Group (extended)

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

                                                        [Entry Fields]
* Resource Group Name                                [ORARG]

  Inter-Site Management Policy                       [Online On Both Sites]    +
* Participating Nodes from Primary Site              [r6r4m51 r6r4m52]         +
  Participating Nodes from Secondary Site            [satsspc4 satsspc2]       +

  Startup Policy                                  Online On All AvailableNodes +
  Fallover Policy                           Bring Offline (On Error Node Only) +
  Fallback Policy                           Never Fallback                     +

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

In the same way as we did on the single-node HyperSwap configuration, we add the configured mirror group to the resource group definition by using fast path: **smitty cm_change_show_rg_resource** → **Change/Show Resources and Attributes for a Resource Group**. We pick from the list ORARG and add the desired Mirror Group and all disks RAW or configured on volume groups in the configuration as shown in Example 8-105 on page 291.

Example 8-105 Adding Mirror Group in Resource configuration

```
         Change/Show All Resources and 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                                    ORARG
  Inter-site Management Policy                           Online On Both Sites
  Participating Nodes from Primary Site                  r6r4m51 r6r4m52
  Participating Nodes from Secondary Site                satsspc4 satsspc2

  Startup Policy                                         Online On All Availab>
  Fallover Policy                                        Bring Offline (On Err>
  Fallback Policy                                        Never Fallback

  Concurrent Volume Groups                              []                      +
  Use forced varyon of volume groups, if necessary       false                  +
  Automatically Import Volume Groups                     false                  +
.............................<snippet>...................................
Raw Disk UUIDs/hdisks                                [2198648b-a136-2416-d6> +

  PPRC Replicated Resources                             []                      +
  Workload Manager Class                                []                      +

  Disk Error Management?                                 no                     +
  Miscellaneous Data                                    []
  SVC PPRC Replicated Resources                         []                      +
  EMC SRDF(R) Replicated Resources                      []                      +
  DS8000 Global Mirror Replicated Resources                                     +
  XIV Replicated Resources                                                      +
  TRUECOPY Replicated Resources                         []                      +
  DS8000-Metro Mirror (In-band) Resources                ORA_MG                 +
[BOTTOM]

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 and synchronize the cluster configuration. If any inconsistencies between the resource group-configured disks and the mirror group-defined disks are detected, an error message appears, and the configuration for the corresponding mirror group and RG should be redone.

After finalizing the cluster configuration, we start cluster services and bring the ORARG resource group online on all available nodes. The resource group status is shown in Example 8-106.

Example 8-106 Resource group availability

```
root@r6r4m51:/> clRGinfo -p -v

Cluster Name: orahyp1

Resource Group Name: ORARG
```

```
Startup Policy: Online On All Available Nodes
Fallover Policy: Bring Offline (On Error Node Only)
Fallback Policy: Never Fallback
Site Policy: Online On Both Sites
Node                          Primary State    Secondary State
---------------------------- ---------------
r6r4m51@SITE_A               ONLINE           OFFLINE
r6r4m52@SITE_A               ONLINE           OFFLINE
satsspc4@SITE_B              ONLINE           OFFLINE
satsspc2@SITE_B              ONLINE           OFFLINE
```

To start our tests, we install and configure Oracle Real Application Cluster on all nodes. The status of the resources in the cluster is shown in Example 8-107.

Example 8-107 Status of Oracle RAC resources

```
root@r6r4m51:/> /u01/app/11.2.0/grid/bin/crsctl stat res -t
--------------------------------------------------------------------------------
NAME           TARGET  STATE        SERVER                   STATE_DETAILS
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.DATA.dg
               ONLINE  ONLINE       r6r4m51
               ONLINE  ONLINE       r6r4m52
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
ora.LISTENER.lsnr
               ONLINE  ONLINE       r6r4m51
               ONLINE  ONLINE       r6r4m52
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
ora.asm
               ONLINE  ONLINE       r6r4m51                  Started
               ONLINE  ONLINE       r6r4m52                  Started
               ONLINE  ONLINE       satsspc2                 Started
               ONLINE  ONLINE       satsspc4                 Started
ora.gsd
               OFFLINE OFFLINE      r6r4m51
               OFFLINE OFFLINE      r6r4m52
               OFFLINE OFFLINE      satsspc2
               OFFLINE OFFLINE      satsspc4
ora.net1.network
               ONLINE  ONLINE       r6r4m51
               ONLINE  ONLINE       r6r4m52
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
ora.ons
               ONLINE  ONLINE       r6r4m51
               ONLINE  ONLINE       r6r4m52
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
ora.registry.acfs
               ONLINE  ONLINE       r6r4m51
               ONLINE  ONLINE       r6r4m52
```

```
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       r6r4m51
ora.cvu
      1        ONLINE  ONLINE       r6r4m52
ora.itsodb.db
      1        ONLINE  ONLINE       r6r4m51                  Open
      2        ONLINE  ONLINE       r6r4m52                  Open
      3        ONLINE  ONLINE       satsspc4                 Open
      4        ONLINE  ONLINE       satsspc2                 Open
ora.oc4j
      1        ONLINE  ONLINE       r6r4m52
ora.r6r4m51.vip
      1        ONLINE  ONLINE       r6r4m51
ora.r6r4m52.vip
      1        ONLINE  ONLINE       r6r4m52
ora.satsspc2.vip
      1        ONLINE  ONLINE       satsspc2
ora.satsspc4.vip
      1        ONLINE  ONLINE       satsspc4
ora.scan1.vip
      1        ONLINE  ONLINE       r6r4m51
```

In our environment, the grid infrastructure and the Oracle database have the application binaries installed on local disks, and the database files on the disks are managed by ASM (Example 8-108).

Example 8-108 ASM disks

```
ASMCMD>  lsdsk -p -G DATA
Group_Num  Disk_Num      Incarn  Mount_Stat  Header_Stat  Mode_Stat  State   Path
        1         0  2515950858  CACHED      MEMBER       ONLINE     NORMAL  /dev/asm_disk1
        1         1  2515950859  CACHED      MEMBER       ONLINE     NORMAL  /dev/asm_disk2
        1         2  2515950860  CACHED      MEMBER       ONLINE     NORMAL  /dev/asm_disk3
        1         6  2515950861  CACHED      MEMBER       ONLINE     NORMAL  /dev/asm_disk4
        1         5  2515950862  CACHED      MEMBER       ONLINE     NORMAL  /dev/asm_disk5
```

The status of the database instances is shown in Example 8-109.

Example 8-109 Status of database instances

```
SQL> select INST_ID,INSTANCE_NUMBER,INSTANCE_NAME,HOST_NAME,DATABASE_STATUS,INSTANCE_ROLE,status from gv$instance;

   INST_ID INSTANCE_NUMBER INSTANCE_NAME    HOST_NAME
DATABASE_STATUS   INSTANCE_ROLE      STATUS
---------- --------------- ---------------- ----------------------------------------------------------------
----------------- ------------------ ------------
         1               1 itsodb1          r6r4m51.austin.ibm.com                                          ACTIVE
PRIMARY_INSTANCE   OPEN
         2               2 itsodb2          r6r4m52.austin.ibm.com                                          ACTIVE
PRIMARY_INSTANCE   OPEN
         4               4 itsodb4          satsspc4.austin.ibm.com                                         ACTIVE
PRIMARY_INSTANCE   OPEN
```

```
      3          3 itsodb3        satsspc2.austin.ibm.com                                  ACTIVE
PRIMARY_INSTANCE   OPEN
```

The itsodb database data files are located in the DATA disk group, which is managed by ASM, as shown in Example 8-110.

Example 8-110 Database data files location

```
SQL> select file_name,TABLESPACE_NAME,ONLINE_STATUS,STATUS from dba_data_files;
+DATA/itsodb/datafile/users.259.837301907
USERS                          ONLINE  AVAILABLE

+DATA/itsodb/datafile/undotbs1.258.837301907 UNDOTBS1                      ONLINE
AVAILABLE

+DATA/itsodb/datafile/sysaux.257.837301907 SYSAUX                        ONLINE  AVAILABLE

+DATA/itsodb/datafile/system.256.837301907 SYSTEM                        SYSTEM  AVAILABLE

+DATA/itsodb/datafile/example.265.837302015 EXAMPLE                       ONLINE
AVAILABLE

+DATA/itsodb/datafile/undotbs2.266.837302379 UNDOTBS2                      ONLINE
AVAILABLE

+DATA/itsodb/itsodf01 ITSOTBLSP                      ONLINE  AVAILABLE

+DATA/itsodb/datafile/undotbs3.271.837628247 UNDOTBS3                      ONLINE
AVAILABLE

+DATA/itsodb/datafile/undotbs4.275.837627065 UNDOTBS4                      ONLINE
AVAILABLE

9 rows selected.
```

8.20.2 Adding new disks to the ASM configuration: Oracle RAC HyperSwap

Bringing new disks into the ASM configuration, as in the case of the single Oracle database instance, requires additional procedures and taking into account the disk configuration for HyperSwap on each Oracle RAC node.

These are the steps for adding new disks:

1. Stop cluster services with the Unmanage option.
2. Restart cluster services.
3. Modify the mirror group adding new disks in the configuration.
4. Update resource group configuration by adding new disks.
5. Verify and synchronize the cluster.
6. Validate in the `clxd.log` that the Mirror Group has been successfully refreshed.
7. Bring resource groups online.

Taking advantage of the Unmanaged HyperSwap level, we put the resource groups in Unmanaged mode by stopping cluster services, as shown in Example 8-111 on page 295.

Example 8-111 Stopping cluster services

```
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                    +
  Stop Cluster Services on these nodes               [satsspc4,r6r4m51,sats> +
  BROADCAST cluster shutdown?                         false                  +
* Select an Action on Resource Groups                 Unmanage Resource Gro> +

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

COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

satsspc4: 0513-044 The clevmgrdES Subsystem was requested to stop.
satsspc4: Jan 23 2014 18:59:36 /usr/es/sbin/cluster/utilities/clstop: called with flags -N -s -f
r6r4m51: 0513-044 The clevmgrdES Subsystem was requested to stop.
r6r4m51: Jan 23 2014 18:59:20 /usr/es/sbin/cluster/utilities/clstop: called with flags -N -s -f
satsspc2: 0513-044 The clevmgrdES Subsystem was requested to stop.
satsspc2: Jan 23 2014 18:59:55 /usr/es/sbin/cluster/utilities/clstop: called with flags -N -s -f
r6r4m52: 0513-044 The clevmgrdES Subsystem was requested to stop.
r6r4m52: Jan 23 2014 19:00:06 /usr/es/sbin/cluster/utilities/clstop: called with flags -N -s -f

root@r6r4m51:/u01/app/11.2.0/grid/bin> clRGinfo
-----------------------------------------------------------------------------
Group Name     State                          Node
-----------------------------------------------------------------------------
ORARG          UNMANAGED                      r6r4m51@SITE_A
               UNMANAGED                      r6r4m52@SITE_A
               UNMANAGED                      satsspc4@SITE_
               UNMANAGED                      satsspc2@SITE_

root@r6r4m51:/u01/app/11.2.0/grid/bin>
```

We add two new disks to the configuration, hdisk80 and hdisk89, from the same LSS, c5, as shown in Example 8-112.

Example 8-112 Adding new disks in the mirror group

```
                    Change/Show a User Mirror Group

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

                                                       [Entry Fields]
  Mirror Group Name                                   ORA_MG
  New Mirror Group Name                              []
  Volume Group(s) +
```

```
  Raw Disk(s)                     [hdisk41:f64bde11-9356-53fe-68bb-6a2aebc647a1
hdisk42:2198648b-a136-2416-d66f-9aa04> +
  Associated Storage System(s)                          STG_A STG_B +
  HyperSwap                                             Enabled +
  Consistency Group                                     Enabled +
  Unplanned HyperSwap Timeout (in sec)                 [60] #
  HyperSwap Priority                                    Medium
  Recovery Action                                       Automatic +
  Re-sync Action                                        Automatic +


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

We modify the corresponding resource group and perform a verify and synchronize cluster configuration. We bring the resource groups online and validate the clxd.log as shown in Example 8-113.

Example 8-113 All five disks appear as being part of ORA_MG mirror group

```
INFO      |2014-01-23T19:50:20.395011|Number of Opaque Attributes Values = '0'
INFO      |2014-01-23T19:50:20.395039|HyperSwap Policy = Enabled
INFO      |2014-01-23T19:50:20.395067|MG Type = user
INFO      |2014-01-23T19:50:20.395096|HyperSwap Priority = medium
INFO      |2014-01-23T19:50:20.395125|Unplanned HyperSwap timeout = 60
INFO      |2014-01-23T19:50:20.395173|Raw Disks = f64bde11-9356-53fe-68bb-6a2aebc647a1
INFO      |2014-01-23T19:50:20.395203|Raw Disks = 2198648b-a136-2416-d66f-9aa04b1d63e6
INFO      |2014-01-23T19:50:20.395233|Raw Disks = 866b8a2f-b746-1317-be4e-25df49685e26
INFO      |2014-01-23T19:50:20.395262|Raw Disks = 46da3c11-6933-2eba-a31c-403f43439a37
INFO      |2014-01-23T19:50:20.395292|Raw Disks = 420f340b-c108-2918-e11e-da985f0f8acd
INFO      |2014-01-23T19:50:20.396019|old_mg_name is: ORA_MG
INFO      |2014-01-23T19:50:20.409919|old_mg_name is: ORA_MG
INFO      |2014-01-23T19:50:20.503417|Successfully changed a Mirror Group 'ORA_MG'
```

When we bring the resource group online, we get the output shown in Example 8-114. Ignore the Failed message, because it is a known problem that will be addressed in a future service pack, but the movement of the resource group is successful.

Example 8-114 Bringing the resource group online

```
Attempting to bring group ORARG online on node ORARG:NONE:r6r4m52.
Attempting to bring group ORARG online on node r6r4m51.
Attempting to bring group ORARG online on node ORARG:NONE:satsspc4.
Attempting to bring group ORARG online on node ORARG:NONE:satsspc2.
No HACMPnode class found with name = ORARG:NONE:r6r4m52
No HACMPnode class found with name = ORARG:NONE:satsspc4
No HACMPnode class found with name = ORARG:NONE:satsspc2
Usage: clRMupdate operation [ object ] [ script_name ] [ reference ]
Failed to queue resource group movement event in the cluster manager.
Usage: clRMupdate operation [ object ] [ script_name ] [ reference ]
Usage: clRMupdate operation [ object ] [ script_name ] [ reference ]
Failed to queue resource group movement event in the cluster manager.
Failed to queue resource group movement event in the cluster manager.

                               COMMAND STATUS
```

```
Command: failed        stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

[MORE...17]

Resource group movement successful.
```

Also in the `clxd.log`, notice that the *CHANGE_MIRROR_GROUP completed* event is logged.

We issue the **mknod** command for disks hdisk80 and hdisk89. Now the disks are protected by PowerHA and can be added to ASM, as shown in Example 8-115.

Example 8-115 Adding ASM disks in the DATA data group

```
SQL> alter diskgroup data add disk '/dev/asm_disk4' name DATA_NEW;
SQL> alter diskgroup data add disk '/dev/asm_disk5' name DATA_0003;
$ asmcmd
ASMCMD> lsdg
State    Type    Rebal  Sector  Block      AU  Total_MB  Free_MB  Req_mir_free_MB
Usable_file_MB  Offline_disks  Voting_files  Name
MOUNTED  EXTERN  N       512   4096  1048576   169984   129620            0
129620             0             Y  DATA/
ASMCMD> lsdsk
Path
/dev/asm_disk1
/dev/asm_disk2
/dev/asm_disk3
/dev/asm_disk4
/dev/asm_disk5
```

8.20.3 Planned HyperSwap: Oracle RAC

The cluster is configured with two mirror groups: CAA_MG for the cluster repository and ORA_MG as the user repository.

We perform the planned HyperSwap test by using Swingbench as the load generator, as shown in Figure 8-10.

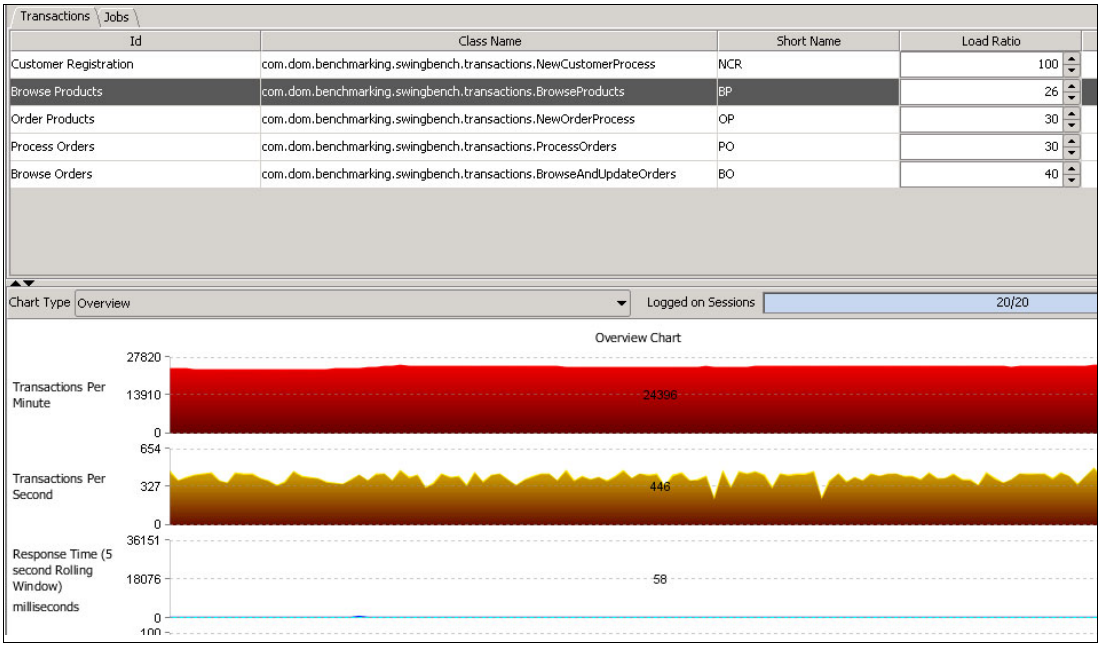


Figure 8-10 Swingbench load during Planned HyperSwap Test

We use 20 users to load the database with mostly writes, reaching almost 5K I/O per second and 23 K transactions per minute. The provided workload is monitored by the Enterprise Control Manager, as shown in Figure 8-11.The disks swap was performed at 05:11 PM. The database load was started at 05:04 PM.

Figure 8-11 Enterprise Control Manager, database Real-Time Performance Monitor

We verify the PowerHA SystemMirror resource group status, as shown in Example 8-116.

Example 8-116 Mirror groups active path status and resource group availability

```
For USER MIRROR Group ORA_MG
COMMAND STATUS

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

```
Before command completion, additional instructions may appear below.

r6r4m51: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
r6r4m51: ORA_MG:SITE_A:SITE_B:STG_A
r6r4m52: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
r6r4m52: ORA_MG:SITE_A:SITE_B:STG_A
satsspc4: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
satsspc4: ORA_MG:SITE_A:SITE_B:STG_A
satsspc2: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
satsspc2: ORA_MG:SITE_A:SITE_B:STG_A


F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next


COMMAND STATUS - For Show Active Path for CAA_MG

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

r6r4m51: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
r6r4m51: CAA_MG:SITE_A:SITE_B:STG_A
r6r4m52: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
r6r4m52: CAA_MG:SITE_A:SITE_B:STG_A
satsspc2: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
satsspc2: CAA_MG:SITE_A:SITE_B:STG_A
satsspc4: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
satsspc4: CAA_MG:SITE_A:SITE_B:STG_A

F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next

root@r6r4m51:/> clRGinfo -v

Cluster Name: orahyp1

Resource Group Name: ORARG
Startup Policy: Online On All Available Nodes
Fallover Policy: Bring Offline (On Error Node Only)
Fallback Policy: Never Fallback
Site Policy: Online On Both Sites
Node                         Primary State    Secondary State
---------------------------- ---------------
r6r4m51@SITE_A               ONLINE           OFFLINE
r6r4m52@SITE_A               ONLINE           OFFLINE
satsspc4@SITE_B              ONLINE           OFFLINE
satsspc2@SITE_B              ONLINE           OFFLINE
```

We also confirm the status of the Oracle RAC resource, as Example 8-107 on page 292 shows.

We start loading the database and perform the swap operation for the ORA_MG mirror group. The operation is logged in the `clxd.log` file. Active paths after the swap are shown in Example 8-117.

Example 8-117 Active path for MG ORA_MG and CAA_MG mirror groups and logged events

```
root@r6r4m51:/> lspprc -Ao |egrep 'hdisk41|hdisk42|hdisk61|hdisk80|hdisk89|hdisk91'
hdisk41   Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk42   Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk61   Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk80   Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk89   Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk91   Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
root@r6r4m51:/>

INFO      |2014-02-05T17:11:15.519122|Received XD CLI request = 'List Mirror Group' (0xc)
root@r6r4m51:/>  tail -f /var/hacmp/xd/log/clxd.log
INFO      |2014-02-05T17:11:15.571763|MG Name='CAA_MG'
.........................<<snippet>>......................................
INFO      |2014-02-05T17:11:15.586003|Printing Storage System Set @(0x20098680)
INFO      |2014-02-05T17:11:15.586023|Num Storage System: '2'
INFO      |2014-02-05T17:11:15.586043|Storage System Name = 'STG_A'
INFO      |2014-02-05T17:11:15.586063|Storage System Name = 'STG_B'
INFO      |2014-02-05T17:11:15.586082|Printing Opaque Attribute Value Set ... @(0x201b095c)
INFO      |2014-02-05T17:11:15.586102|Number of Opaque Attributes Values = '0'
INFO      |2014-02-05T17:11:15.586122|HyperSwap Policy = Enabled
INFO      |2014-02-05T17:11:15.586401|MG Type = user
INFO      |2014-02-05T17:11:15.586664|HyperSwap Priority = medium
INFO      |2014-02-05T17:11:15.586689|Unplanned HyperSwap timeout = 60
INFO      |2014-02-05T17:11:15.586938|Raw Disks = f64bde11-9356-53fe-68bb-6a2aebc647a1
INFO      |2014-02-05T17:11:15.586971|Raw Disks = 2198648b-a136-2416-d66f-9aa04b1d63e6
INFO      |2014-02-05T17:11:15.586998|Raw Disks = 866b8a2f-b746-1317-be4e-25df49685e26
INFO      |2014-02-05T17:11:15.587257|Raw Disks = 8221254f-bf4b-1c0a-31ee-6188b3ca53ac
INFO      |2014-02-05T17:11:15.587294|Raw Disks = a54c9278-42de-babd-6536-1a5b2bfc8d34
INFO      |2014-02-05T17:11:42.459928|Received XD CLI request = '' (0x1d)
INFO      |2014-02-05T17:11:43.464527|Received XD CLI request = 'Swap Mirror Group' (0x1c)
INFO      |2014-02-05T17:11:43.464586|Request to Swap Mirror Group 'ORA_MG', Direction
'SITE_B', Outfile ''
INFO      |2014-02-05T17:11:43.745570|No VG found for MG=ORA_MG
INFO      |2014-02-05T17:11:43.745641|No of VG found for MG ORA_MG
INFO      |2014-02-05T17:11:43.745708|Not able to find any VG disks for MG=ORA_MG
...................................<<snippet>>.............................
INFO      |2014-02-05T17:11:44.109279|Calling DO_SWAP
INFO      |2014-02-05T17:11:45.226644|DO_SWAP completed
INFO      |2014-02-05T17:11:45.227087|Swap Mirror Group 'ORA_MG' completed.

Since the CAA_MG has been performed later we found the swap event in clxd.log
..............................<<snippet>>.................................
INFO      |2014-02-05T17:14:04.832259|Swap Mirror Group 'CAA_MG' completed.
```

The planned HyperSwap operation is now complete. The latency shown during the swap operation is between 1.2 ms and 2.4 ms.

8.20.4 Unplanned HyperSwap: Failure of Storage A nodes in Site A

In this scenario, working on the same cluster configuration as we did for the planned HyperSwap, we simulate a storage failure for Site A only. As such, we modify the zoning configuration and remove Site A nodes connectivity with Storage A.

The expected result of the storage failure in Site A is that all nodes are functional, using the swapped disks on Storage B. This result has the same result as a storage failure in Site A for all nodes.

In this scenario, we load the database by using the Swingbench load generator, after starting it by using the OE benchmark. We capture the events in `hacmp.out`, `clxd.log`, `syslog.caa` and also in the indicated file by using `syslog.conf` and `/var/hacmp/xd/log/syslog.phake` for kernel debugging.

We verify the cluster status again, as well as the disk replication direction, as shown in Example 8-118.

Example 8-118 Identifying the source disks and the resource group status

```
root@r6r4m51:/> lspprc -Ao |egrep
'hdisk41|hdisk42|hdisk61|hdisk80|hdisk89|hdisk91|hdisk100'
hdisk41   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk42   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk61   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk80   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk89   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk91   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk100 Active    0(s)        1          5005076309ffc5d5  5005076308ffc6d4
root@r6r4m51:/> clRGinfo
-----------------------------------------------------------------------------
Group Name     State                        Node
-----------------------------------------------------------------------------
ORARG          ONLINE                       r6r4m51@SITE_A
               ONLINE                       r6r4m52@SITE_A
               ONLINE                       satsspc4@SITE_
               ONLINE                       satsspc2@SITE_
```

We modify the zones between the nodes r6r4m51 and r6r4m52 and the DS5K storage, as shown in Example 8-119.

Example 8-119 Deactivate zones for Storage A with nodes on Site A

```
hastk5-12:admin> cfgremove stk5_cfg","r6r4m51_fcs0_ds8k5;r6r4m51_fcs1_ds8k5;
r6r4m52_fcs0_ds8k5;r6r4m52_fcs1_ds8k5"
hastk5-12:admin> cfgenable stk5_cfg
You are about to enable a new zoning configuration.
This action will replace the old zoning configuration with the
current configuration selected.
Do you want to enable 'stk5_cfg' configuration  (yes, y, no, n): [no] y
zone config "stk5_cfg" is in effect
Updating flash ...
```

We observe the number of transactions that take place while swapping the disks occurs, as shown in Figure 8-12.

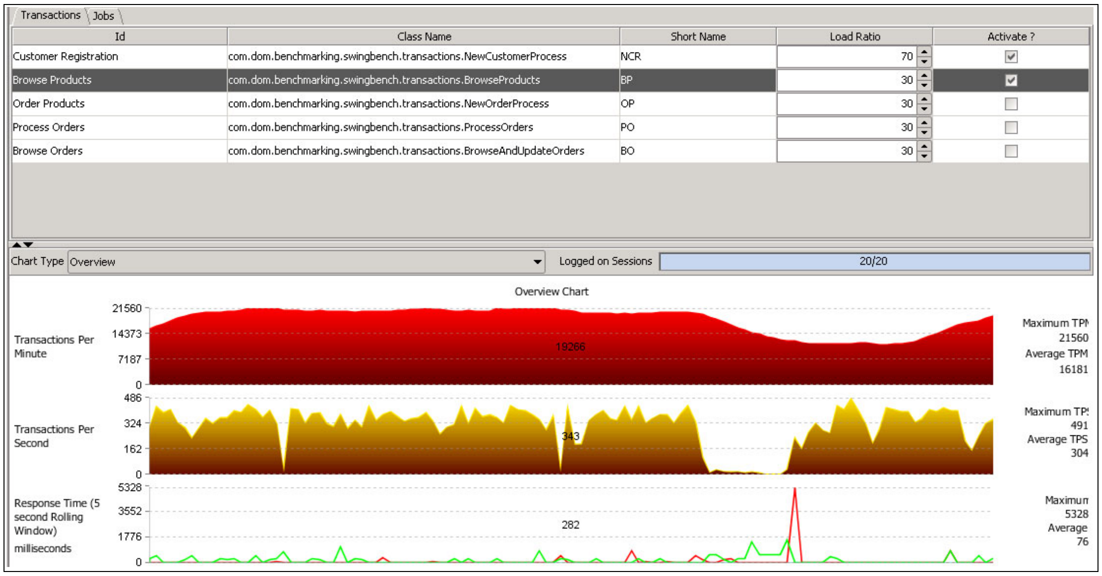


Figure 8-12 Swingbench load monitor

On the Enterprise Control Manager, we validate the continuous load and the latency during the swap, as shown in Figure 8-13.

Figure 8-13 Enterprise Control Manager real time monitoring

We also observe the status of the disk paths, as shown in Example 8-120 on page 303. The paths for nodes r6r4m51 and r6r4m52 to the storage with wwpn 5005076309ffc5d5 are missing, and on satsspc2 and satsspc4 are swapped to Storage B from Site B.

Example 8-120 The active PPRC paths for the disks on all nodes

```
root@r6r4m51:/> lspprc -Ao |egrep
'hdisk41|hdisk42|hdisk61|hdisk80|hdisk89|hdisk91|hdisk100'
hdisk100  Active   0, 1(s)      -1          5005076309ffc5d5,5005076308ffc6d4
hdisk41   Active   0, 1(s)      -1          5005076309ffc5d5,5005076308ffc6d4
hdisk42   Active   0, 1(s)      -1          5005076309ffc5d5,5005076308ffc6d4
hdisk61   Active   0, 1(s)      -1          5005076309ffc5d5,5005076308ffc6d4
hdisk80   Active   0, 1(s)      -1          5005076309ffc5d5,5005076308ffc6d4
hdisk89   Active   0, 1(s)      -1          5005076309ffc5d5,5005076308ffc6d4

root@r6r4m52:/> /work/status_disks.sh
hdisk81   Active   0, 1(s)      -1          5005076309ffc5d5,5005076308ffc6d4
hdisk82   Active   0, 1(s)      -1          5005076309ffc5d5,5005076308ffc6d4
hdisk83   Active   1(s)         0           5005076308ffc6d4  5005076309ffc5d5
hdisk85   Active   0, 1(s)      -1          5005076309ffc5d5,5005076308ffc6d4
hdisk86   Active   0, 1(s)      -1          5005076309ffc5d5,5005076308ffc6d4
hdisk94   Active   1(s)         0           5005076308ffc6d4  5005076309ffc5d5

root@satsspc2:/>  for i in `/work/lshostvol.sh |egrep
'C304|C305|C404|C501|C502|C901'|awk '{print $1}'`; do lspprc -Ao|grep $i;done
hdisk84   Active   1(s)         0           5005076308ffc6d4  5005076309ffc5d5
hdisk85   Active   1(s)         0           5005076308ffc6d4  5005076309ffc5d5
hdisk86   Active   1(s)         0           5005076308ffc6d4  5005076309ffc5d5
hdisk88   Active   1(s)         0           5005076308ffc6d4  5005076309ffc5d5
hdisk89   Active   1(s)         0           5005076308ffc6d4  5005076309ffc5d5
hdisk97   Active   1(s)         0           5005076308ffc6d4  5005076309ffc5d5

root@satsspc4:/work>  for i in `lshostvol.sh |egrep
'C304|C305|C404|C501|C502|C901'|awk '{print $1}'`; do lspprc -Ao|grep $i;done
hdisk83   Active   1(s)         0           5005076308ffc6d4  5005076309ffc5d5
hdisk84   Active   1(s)         0           5005076308ffc6d4  5005076309ffc5d5
hdisk85   Active   1(s)         0           5005076308ffc6d4  5005076309ffc5d5
hdisk87   Active   1(s)         0           5005076308ffc6d4  5005076309ffc5d5
hdisk88   Active   1(s)         0           5005076308ffc6d4  5005076309ffc5d5
hdisk99   Active   1(s)         0           5005076308ffc6d4  5005076309ffc5d5
```

All disks that belong to CAA_MG and ORA_MG were swapped to Storage B. Example 8-121 shows the operation status and significant events that were captured during the swap.

Example 8-121 Events captured for HyperSwap events /var/hacmp/xd/log/syslog.phake

```
root@r6r4m51:/> tail -f /var/hacmp/xd/log/syslog.phake

At this moment the Oracle Rac cluster is configured:

Feb  9 04:11:27 r6r4m51 kern:crit unix:
Feb  9 04:11:27 r6r4m51 kern:crit unix: [Oracle OKS] Node count 4, Local node number 1
Feb  9 04:11:27 r6r4m51 kern:crit unix: ADVMK-00013: Cluster reconfiguration started.
Feb  9 04:11:27 r6r4m51 kern:crit unix:
Feb  9 04:11:30 r6r4m51 kern:crit unix: ADVMK-00014: Cluster reconfiguration completed.
Feb  9 04:11:30 r6r4m51 kern:crit unix:
Feb  9 04:11:30 r6r4m51 kern:crit unix: ADVMK-00014: Cluster reconfiguration completed.
Feb  9 04:11:30 r6r4m51 kern:crit unix:
Feb  9 04:11:30 r6r4m51 kern:crit unix: OKSK-00009: Cluster Membership change setup complete.
Feb  9 04:11:30 r6r4m51 kern:crit unix:

CAA_MG significant logged events
```

```
root@r6r4m51:/> grep CAA_MG /var/hacmp/xd/log/syslog.phake
Feb  9 04:31:14 r6r4m51 kern:debug unix:   phake_swap.c:  44957793:        process_p1_request():
Responder Thread [0x2AE0061] for MG[CAA_MG 10] completed processing of SWAP P1 message with rc=0.
Feb  9 04:31:14 r6r4m51 kern:debug unix:   phake_swap.c:  44957793:           wait_for_p2_msg():
Waiting for P2 message to arrive for MG[CAA_MG 10]. curTime=24152 p2EndTime=24165 timeout=13
Feb  9 04:31:14 r6r4m51 kern:debug unix:   phake_swap.c:  44957793:           wait_for_p2_msg(): P2
processing completed for MG[CAA_MG 10] with rc=0.
Feb  9 04:31:14 r6r4m51 kern:debug unix:   phake_swap.c:  44957793:       resume_mirror_group():
Attempting to Resume 1 RDGs associated with the MG[CAA_MG 10].
Feb  9 04:31:14 r6r4m51 kern:debug unix:   phake_swap.c:  44957793:     set_state_for_rdg_set():
Attempting to set the state for '1' RDGs included in MG[CAA_MG 10] to 'PPRC_GS_RESUME'.
Feb  9 04:31:14 r6r4m51 kern:debug unix:   phake_swap.c:  44957793:     set_state_for_rdg_set():
Attempt to set the state for '1' RDGs included in MG[CAA_MG 10] to 'PPRC_GS_RESUME' completed with
rc=0.
Feb  9 04:31:14 r6r4m51 kern:debug unix:   phake_swap.c:  44957793:       resume_mirror_group():
Operation to Resume 1 RDGs associated with the MG[CAA_MG 10] completed with rc=0
Feb  9 04:32:42 r6r4m51 kern:debug unix:  phake_event.c:  54919315:         process_sfw_event():
Processing SFW Event '0x40000' for MG[CAA_MG 10] @ '0xF1000A03E09E3400'. RDG
EA362798-D04A-AE07-B96E-25D1C36617F7
Feb  9 04:32:42 r6r4m51 kern:debug unix:  phake_event.c:  54919315:           post_sfw_action():
Posting Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082D618' MG[CAA_MG 10]
RDG[pha_10654812011rdg0]
Feb  9 04:32:42 r6r4m51 kern:debug unix:  phake_event.c:  54919315:           post_sfw_action():
[ERROR] Failed to post 'PPRC_ACT_DO_NOTHING' action for event_handle='0xF1000A05D082D618' MG[CAA_MG
10] RDG[pha_10654812011rdg0] sfwpAckPPRCEvent() failed with rc=22.
Feb  9 04:32:42 r6r4m51 kern:debug unix:  phake_event.c:  54919315:           post_sfw_action():
Posting of Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082D618' MG[CAA_MG 10]
RDG[pha_10654812011rdg0] completed with rc=22
Feb  9 04:32:42 r6r4m51 kern:debug unix:  phake_event.c:  54919315:         process_sfw_event():
Processing of SFW Event '0x40000' for MG[CAA_MG 10] @ '0xF1000A03E09E3400' completed with rc=0.

ORA_MG significant logged events

root@r6r4m51:/> grep ORA_MG /var/hacmp/xd/log/syslog.phake
Feb  9 04:31:13 r6r4m51 kern:debug unix:   phake_swap.c:  46137489: passive_swap_preprocessing(): Swap
state is 'PPRC_SS_SWAPPABLE' for MG[ORA_MG 9].
Feb  9 04:31:13 r6r4m51 kern:debug unix:   phake_swap.c:  46137489: passive_swap_preprocessing(): Swap
Pre-Processing for MG[ORA_MG 9] completed with rc=0. resp=1 reason=0
Feb  9 04:31:14 r6r4m51 kern:debug unix:   phake_swap.c:  46137489:        process_p1_request():
Responder Thread [0x2C00091] for MG[ORA_MG 9] completed processing of SWAP P1 message with rc=0.
Feb  9 04:31:14 r6r4m51 kern:debug unix:   phake_swap.c:  46137489:           wait_for_p2_msg():
Waiting for P2 message to arrive for MG[ORA_MG 9]. curTime=24152 p2EndTime=24155 timeout=3
Feb  9 04:31:14 r6r4m51 kern:debug unix:   phake_swap.c:  46137489:           wait_for_p2_msg(): P2
processing completed for MG[ORA_MG 9] with rc=0.
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:         process_sfw_event():
Processing SFW Event '0x40000' for MG[ORA_MG 9] @ '0xF1000A03E09EF000'. RDG
DCCA8F95-0A0F-6CE6-3FA7-9F8CD5948071
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:           post_sfw_action():
Posting Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082C4E0' MG[ORA_MG 9]
RDG[pha_9654751954rdg0]
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:           post_sfw_action():
[ERROR] Failed to post 'PPRC_ACT_DO_NOTHING' action for event_handle='0xF1000A05D082C4E0' MG[ORA_MG 9]
RDG[pha_9654751954rdg0] sfwpAckPPRCEvent() failed with rc=22.
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:           post_sfw_action():
Posting of Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082C4E0' MG[ORA_MG 9]
RDG[pha_9654751954rdg0] completed with rc=22
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:         process_sfw_event():
Processing of SFW Event '0x40000' for MG[ORA_MG 9] @ '0xF1000A03E09EF000' completed with rc=0.
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:         process_sfw_event():
Processing SFW Event '0x80000' for MG[ORA_MG 9] @ '0xF1000A03E09EF000'. RDG
DCCA8F95-0A0F-6CE6-3FA7-9F8CD5948071
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:           post_sfw_action():
Posting Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082C6B0' MG[ORA_MG 9]
RDG[pha_9654751954rdg0]
```

```
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:            post_sfw_action():
[ERROR] Failed to post 'PPRC_ACT_DO_NOTHING' action for event_handle='0xF1000A05D082C6B0' MG[ORA_MG 9]
RDG[pha_9654751954rdg0] sfwpAckPPRCEvent() failed with rc=22.
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:            post_sfw_action():
Posting of Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082C6B0' MG[ORA_MG 9]
RDG[pha_9654751954rdg0] completed with rc=22
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:          process_sfw_event():
Processing of SFW Event '0x80000' for MG[ORA_MG 9] @ '0xF1000A03E09EF000' completed with rc=0.
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:          process_sfw_event():
Processing SFW Event '0x40000' for MG[ORA_MG 9] @ '0xF1000A03E09EF000'. RDG
236498A6-A470-736C-C1AC-C23E4BB7B222
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:            post_sfw_action():
Posting Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082CB38' MG[ORA_MG 9]
RDG[pha_9654771971rdg2]
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:            post_sfw_action():
[ERROR] Failed to post 'PPRC_ACT_DO_NOTHING' action for event_handle='0xF1000A05D082CB38' MG[ORA_MG 9]
RDG[pha_9654771971rdg2] sfwpAckPPRCEvent() failed with rc=22.
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:            post_sfw_action():
Posting of Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082CB38' MG[ORA_MG 9]
RDG[pha_9654771971rdg2] completed with rc=22
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:          process_sfw_event():
Processing of SFW Event '0x40000' for MG[ORA_MG 9] @ '0xF1000A03E09EF000' completed with rc=0.
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:          process_sfw_event():
Processing SFW Event '0x80000' for MG[ORA_MG 9] @ '0xF1000A03E09EF000'. RDG
236498A6-A470-736C-C1AC-C23E4BB7B222
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:            post_sfw_action():
Posting Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082CD08' MG[ORA_MG 9]
RDG[pha_9654771971rdg2]
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:            post_sfw_action():
[ERROR] Failed to post 'PPRC_ACT_DO_NOTHING' action for event_handle='0xF1000A05D082CD08' MG[ORA_MG 9]
RDG[pha_9654771971rdg2] sfwpAckPPRCEvent() failed with rc=22.
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:              post_sfw_action():
Posting of Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082CD08' MG[ORA_MG 9]
RDG[pha_9654771971rdg2] completed with rc=22
Feb  9 04:31:14 r6r4m51 kern:debug unix:  phake_event.c:  14680081:          process_sfw_event():
Processing of SFW Event '0x80000' for MG[ORA_MG 9] @ '0xF1000A03E09EF000' completed with rc=0.
Feb  9 04:31:14 r6r4m51 kern:debug unix:   phake_swap.c:  46137489:        resume_mirror_group():
Attempting to Resume 3 RDGs associated with the MG[ORA_MG 9].
Feb  9 04:31:14 r6r4m51 kern:debug unix:   phake_swap.c:  46137489:      set_state_for_rdg_set():
Attempting to set the state for '3' RDGs included in MG[ORA_MG 9] to 'PPRC_GS_RESUME'.
Feb  9 04:31:14 r6r4m51 kern:debug unix:   phake_swap.c:  46137489:      set_state_for_rdg_set():
Attempt to set the state for '3' RDGs included in MG[ORA_MG 9] to 'PPRC_GS_RESUME' completed with
rc=0.
Feb  9 04:31:14 r6r4m51 kern:debug unix:   phake_swap.c:  46137489:        resume_mirror_group():
Operation to Resume 3 RDGs associated with the MG[ORA_MG 9] completed with rc=0
Feb  9 04:32:05 r6r4m51 kern:debug unix:  phake_event.c:  14680081:          process_sfw_event():
Processing SFW Event '0x40000' for MG[ORA_MG 9] @ '0xF1000A03E09EF000'. RDG
DCCA8F95-0A0F-6CE6-3FA7-9F8CD5948071
Feb  9 04:32:05 r6r4m51 kern:debug unix:  phake_event.c:  14680081:            post_sfw_action():
Posting Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082CED8' MG[ORA_MG 9]
RDG[pha_9654751954rdg0]
Feb  9 04:32:05 r6r4m51 kern:debug unix:  phake_event.c:  14680081:            post_sfw_action():
[ERROR] Failed to post 'PPRC_ACT_DO_NOTHING' action for event_handle='0xF1000A05D082CED8' MG[ORA_MG 9]
RDG[pha_9654751954rdg0] sfwpAckPPRCEvent() failed with rc=22.
Feb  9 04:32:05 r6r4m51 kern:debug unix:  phake_event.c:  14680081:            post_sfw_action():
Posting of Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082CED8' MG[ORA_MG 9]
RDG[pha_9654751954rdg0] completed with rc=22
Feb  9 04:32:05 r6r4m51 kern:debug unix:  phake_event.c:  14680081:          process_sfw_event():
Processing of SFW Event '0x40000' for MG[ORA_MG 9] @ '0xF1000A03E09EF000' completed with rc=0.
Feb  9 04:32:05 r6r4m51 kern:debug unix:  phake_event.c:  14680081:          process_sfw_event():
Processing SFW Event '0x40000' for MG[ORA_MG 9] @ '0xF1000A03E09EF000'. RDG
236498A6-A470-736C-C1AC-C23E4BB7B222
Feb  9 04:32:05 r6r4m51 kern:debug unix:  phake_event.c:  14680081:            post_sfw_action():
Posting Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082D0A8' MG[ORA_MG 9]
RDG[pha_9654771971rdg2]
```

```
Feb  9 04:32:05 r6r4m51 kern:debug unix:  phake_event.c:  14680081:          post_sfw_action():
[ERROR] Failed to post 'PPRC_ACT_DO_NOTHING' action for event_handle='0xF1000A05D082D0A8' MG[ORA_MG 9]
RDG[pha_9654771971rdg2] sfwpAckPPRCEvent() failed with rc=22.
Feb  9 04:32:05 r6r4m51 kern:debug unix:  phake_event.c:  14680081:          post_sfw_action():
Posting of Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082D0A8' MG[ORA_MG 9]
RDG[pha_9654771971rdg2] completed with rc=22
Feb  9 04:32:05 r6r4m51 kern:debug unix:  phake_event.c:  14680081:        process_sfw_event():
Processing of SFW Event '0x40000' for MG[ORA_MG 9] @ '0xF1000A03E09EF000' completed with rc=0.
Feb  9 04:32:10 r6r4m51 kern:debug unix:  phake_event.c:  14680081:        process_sfw_event():
Processing SFW Event '0x40000' for MG[ORA_MG 9] @ '0xF1000A03E09EF000'. RDG
1D4EC611-59F9-0A6F-6FB0-2D01D531A6F3
Feb  9 04:32:10 r6r4m51 kern:debug unix:  phake_event.c:  14680081:          post_sfw_action():
Posting Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082D278' MG[ORA_MG 9]
RDG[pha_9654761964rdg1]
Feb  9 04:32:10 r6r4m51 kern:debug unix:  phake_event.c:  14680081:          post_sfw_action():
[ERROR] Failed to post 'PPRC_ACT_DO_NOTHING' action for event_handle='0xF1000A05D082D278' MG[ORA_MG 9]
RDG[pha_9654761964rdg1] sfwpAckPPRCEvent() failed with rc=22.
Feb  9 04:32:10 r6r4m51 kern:debug unix:  phake_event.c:  14680081:          post_sfw_action():
Posting of Action 'PPRC_ACT_DO_NOTHING' to SFW for event_handle='0xF1000A05D082D278' MG[ORA_MG 9]
RDG[pha_9654761964rdg1] completed with rc=22
Feb  9 04:32:10 r6r4m51 kern:debug unix:  phake_event.c:  14680081:        process_sfw_event():
Processing of SFW Event '0x40000' for MG[ORA_MG 9] @ '0xF1000A03E09EF000' completed with rc=0.
```

Note: For an unplanned HyperSwap, the `clxd.log` does not record the events.

We validate the active paths from the C-SPOC by using fast path, as shown in Example 8-122:

`smitty -C cm_user_mirr_gp`

Example 8-122 Showing active paths for the ORA_MG mirror group

```
COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

r6r4m51: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
r6r4m51: ORA_MG:SITE_B:SITE_A:STG_B
r6r4m52: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
r6r4m52: ORA_MG:SITE_B:SITE_A:STG_B
satsspc4: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
satsspc4: ORA_MG:SITE_B:SITE_A:STG_B
satsspc2: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
satsspc2: ORA_MG:SITE_B:SITE_A:STG_B


F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next
```

8.20.5 Unplanned HyperSwap: Storage A unavailable for both sites

In this scenario, Storage A in Site A becomes unavailable for all nodes on our stretched cluster. The expected operation is to have all nodes up and the disks swapped on Storage B, as it happened for the storage failure for Site A. The workload used in this scenario intensively writes to the ACFS file system during the storage failure.

Example 8-122 on page 306 shows our configuration starting point. We restore the environment at initial configuration with all disks and paths available by activating the zones and performing the swap operation back to Site A.

After activating the zones, we perform the operation for both mirror groups, as shown in Example 8-123.

Example 8-123 Refresh mirror group operation

```
                              Manage User Mirror Group(s)

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

                                                        [Entry Fields]
* Mirror Group(s)                                     ORA_MG              +
* Operation                                           Refresh             +




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 validate the disk configuration and replication direction. In this scenario, we expect to have only the disks of Storage A swapping to Storage B, as in previous scenarios, and we write directly to the Oracle ACFS file system. The ACFS file system configuration is shown in Example 8-124.

Example 8-124 ACFS file system

```
root@r6r4m51:/> acfsutil registry
Mount Object:
  Device: /dev/asm/asmfsu02-29
  Mount Point: /u02
  Disk Group: DATA
  Volume: ASMFSU02
  Options: none
  Nodes: all

root@r6r4m51:/> dsh mount |grep u02
r6r4m51.austin.ibm.com:         /dev/asm/asmfsu02-29 /u02             acfs   Feb
08 22:19 rw
r6r4m52.austin.ibm.com:         /dev/asm/asmfsu02-29 /u02             acfs   Feb
08 22:20 rw
satsspc2.austin.ibm.com:         /dev/asm/asmfsu02-29 /u02            acfs   Feb
08 22:20 rw
satsspc4.austin.ibm.com:         /dev/asm/asmfsu02-29 /u02            acfs   Feb
09 04:12 rw
root@r6r4m51:/>

ASMCMD> volinfo -a
Diskgroup Name: DATA

         Volume Name: ASMFSU02
         Volume Device: /dev/asm/asmfsu02-29
```

```
State: ENABLED
Size (MB): 20480
Resize Unit (MB): 32
Redundancy: UNPROT
Stripe Columns: 4
Stripe Width (K): 128
Usage: ACFS
Mountpath: /u02
```

The Cluster Synchronization Services (CSS) heartbeat values set in our test system are shown in Example 8-125.

Example 8-125 CSS heartbeat values

```
root@r6r4m51:/> /u01/app/11.2.0/grid/bin/crsctl get css misscount
CRS-4678: Successful get misscount 100 for Cluster Synchronization Services.
root@r6r4m51:/> /u01/app/11.2.0/grid/bin/crsctl get css disktimeout
CRS-4678: Successful get disktimeout 200 for Cluster Synchronization Services.
```

We start writing on the ACFS file system, as shown in Example 8-126, and start **iostat** for the hdisk80 disk.

Example 8-126 Writing on ACFS /u02

```
root@r6r4m51:/> dd if=/dev/zero of=/u02/15G bs=32k count=491520 &
```

We deactivate the zones for the DS5k storage for all nodes, as shown in Example 8-127.

Example 8-127 Deactivate zones for DS5k storage

```
hastk5-12:admin> cfgremove  "stk5_cfg",
r6r4m51_fcs0_ds8k5;r6r4m51_fcs1_ds8k5;r6r4m52_fcs0_ds8k5;
satsspc2_fcs0_ds8k5;satsspc4_fcs0_ds8k5;r6r4m52_fcs1_ds8k5"
hastk5-12:admin> cfgenable stk5_cfg
You are about to enable a new zoning configuration.
This action will replace the old zoning configuration with the
current configuration selected.
Do you want to enable 'stk5_cfg' configuration  (yes, y, no, n): [no] y
zone config "stk5_cfg" is in effect
Updating flash ...
```

Example 8-128 shows the **iostat** output. The written kilobytes become “0” when the zone deactivation is detected.

Example 8-128 iostat output

```
root@r6r4m51:/> iostat -d 1 |grep hdisk80
Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn
hdisk80        100.0     146756.0     7761.0        84    146672
hdisk80         99.0     113354.0     6894.0        22    113332
hdisk80        100.0     125748.0     7353.0        36    125712
hdisk80         99.0     136020.0     7725.0        52    135968
hdisk80        100.0     112967.0     7071.0       108    112859
hdisk80         99.0     148212.0     7867.0        20    148192
hdisk80        100.0     110772.0     6996.0        36    110736
hdisk80        100.0     140738.0     7529.0        22    140716
```

```
hdisk80       100.0     110812.0     7083.0        52    110760
hdisk80       100.0     139660.0     7852.0        20    139640
hdisk80       100.0     113948.0     6907.0        36    113912
hdisk80        98.0     128774.0     7582.0        22    128752
hdisk80       100.0     127804.0     7330.0        68    127736
hdisk80       100.0     126088.0     7512.0        20    126068
hdisk80        98.0     129623.0     7462.0       108    129515
hdisk80       100.0       0.0       0.0          0         0
hdisk80       100.0       0.0       0.0          0         0
hdisk80       100.0       0.0       0.0          0         0
hdisk80       100.0       0.0       0.0          0         0
hdisk80       100.0       0.0       0.0          0         0
hdisk80       100.0       0.0       0.0          0         0
hdisk80       100.0       0.0       0.0          0         0
hdisk80       100.0       0.0       0.0          0         0
hdisk80       100.0       0.0       0.0          0         0
...............<<snippet, see the comment >>...............
hdisk80         0.0       0.0       0.0          0         0
hdisk80         0.0       0.0       0.0          0         0
hdisk80         0.0       0.0       0.0          0         0
hdisk80         0.0       0.0       0.0          0         0
hdisk80        57.0     38066.0     3488.0       79     37987
hdisk80        79.0     73376.0     5054.0       44     73332
hdisk80       100.0     120460.0     6163.0        36    120424
hdisk80       100.0     34582.0     1886.0       66     34516
hdisk80       100.0     115780.0     6744.0        52    115728
hdisk80        99.0     139168.0     7650.0        20    139148
hdisk80        16.0     18838.0     1135.0       66     18772
hdisk80        93.0     112936.0     6611.0        36    112900
hdisk80       100.0     132155.0     7479.0        72    132083
hdisk80        96.0     119012.0     7079.0        36    118976
hdisk80        99.0     124910.0     7109.0        38    124872
hdisk80       100.0     129992.0     7070.0        20    129972
hdisk80       100.0     115836.0     6971.0        36    115800
hdisk80       100.0     131974.0     7321.0        38    131936
hdisk80        98.0     105633.0     6908.0        84    105549
hdisk80       100.0     139092.0     7470.0        20    139072
hdisk80        99.0     120988.0     7086.0        40    120948
hdisk80       100.0     139274.0     7613.0        22    139252
hdisk80       100.0     118337.0     6999.0       106    118231
hdisk80        99.0     136420.0     7442.0        20    136400
hdisk80       100.0     112083.0     6931.0       259    111824
hdisk80        99.0     124606.0     7408.0       582    124024
hdisk80        99.0     119176.0     7218.0        40    119136
hdisk80       100.0     135352.0     7325.0        68    135284
hdisk80        76.0     92004.0     5302.0       36     91968
hdisk80        46.0     52877.0     3328.0       74     52803
```

We count 79 seconds that the ASM disks were not available. The writing rate is more than 105 MB/s.

Consulting the log, we verify the start and end swap time for every mirror group defined in our cluster, as shown in Example 8-129 on page 310.

Example 8-129 /var/hacmp/xd/log/syslog.phake

```
For ORA_MG Mirror Group:

Feb  9 11:04:26 r6r4m51 kern:debug unix:   phake_swap.c:  23199887:
process_p1_request(): Responder Thread [0x162008F] for MG[ORA_MG 9] processing SWAP P1
message. p1EndTime=5349 p2EndTime=5339
Feb  9 11:04:26 r6r4m51 kern:debug unix:   phake_swap.c:  23199887:
passive_swap_preprocessing(): Running Passive Swap Pre-Processing for MG[ORA_MG 9]
p1EndTime=5349 p2EndTime=5339
...............................<<snippet>>...................................
Feb  9 11:05:39 r6r4m51 kern:debug unix:   phake_swap.c:  23199887:
set_state_for_rdg_set(): Attempt to set the state for '3' RDGs included in MG[ORA_MG 9] to
'PPRC_GS_RESUME' completed with rc=0.
Feb  9 11:05:39 r6r4m51 kern:debug unix:   phake_swap.c:  23199887:
resume_mirror_group(): Operation to Resume 3 RDGs associated with the MG[ORA_MG 9]
completed with rc=0


For CAA_MG Mirror Group:

Feb  9 11:04:19 r6r4m51 kern:debug unix:  phake_event.c:  39911549:
process_sfw_event(): Processing SFW Event '0x100' for MG[CAA_MG 10] @ '0xF1000A03E0E05C00'.
RDG EA362798-D04A-AE07-B96E-25D1C36617F7
.................................<<snippet>..................................
Feb  9 11:05:26 r6r4m51 kern:debug unix:   phake_swap.c:  41877543:
resume_mirror_group(): Operation to Resume 1 RDGs associated with the MG[CAA_MG 10]
completed with rc=0
```

Also, the `ocssd.log` shows the time during which there were missed disk heartbeats, as shown Example 8-130.

Example 8-130 ocssd.log disk ping log

```
2014-02-09 11:05:19.903: [    CSSD][3862]clssnmSendingThread: sending status msg
to all nodes
2014-02-09 11:05:19.903: [    CSSD][3862]clssnmSendingThread: sent 4 status msgs
to all nodes
2014-02-09 11:05:22.755: [    CSSD][4376]clssscMonitorThreads
clssnmvDiskPingThread not scheduled for 76743 msecs
```

The commands **`lspprc`** or smitty fast path **smitty cm_mng_mirror_groups** → **Manage User Mirror Group(s)** can be used to identify where the active paths are pointed to. Using the SMIT menu, we see that all active disk paths appear as active in Site B, as shown in Example 8-131.

Example 8-131 Active disk paths after swap

```
COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

r6r4m51: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
r6r4m51: ORA_MG:SITE_B:SITE_A:STG_B
r6r4m52: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
r6r4m52: ORA_MG:SITE_B:SITE_A:STG_B
satsspc4: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
```

```
satsspc4: ORA_MG:SITE_B:SITE_A:STG_B
satsspc2: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
satsspc2: ORA_MG:SITE_B:SITE_A:STG_B


F1=Help                    F2=Refresh                  F3=Cancel
F6=Command
F8=Image                    F9=Shell                    F10=Exit
/=Find
n=Find Next
```

8.20.6 Tie breaker considerations: Oracle RAC in a HyperSwap environment

The tie breaker mechanism is used to determine which partitioned site is allowed to continue to operate when a cluster split event occurs. Each partition attempts to acquire the tie breaker by placing a lock on the tie breaker disk.

The tie breaker disk has the following requirements and restrictions:

- SCSI-3 persistent reservation support is required for I/O fencing. Technologies such iSCSI, SCSi, or FCoE are supported.
- The disk must be accessible on all cluster nodes.
- The CAA repository disk cannot be used as tie breaker.
- Oracle RAC disks cannot be used as tie breakers.
- A third location is required.

PowerHA SystemMirror stretched cluster configuration takes advantage of all CAA cluster communication mechanisms through these channels:

- IP network
- SAN fabric
- Repository disk

Providing a high level of redundancy for all components and devices that are part of the cluster configuration and eliminating all single points of failure are recommended. For example, all network interface cards per network type are in the cluster node, there are communication links between sites, and the network devices are redundant.

Nevertheless, when all communication between sites is lost, the cluster mechanisms determine how the cluster reacts. In a HyperSwap environment, only PowerHA SystemMirror determines whether the disks must be swapped to the auxiliary storage, based on the split and merge policies.

Either of these can be PowerHA SystemMirror split handling policies:

None
: The cluster will take no action.

Tie breaker
: PowerHA reboots the nodes on the site where the tie breaker disk is not reachable.

Merge policy can be based on these alternatives:

Majority
: In case of a merge, the site with the larger number of nodes survives.

Tie breaker
: In case of a merge, the site that reaches the tie breaker disk survives.

In case of site and also source storage failure (where the disks are sources for Metro Mirror replication), PowerHA determines whether there is a split brain situation and reacts with a site-down event, based on the defined split policy. Without a tie breaker disk, the split-handling policy takes no action. Therefore, on the secondary site, the disks are not be swapped by rebooting the nodes because the CAA repository cannot be accessed. Also, the Oracle disks are not available. After the nodes reboot, if the disks on the primary site are not seen, the application can be started or it starts automatically.

When a site failure event occurs and the Metro Mirror source disks are located on the storage on the same site that failed, if the split policy defined is None, the messages from Example 8-132 appear in the sysphake log. The nodes will be rebooted on the survival site.

Example 8-132 Split Policy Handling when is set None

```
kern:debug unix:  phake_event.c:  24510569:         process_sfw_event(): [ERROR]
Failed to process unplanned swap request for MG[CAA_MG 10]. rc=-1
process_sfw_event(): [ERROR] Failed to process unplanned swap request for
MG[ORA_MG 9]. rc=-1
```

It is highly recommended that you use the tie breaker disk for any application that is configured in a stretched cluster, in addition to the hardware redundancy that is required for such an implementation.

Warning messages also appear during the verify and synchronize operation, for example:

> The possibility of cluster/site partition can be minimized by adding redundancy in communication paths and eliminating all single-point-of-failures.

In PowerHA, it is easy to configure the policies for how the cluster will behave when a split and merge event takes place. Use either of these fast paths for configuring the tie breaker disk:

smitty -C cm_cluster_split_merge

or

smitty sysmirror → **Custom Cluster Configuration** → **Cluster Nodes and Networks** → **Initial Cluster Setup (Custom)** → **Configure Cluster Split and Merge Policy**.

8.20.7 Unplanned HyperSwap: Site A failure, Oracle RAC

In this scenario, we simulate a site failure by forcibly deactivating the nodes' zones to the DS5K storage and by using the Hardware Management Console (HMC) to shut down the LPARs on Site A. Site A is brought down while controlled, so this could be a cluster split-brain situation, and the cluster must decide which site is the survival site. In this configuration, we use the tie breaker disk, configured from a third storage repository, as shown in Figure 8-14 on page 323.

We configure the Split and Merge PowerHA policies as shown in Example 8-133 on page 313, indicating the tie breaker disk. The disk must be seen on all cluster nodes.

Example 8-133 Defining split and merge policies

```
     Configure Split and Merge Policy for a Stretched Cluster

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

                                                        [Entry Fields]
  Split Handling Policy                               Tie Breaker          +
  Merge Handling Policy                               Tie Breaker          +

  Split and Merge Action Plan                         Reboot

  Select Tie Breaker                                 (00cdb3117a5af183) +



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

We use the Swingbench load generator to simulate a database workload. We execute the PL/SQL procedure to know when the last database insert was done, when the failure instance was up, and at what time the first insert was committed using the new instance.

We start execution of the PL/SQL procedure by using the sqlplus client. We verify our SQL*Net connection string for connection to the remote listener, as shown in Example 8-134.

Example 8-134 Tnsping to itsodb

```
F:\oracle\app\M\product\11.2.0\client_1\network\admin\sqlnet.ora

Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST =
scanr6r4sat.austin.ibm.com)(PORT = 1521)) (LOAD_BALANCE=yes)(FAILOVER=ON)
(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = itsodb.austin.ibm.com)
(FAILOVER_MODE= (TYPE=select)( METHOD=basic)(RETRIES=30)(DELAY=5))))
OK (480 msec)
```

We validate the Oracle RAC resource availability, as shown in Example 8-135.

Example 8-135 RAC resource availability

```
root@r6r4m51:/>  /u01/app/11.2.0/grid/bin/crsctl stat res -t
--------------------------------------------------------------------------------
NAME           TARGET  STATE        SERVER                   STATE_DETAILS
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.DATA.dg
               ONLINE  ONLINE       r6r4m51
               ONLINE  ONLINE       r6r4m52
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
ora.LISTENER.lsnr
               ONLINE  ONLINE       r6r4m51
               ONLINE  ONLINE       r6r4m52
```

```
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
ora.asm
               ONLINE  ONLINE       r6r4m51                  Started
               ONLINE  ONLINE       r6r4m52                  Started
               ONLINE  ONLINE       satsspc2                 Started
               ONLINE  ONLINE       satsspc4                 Started
ora.gsd
               OFFLINE OFFLINE      r6r4m51
               OFFLINE OFFLINE      r6r4m52
               OFFLINE OFFLINE      satsspc2
               OFFLINE OFFLINE      satsspc4
ora.net1.network
               ONLINE  ONLINE       r6r4m51
               ONLINE  ONLINE       r6r4m52
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
ora.ons
               ONLINE  ONLINE       r6r4m51
               ONLINE  ONLINE       r6r4m52
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
ora.registry.acfs
               ONLINE  ONLINE       r6r4m51
               ONLINE  ONLINE       r6r4m52
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       r6r4m52
ora.cvu
      1        ONLINE  ONLINE       r6r4m52
ora.itsodb.db
      1        ONLINE  ONLINE       r6r4m51                  Open
      2        ONLINE  ONLINE       r6r4m52                  Open
      3        ONLINE  ONLINE       satsspc4                 Open
      4        ONLINE  ONLINE       satsspc2                 Open
ora.oc4j
      1        OFFLINE OFFLINE
ora.r6r4m51.vip
      1        ONLINE  ONLINE       r6r4m51
ora.r6r4m52.vip
      1        ONLINE  ONLINE       r6r4m52
ora.satsspc2.vip
      1        ONLINE  ONLINE       satsspc2
ora.satsspc4.vip
      1        ONLINE  ONLINE       satsspc4
ora.scan1.vip
      1        ONLINE  ONLINE       r6r4m52
```

We deactivate the zones for all four nodes of the DS5K storage, as shown in Example 8-136 on page 315.

Example 8-136 Deactivate nodes' zones and halt the nodes r6r4m51 and r6r4m52

```
hastk5-12:admin> cfgremove  "stk5_cfg",
"r6r4m51_fcs0_ds8k5;r6r4m51_fcs1_ds8k5;r6r4m52_fcs0_ds8k5;satsspc2_fcs0_ds8k5;sats
spc4_fcs0_ds8k5;r6r4m52_fcs1_ds8k5"
hastk5-12:admin> cfgenable stk5_cfg
You are about to enable a new zoning configuration.
This action will replace the old zoning configuration with the
current configuration selected.
Do you want to enable 'stk5_cfg' configuration  (yes, y, no, n): [no] y
zone config "stk5_cfg" is in effect
Updating flash ...
hastk5-12:admin>

The nodes r6r4m51 and r6r4m52 are powered off by HMC using immediate option.
```

In the `syslog.phake` file, we observe when the ORAM_MG mirror group has been fully processed and monitor the messages from the ORACLE RAC cluster reconfiguration, as shown in Example 8-137.

Example 8-137 Oracle RAC reconfiguration and ORA_MG mirror group swap

```
Feb 10 01:17:34 satsspc4 kern:debug unix:  phake_event.c:  35127383:
post_sfw_action(): Posting of Action 'PPRC_ACT_DO_NOTHING' to SFW for
event_handle='0xF100010037567F20' MG[ORA_MG 9] RDG[pha_9654761964rdg1] completed
with rc=22
Feb 10 01:17:34 satsspc4 kern:debug unix:  phake_event.c:  35127383:
process_sfw_event(): Processing of SFW Event '0x40000' for MG[ORA_MG 9] @
'0xF100010FE8F76800' completed with rc=0.

Feb 10 01:18:13 satsspc4 kern:crit unix:
Feb 10 01:18:13 satsspc4 kern:crit unix: [Oracle OKS] Node count 2, Local node
number 4
Feb 10 01:18:13 satsspc4 kern:crit unix: ADVMK-00013: Cluster reconfiguration
started.
Feb 10 01:18:13 satsspc4 kern:crit unix:
Feb 10 01:18:19 satsspc4 kern:crit unix: ADVMK-00014: Cluster reconfiguration
completed.
Feb 10 01:18:19 satsspc4 kern:crit unix:
Feb 10 01:18:19 satsspc4 kern:crit unix: ADVMK-00014: Cluster reconfiguration
completed.
Feb 10 01:18:19 satsspc4 kern:crit unix:
Feb 10 01:18:20 satsspc4 kern:crit unix: OKSK-00009: Cluster Membership change
setup complete.
Feb 10 01:18:20 satsspc4 kern:crit unix:
```

We validate the cluster status after the site failure, as shown in Example 8-138 on page 316.

Example 8-138 Resource status

```
root@satsspc4:/> /u01/app/11.2.0/grid/bin/crsctl stat res -t
--------------------------------------------------------------------------------
NAME           TARGET  STATE        SERVER                   STATE_DETAILS
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.DATA.dg
               ONLINE  ONLINE       satsspc2
               ONLINE  INTERMEDIATE satsspc4
ora.LISTENER.lsnr
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
ora.asm
               ONLINE  ONLINE       satsspc2                 Started
               ONLINE  ONLINE       satsspc4                 Started
ora.gsd
               OFFLINE OFFLINE      satsspc2
               OFFLINE OFFLINE      satsspc4
ora.net1.network
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
ora.ons
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
ora.registry.acfs
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       satsspc4
ora.cvu
      1        ONLINE  ONLINE       satsspc2
ora.itsodb.db
      1        ONLINE  OFFLINE
      2        ONLINE  OFFLINE
      3        ONLINE  ONLINE       satsspc4                 Open
      4        ONLINE  ONLINE       satsspc2                 Open
ora.oc4j
      1        OFFLINE OFFLINE
ora.r6r4m51.vip
      1        ONLINE  INTERMEDIATE satsspc2                 FAILED OVER
ora.r6r4m52.vip
      1        ONLINE  INTERMEDIATE satsspc2                 FAILED OVER
ora.satsspc2.vip
      1        ONLINE  ONLINE       satsspc2
ora.satsspc4.vip
      1        ONLINE  ONLINE       satsspc4
ora.scan1.vip
      1        ONLINE  ONLINE       satsspc4
root@satsspc4:/>
```

Then, we verify the insert sequence, as shown in Example 8-139.

Example 8-139 Insert into database after swap

```
SQL> select min(data) data_start,max(data) data_end,instance from performance
group by instance;
DATA_START       DATA_END         instance
---------------------------------------------------------------------------
10-FEB-14 01.16.42.908554 AM10-FEB-14 01.17.02.307496 AM  itsodb1
10-FEB-14 01.18.35.887111 AM10-FEB-14 01.19.57.605028 AM  itsodb3
```

8.20.8 CAA dynamic disk addition in a HyperSwap environment

One of the newest features of the Enterprise Edition of PowerHA SystemMirror 7.1.3 is to migrate and dynamically configure a CAA repository disk as a HyperSwap protected disk.

This operation requires the following steps:

1. Configure a HyperSwap-enabled disk on all cluster nodes.
2. Create a new Cluster_Repository mirror group with the corresponding HyperSwap disk as being used for the CAA repository disk. The actual CAA repository is indicated as not a HyperSwap disk.
3. Verify and synchronize.
4. Validate the new Cluster Repository mirror group in the clxd.log.
5. Validate the CAA cluster configuration with the new HyperSwap disk.

First, we verify the existing repository disk configuration, as shown in Example 8-140.

Example 8-140 Repository disk configuration

```
root@r6r4m51:/> odmget HACMPsircol

HACMPsircol:
        name = "orahyp1_sircol"
        id = 0
        uuid = "0"
        ip_address = ""
        repository = "00cdb31104eb34c3"
        backup_repository = ""
root@r6r4m51:/> lscluster -d
Storage Interface Query

Cluster Name: orahyp1
Cluster UUID: c5c8b7ca-8eda-11e3-9fc8-001a64b94abd
Number of nodes reporting = 4
Number of nodes expected = 4

Node r6r4m51.austin.ibm.com
Node UUID = c5b720be-8eda-11e3-9fc8-001a64b94abd
Number of disks discovered = 1
        hdisk22:
              State : UP
               uDid : 200B75TL771520507210790003IBMfcp
               uUid : 872ba55b-b512-a9b4-158b-043f8bc50000
          Site uUid : 51735173-5173-5173-5173-517351735173
```

```
              Type : REPDISK

Node satsspc2.austin.ibm.com
Node UUID = c5b723f2-8eda-11e3-9fc8-001a64b94abd
Number of disks discovered = 1
        hdisk57:
             State : UP
              uDid : 200B75TL771520507210790003IBMfcp
              uUid : 872ba55b-b512-a9b4-158b-043f8bc50000
          Site uUid : 51735173-5173-5173-5173-517351735173
              Type : REPDISK

Node r6r4m52.austin.ibm.com
Node UUID = c5b72334-8eda-11e3-9fc8-001a64b94abd
Number of disks discovered = 1
        hdisk52:
             State : UP
              uDid : 200B75TL771520507210790003IBMfcp
              uUid : 872ba55b-b512-a9b4-158b-043f8bc50000
          Site uUid : 51735173-5173-5173-5173-517351735173
              Type : REPDISK

Node satsspc4.austin.ibm.com
Node UUID = c5b7249c-8eda-11e3-9fc8-001a64b94abd
Number of disks discovered = 1
        hdisk54:
             State : UP
              uDid : 200B75TL771520507210790003IBMfcp
              uUid : 872ba55b-b512-a9b4-158b-043f8bc50000
          Site uUid : 51735173-5173-5173-5173-517351735173
              Type : REPDISK

root@r6r4m51:/> lspv -u |grep hdisk22
hdisk22         00cdb31104eb34c3                    caavg_private   active
200B75TL771520507210790003IBMfcp
872ba55b-b512-a9b4-158b-043f8bc50000
root@r6r4m51:/>
```

We check hdisk100 as a HyperSwap-configured disk on host r6r4m51 and on the other hosts, as shown in Example 8-141.

Example 8-141 Checking the HyperSwap-configured disk 100 on host r6r4m51

```
root@r6r4m51:/> lspprc -v hdisk100

HyperSwap lun unique
identifier.............352037354e52353731433930310052f416e907210790003IBMfcp

hdisk100        Primary         MPIO IBM 2107 FC Disk

        Manufacturer..................IBM
        Machine Type and Model........2107900
        ROS Level and ID..............2E393330
        Serial Number.................75NR571C
        Device Specific.(Z7)..........C901
        Device Specific.(Z0)..........000005329F101002
```

```
        Device Specific.(Z1)..........901
        Device Specific.(Z2)..........075
        Unique Device Identifier......200B75NR571C90107210790003IBMfcp
        Logical Subsystem ID..........0xc9
        Volume Identifier.............0x01
        Subsystem Identifier(SS ID)...0xFFC9
        Control Unit Sequence Number..00000NR571
        Storage Subsystem WWNN........5005076309ffc5d5
        Logical Unit Number ID........40c9400100000000

hdisk100        Secondary       MPIO IBM 2107 FC Disk

        Manufacturer..................IBM
        Machine Type and Model........2107900
        ROS Level and ID..............2E313336
        Serial Number.................75LY981E
        Device Specific.(Z7)..........EA01
        Device Specific.(Z0)..........000005329F101002
        Device Specific.(Z1)..........A01
        Device Specific.(Z2)..........075
        Unique Device Identifier......200B75LY981EA0107210790003IBMfcp
        Logical Subsystem ID..........0xea
        Volume Identifier.............0x01
        Subsystem Identifier(SS ID)...0xFFEA
        Control Unit Sequence Number..00000LY981
        Storage Subsystem WWNN........5005076308ffc6d4
        Logical Unit Number ID........40ea400100000000

root@r6r4m51:/>  dsh /work/lshostvol.sh |egrep 'C901|EA01'
r6r4m51.austin.ibm.com:  hdisk100         IBM.2107-75NR571/C901
r6r4m52.austin.ibm.com:  hdisk94          IBM.2107-75NR571/C901
satsspc2.austin.ibm.com:  hdisk97         IBM.2107-75NR571/C901
satsspc4.austin.ibm.com:  hdisk99         IBM.2107-75NR571/C901
```

We also validate the UUID for the new CAA hdisk, as shown in Example 8-142.

Example 8-142 Validating the UUID for the new CAA disk

```
root@r6r4m51:/> -a561-ae19-311fca3ed3f7|dshbak -c                          <
HOSTS -------------------------------------------------------------------------
r6r4m51.austin.ibm.com
-------------------------------------------------------------------------------
hdisk100   00cdb3110988789d        caavg_private active
352037354e52353731433930310052f416e907210790003IBMfcp   af87d5be-0ac
c-a561-ae19-311fca3ed3f7

HOSTS -------------------------------------------------------------------------
r6r4m52.austin.ibm.com
-------------------------------------------------------------------------------
hdisk94    00cdb3110988789d        caavg_private active
352037354e52353731433930310052f416e907210790003IBMfcp   af87d5be-0ac
c-a561-ae19-311fca3ed3f7

HOSTS -------------------------------------------------------------------------
satsspc2.austin.ibm.com
-------------------------------------------------------------------------------
```

```
hdisk97    00cdb3110988789d        caavg_private active
352037354e52353731433930310052f416e907210790003IBMfcp    af87d5be-0ac
c-a561-ae19-311fca3ed3f7

HOSTS -------------------------------------------------------------------------
satsspc4.austin.ibm.com
-------------------------------------------------------------------------------
hdisk99    00cdb3110988789d        caavg_private active
352037354e52353731433930310052f416e907210790003IBMfcp    af87d5be-0ac
c-a561-ae19-311fca3ed3f7
```

After the verification, we add a new Cluster_Repository mirror group by accessing the fast path, as shown in Example 8-143:

smitty cm_add_mirr_gps_select

Example 8-143 Adding a Cluster_Repository mirror group

```
                       Add cluster Repository Mirror Group

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

                                                         [Entry Fields]
  Mirror Group Name                                    CAA_MG
  New Mirror Group Name                               []
* Site Name                                            SITE_A SITE_B         +
  Non HyperSwap Disk                                  [hdisk22:872ba55b-b512> +
* HyperSwap Disk                                      [hdisk100:af87d5be-0cc> +
  Associated Storage System(s)                         STG_A STG_B           +
  HyperSwap                                            Enabled               +
  Consistency Group                                    yes
  Unplanned HyperSwap Timeout (in sec)                [60]                    #
  HyperSwap Priority                                   High
  Re-sync Action                                       Manual                +



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

The only step left to activate the new CAA HyperSwap disk is to verify and synchronize the cluster. During this step, the disk repository is changed to a HyperSwap-enabled disk, as shown in Example 8-144. The operation logs are in the `clxd.log`.

Example 8-144 HyperSwap repository disk, new configuration

```
root@r6r4m51:/> lscluster -d
Storage Interface Query

Cluster Name: orahyp1
Cluster UUID: c5c8b7ca-8eda-11e3-9fc8-001a64b94abd
Number of nodes reporting = 4
Number of nodes expected = 4
```

```
Node r6r4m51.austin.ibm.com
Node UUID = c5b720be-8eda-11e3-9fc8-001a64b94abd
Number of disks discovered = 1
        hdisk100:
              State : UP
               uDid : 352037354e52353731433930310052f416e907210790003IBMfcp
               uUid : af87d5be-0acc-a561-ae19-311fca3ed3f7
           Site uUid : 51735173-5173-5173-5173-517351735173
               Type : REPDISK

Node satsspc4.austin.ibm.com
Node UUID = c5b7249c-8eda-11e3-9fc8-001a64b94abd
Number of disks discovered = 1
        hdisk99:
              State : UP
               uDid : 352037354e52353731433930310052f416e907210790003IBMfcp
               uUid : af87d5be-0acc-a561-ae19-311fca3ed3f7
           Site uUid : 51735173-5173-5173-5173-517351735173
               Type : REPDISK

Node satsspc2.austin.ibm.com
Node UUID = c5b723f2-8eda-11e3-9fc8-001a64b94abd
Number of disks discovered = 1
        hdisk97:
              State : UP
               uDid : 352037354e52353731433930310052f416e907210790003IBMfcp
               uUid : af87d5be-0acc-a561-ae19-311fca3ed3f7
           Site uUid : 51735173-5173-5173-5173-517351735173
               Type : REPDISK

Node r6r4m52.austin.ibm.com
Node UUID = c5b72334-8eda-11e3-9fc8-001a64b94abd
Number of disks discovered = 1
        hdisk94:
              State : UP
               uDid : 352037354e52353731433930310052f416e907210790003IBMfcp
               uUid : af87d5be-0acc-a561-ae19-311fca3ed3f7
           Site uUid : 51735173-5173-5173-5173-517351735173
               Type : REPDISK
```

You can easily revert to a non-HyperSwap disk by using the standard procedure for CAA repository disk replacement:

1. Add a new repository disk (use either the **smitty cm_add_repository_disk** or the **clmgr add repository <disk>** command). The disk should meet CAA repository disk requirements.
2. Replace the repository disk (**smitty cl_replace_repository_nm** or **clmgr replace repository <new_repository>**). For more **clmgr** options, use the **clmgr** contextual help.

8.21 Online storage migration: Oracle RAC in a HyperSwap configuration

The HyperSwap feature offers online storage migration. The storage migration is performed by following the same steps as for the single-node HyperSwap storage migration, but the operations should be performed on all Oracle RAC nodes.

These are the storage migration steps:

1. Validate the source disk location to be on the storage that will be removed. If this condition is not satisfied, a planned swap is required to bring all disks onto the storage that will be removed.
2. PowerHA services must be stopped with Unmanaged groups options (on all of the nodes where resource groups are online).
3. Use the **chdev -l hdisk# -a san_rep_cfg=revert_disk -U** command (for all the disks part of MG) on all Oracle RAC nodes.
4. Use **rmdev -dl hdisk#** (for all LUNs from the auxiliary storage) on all RAC nodes.
5. Use **rmpprc** for the HyperSwap disks to the existing auxiliary storage.
6. (Optional) Use **rmpprcpath** to delete remote mirroring and copy paths to existing auxiliary storage.
7. Remove disks from the volume group configuration so they are not available on the host.
8. Create the PPRC paths by using the **mkpprcpath** command with the new auxiliary storage.
9. Use **mkpprc** for existing disks to create a remote mirror and copy the relationship to the new auxiliary storage.
10. Configure zones for every host with the new auxiliary storage.
11. Configure the hostconnect HyperSwap profile for every host that is attached to the new auxiliary storage.
12. Use **cfgmgr**, **chdev** to `no_reserve` and **chdev -l hdisk# -a san_rep_cfg=migrate_disk -U** (for all of the disks from new auxiliary storage),
13. Create the new storage subsystem.
14. Start PowerHA services on all nodes in Unmanaged mode for the resource groups.
15. Change the mirror group and add raw disks and volume groups again.
16. Verify and synchronize.
17. Bring resource groups online.
18. Inspect `clxd.log` for errors.

8.21.1 Online storage migration for Oracle RAC in a HyperSwap configuration

For this example, we configured an Oracle Real Application Cluster with four nodes: Two nodes in Site A and two nodes in Site B. In each site, there is one DS8800 storage repository, as shown in Figure 8-14. Metro Mirror data replication is configured for the ASM disks and the CAA repository disk. The Oracle application binaries are installed locally.

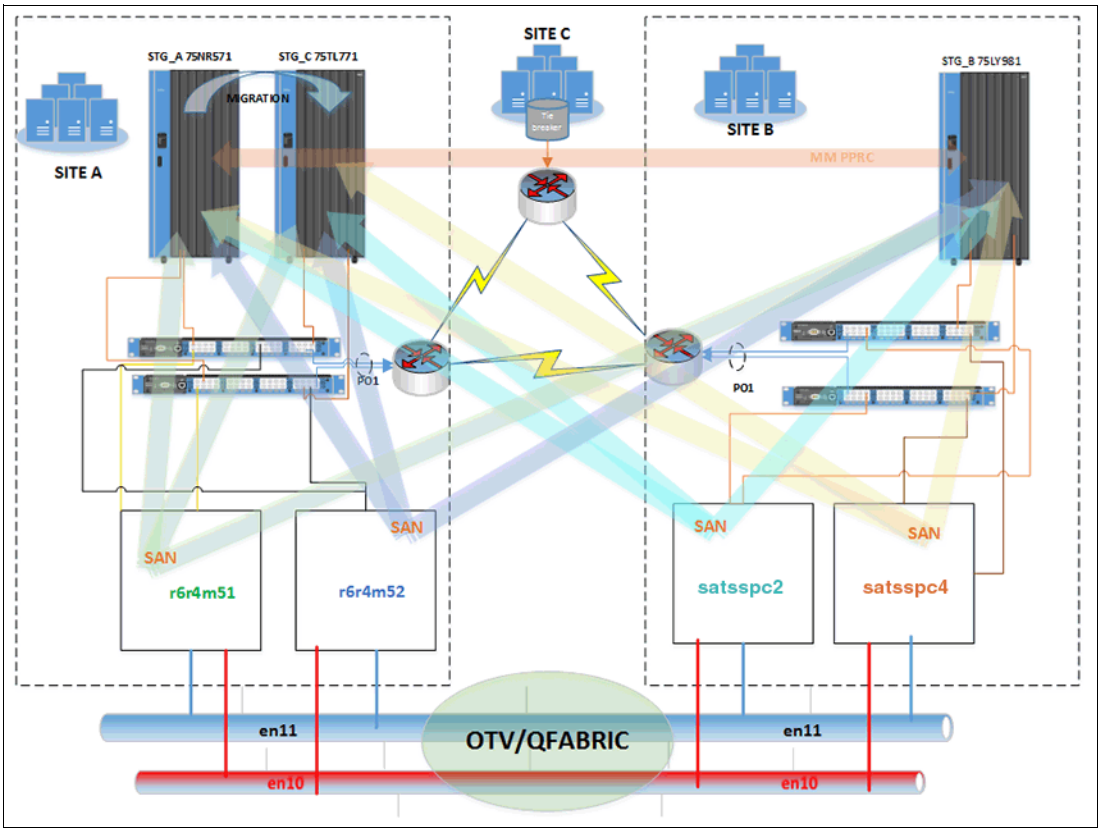


Figure 8-14 Oracle RAC, 4 nodes 2 sites

The disks used for ASM configuration and their storage membership are shown in Table 8-7.

Table 8-7 ASM configuration and storage membership

Host	Storage	asm_disk1	asm_disk2	asm_disk3	asm_disk4	asm_disk5	caa disk
R6R4M51	STG A	C304/hdisk49	C305/hdisk50	C404/hdisk97	C501/hdisk99	C502/hdisk100	C901/hdisk101
	STG B	7F01/hdisk44	7F02/hdisk45	9F01/hdisk46	E799/hdisk73	E798/hdisk77	EA01/hdisk78
	STG C	A204/hdisk102	A205/hdisk103	2F02/hdisk105	3700/hdisk106	3701/hdisk107	2E01/hdisk104
R6R4M52	STG A	C304/hdisk59	C305/hdisk98	C404/hdisk99	C501/hdisk101	C502/hdisk102	C901/hdisk100
	STG B	7F01/hdisk48	7F02/hdisk49	9F01/hdisk61	E799/hdisk62	E798/hdisk63	EA01/hdisk97
	STG C	A204/hdisk103	A205/hdisk104	2F02/hdisk108	3700/hdisk106	3701/hdisk107	2E01/hdisk105
SATSSPC2	STG A	C304/hdisk101	C305/hdisk102	C404/hdisk103	C501/hdisk105	C502/hdisk106	C901/hdisk101
	STG B	7F01/hdisk63	7F02/hdisk64	9F01/hdisk67	E799/hdisk68	E798/hdisk69	EA01/hdisk98
	STG C	A204/hdisk97	A205/hdisk94	2F02/hdisk95	3700/hdisk96	3701/hdisk99	2E01/hdisk43

Host	Storage	asm_disk1	asm_disk2	asm_disk3	asm_disk4	asm_disk5	caa disk
SATSSPC4	STG A	**C3**04/hdisk98	**C3**05/hdisk99	**C4**04/hdisk101	**C5**01/hdisk103	**C5**02/hdisk104	**C9**01/hdisk93
	STG B	**7F**01/hdisk71	**7F**02/hdisk72	**9F**01/hdisk75	**E7**99/hdisk93	**E7**98/hdisk92	**EA**01/hdisk86
	STG C	**A2**04/hdisk87	**A2**05/hdisk88	**2F**02/hdisk97	**37**00/hdisk95	**37**01/hdisk96	**2E**01/hdisk94

The hdisks marked in blue in the preceding table remain in their positions during migration. The LSS membership of each volume is also indicated in blue.

We follow the configuration steps in 8.21, "Online storage migration: Oracle RAC in a HyperSwap configuration" on page 322, using the Swingbench to load the database that we configured in the Oracle RAC environment.

We also use the Enterprise Manager Console to observe how all configurations are doing their various steps for storage migration as reflected in our test environment.

We start by verifying the Oracle RAC resources status as shown in Example 8-145.

Example 8-145 Oracle RAC resource status

```
root@r6r4m51:/work>  /u01/app/11.2.0/grid/bin/crsctl stat res -t
--------------------------------------------------------------------------------
NAME           TARGET  STATE        SERVER                   STATE_DETAILS
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.DATA.dg
               ONLINE  ONLINE       r6r4m51
               ONLINE  ONLINE       r6r4m52
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
ora.LISTENER.lsnr
               ONLINE  ONLINE       r6r4m51
               ONLINE  ONLINE       r6r4m52
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
ora.asm
               ONLINE  ONLINE       r6r4m51                  Started
               ONLINE  ONLINE       r6r4m52                  Started
               ONLINE  ONLINE       satsspc2                 Started
               ONLINE  ONLINE       satsspc4                 Started
ora.gsd
               OFFLINE OFFLINE      r6r4m51
               OFFLINE OFFLINE      r6r4m52
               OFFLINE OFFLINE      satsspc2
               OFFLINE OFFLINE      satsspc4
ora.net1.network
               ONLINE  ONLINE       r6r4m51
               ONLINE  ONLINE       r6r4m52
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
ora.ons
               ONLINE  ONLINE       r6r4m51
               ONLINE  ONLINE       r6r4m52
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
ora.registry.acfs
               ONLINE  ONLINE       r6r4m51
```

```
               ONLINE  ONLINE       r6r4m52
               ONLINE  ONLINE       satsspc2
               ONLINE  ONLINE       satsspc4
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       r6r4m52
ora.cvu
      1        ONLINE  ONLINE       r6r4m52
ora.itsodb.db
      1        ONLINE  ONLINE       r6r4m51                  Open
      2        ONLINE  ONLINE       r6r4m52                  Open
      3        ONLINE  ONLINE       satsspc4                 Open
      4        ONLINE  ONLINE       satsspc2                 Open
ora.oc4j
      1        OFFLINE OFFLINE
ora.r6r4m51.vip
      1        ONLINE  ONLINE       r6r4m51
ora.r6r4m52.vip
      1        ONLINE  ONLINE       r6r4m52
ora.satsspc2.vip
      1        ONLINE  ONLINE       satsspc2
ora.satsspc4.vip
      1        ONLINE  ONLINE       satsspc4
ora.scan1.vip
      1        ONLINE  ONLINE       r6r4m52
```

We also start the Swingbench test with the configuration, as shown in Figure 8-15.

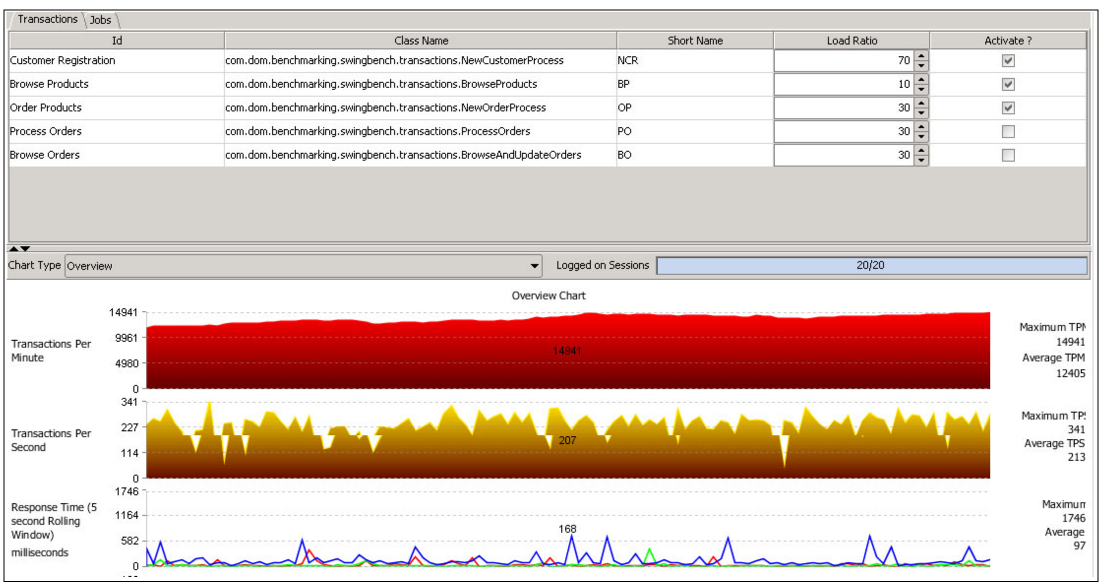


Figure 8-15 Swingbench load

We validate the disk PPRC states, and the path groups IDs as shown in Example 8-146.

Example 8-146 Validating the PPRC states and the path groups ID

```
root@r6r4m51:/work> dsh /work/"asm_disks_n.sh" |dshbak -c
HOSTS -------------------------------------------------------------------------
r6r4m51.austin.ibm.com
```

```
--------------------------------------------------------------------------------
hdisk49   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk50   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk97   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk99   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk100  Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4

HOSTS --------------------------------------------------------------------------
r6r4m52.austin.ibm.com
--------------------------------------------------------------------------------
hdisk59   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk98   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk99   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk101  Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk102  Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4

HOSTS --------------------------------------------------------------------------
satsspc4.austin.ibm.com
--------------------------------------------------------------------------------
hdisk98   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk99   Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk101  Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk103  Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk104  Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4

HOSTS --------------------------------------------------------------------------
satsspc2.austin.ibm.com
--------------------------------------------------------------------------------
hdisk101  Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk102  Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk103  Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk105  Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
hdisk106  Active   0(s)        1          5005076309ffc5d5  5005076308ffc6d4
```

With the disks with the source in the Storage A, we swap the disks to Storage B. We validate the operation with the `clxd.log` and again issue the command for path and stat validation. The swap operation log is shown in Example 8-147. It marks the start time for that migration operation.

Example 8-147 The swap operation log

```
INFO      |2014-02-19T03:22:17.171240|Raw Disks =
fa4ac646-ef1c-e519-0b65-68fc36ed33dc
INFO      |2014-02-19T03:22:17.171265|Raw Disks =
865b3ec8-a5bf-3e6d-2398-44c3d8ced587
INFO      |2014-02-19T03:22:17.171405|Raw Disks =
b12abf86-5759-8b5e-c3ed-c85a38c82949
INFO      |2014-02-19T03:22:17.171454|Raw Disks =
0da61546-184d-5528-6fbb-cb1c2e9ccd83
INFO      |2014-02-19T03:22:17.171479|Raw Disks =
e9fdd63b-7e90-901a-0350-5e57f8e5dbff
INFO      |2014-02-19T03:22:36.157935|Received XD CLI request = '' (0x1d)
INFO      |2014-02-19T03:22:37.158109|Received XD CLI request = 'Swap Mirror
Group' (0x1c)
```

```
INFO      |2014-02-19T03:22:37.158148|Request to Swap Mirror Group 'ORA_MG',
Direction 'SITE_B', Outfile ''
INFO      |2014-02-19T03:22:37.170244|No VG found for MG=ORA_MG
INFO      |2014-02-19T03:22:37.170268|No of VG found for MG ORA_MG
INFO      |2014-02-19T03:22:37.170290|Not able to find any VG disks for MG=ORA_MG
INFO      |2014-02-19T03:22:37.345501|Calling sfwGetRepGroupInfo()
INFO      |2014-02-19T03:22:37.345565|sfwGetRepGroupInfo() completed
INFO      |2014-02-19T03:22:37.345600|Calling sfwGetRepGroupInfo()
INFO      |2014-02-19T03:22:37.345638|sfwGetRepGroupInfo() completed
INFO      |2014-02-19T03:22:37.345662|Calling sfwGetRepGroupInfo()
INFO      |2014-02-19T03:22:37.345700|sfwGetRepGroupInfo() completed
INFO      |2014-02-19T03:22:37.402053|Calling DO_SWAP
INFO      |2014-02-19T03:22:38.605639|DO_SWAP completed
INFO      |2014-02-19T03:22:38.605932|Swap Mirror Group 'ORA_MG' completed.

root@r6r4m51:/work> dsh /work/"asm_disks_n.sh" |dshbak -c
HOSTS -------------------------------------------------------------------------
r6r4m52.austin.ibm.com
-------------------------------------------------------------------------------
hdisk59   Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk98   Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk99   Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk101  Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk102  Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5

HOSTS -------------------------------------------------------------------------
r6r4m51.austin.ibm.com
-------------------------------------------------------------------------------
hdisk49   Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk50   Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk97   Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk99   Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk100  Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5

HOSTS -------------------------------------------------------------------------
satsspc4.austin.ibm.com
-------------------------------------------------------------------------------
hdisk98   Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk99   Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk101  Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk103  Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk104  Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5

HOSTS -------------------------------------------------------------------------
satsspc2.austin.ibm.com
-------------------------------------------------------------------------------
hdisk101  Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk102  Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk103  Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk105  Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
hdisk106  Active   1(s)        0           5005076308ffc6d4  5005076309ffc5d5
```

Now, we proceed to stop the HACMP services by bringing the resource group to an Unmanaged state, as shown in Example 8-148 on page 328.

Example 8-148   Stopping HACMP services

```
INFO      |2014-02-19T03:25:49.108105|Calling STOP_MG
INFO      |2014-02-19T03:25:49.108230|STOP_MG completed
INFO      |2014-02-19T03:25:49.108383|Stop Mirror Group 'ORA_MG' completed.
root@r6r4m51:/> clRGinfo -p -v

Cluster Name: orahyp1

Resource Group Name: ORARG
Startup Policy: Online On All Available Nodes
Fallover Policy: Bring Offline (On Error Node Only)
Fallback Policy: Never Fallback
Site Policy: Online On Both Sites
Node                         Primary State    Secondary State
---------------------------- ---------------
r6r4m51@SITE_A               UNMANAGED        OFFLINE
r6r4m52@SITE_A               UNMANAGED        OFFLINE
satsspc4@SITE_B              UNMANAGED        OFFLINE
satsspc2@SITE_B              UNMANAGED        OFFLINE
```

We must maintain the hdisk number for all HyperSwap disks, even if we remove the disks from Storage A from the configuration. We use the **chdev** command to update the disk attributes to `revert_disk` with `-U` attribute, as shown in Example 8-149. In this way, the hdisk number is associated with the disk from the secondary storage (Storage B in this example). We change the disk attributes for all HyperSwap disks that are part of the storage migration. If thare non-HyperSwap related disks that also need to be migrated, they must be accounted for at the beginning.

Example 8-149   Updating disk attributes by using revert_disk

```
root@r6r4m51:/> for i in  hdisk49 hdisk50 hdisk97 hdisk99 hdisk100; do chdev -l $i -a
san_rep_cfg=revert_disk -U;done
hdisk49 changed
hdisk50 changed
hdisk97 changed
hdisk99 changed
hdisk100 changed

root@r6r4m52:/> for i in hdisk59 hdisk98 hdisk99 hdisk101 hdisk102 ;do chdev -l $i -a
san_rep_cfg=revert_disk -U;done
hdisk59 changed
hdisk98 changed
hdisk99 changed
hdisk101 changed
hdisk102 changed

root@satsspc2:/> for i in hdisk101 hdisk102 hdisk103 hdisk105 hdisk106; do chdev -l $i -a
san_rep_cfg=revert_disk -U;done
hdisk101 changed
hdisk102 changed
hdisk103 changed
hdisk105 changed
hdisk106 changed

root@satsspc4:/> for i in hdisk98 hdisk99 hdisk101 hdisk103 hdisk104 ; do chdev -l $i -a
san_rep_cfg=revert_disk -U;done
hdisk98 changed
```

```
hdisk99 changed
hdisk101 changed
hdisk103 changed
hdisk104 changed
```

After this operation, the disks are seen on the system as source on Storage B without a path or configured disk in Storage A, as shown in Example 8-150. A path group ID of -1 indicates that there are no paths configured from this initiator to the indicated LUN in the PPRC pair.

Example 8-150 Disk paths after the revert_disk attribute is applied

```
root@r6r4m51:/> dsh /work/"asm_disks_n.sh" |dshbak -c
HOSTS -------------------------------------------------------------------------
r6r4m52.austin.ibm.com
-------------------------------------------------------------------------------
hdisk59   Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
hdisk98   Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
hdisk99   Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
hdisk101  Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
hdisk102  Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5

HOSTS -------------------------------------------------------------------------
r6r4m51.austin.ibm.com
-------------------------------------------------------------------------------
hdisk49   Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
hdisk50   Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
hdisk97   Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
hdisk99   Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
hdisk100  Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5

HOSTS -------------------------------------------------------------------------
satsspc4.austin.ibm.com
-------------------------------------------------------------------------------
hdisk98   Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
hdisk99   Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
hdisk101  Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
hdisk103  Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
hdisk104  Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5

HOSTS -------------------------------------------------------------------------
satsspc2.austin.ibm.com
-------------------------------------------------------------------------------
hdisk101  Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
hdisk102  Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
hdisk103  Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
hdisk105  Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
hdisk106  Active   1(s)        -1          5005076308ffc6d4  5005076309ffc5d5
```

The next step is to remove the PPRC relationships, as shown in Example 8-151.

Example 8-151 Removing PPRC relationships

```
dscli> rmpprc -remotedev IBM.2107-75NR571 -quiet 7f01:C304 7F02:C305 9F01:C404 E798:C502 E799:C501
EA01:C901
Date/Time: February 19, 2014 3:35:54 AM CST IBM DSCLI Version: 6.6.0.305 DS: IBM.2107-75LY981
CMUC00155I rmpprc: Remote Mirror and Copy volume pair 7F01:C304 relationship successfully withdrawn.
CMUC00155I rmpprc: Remote Mirror and Copy volume pair 7F02:C305 relationship successfully withdrawn.
```

```
CMUC00155I rmpprc: Remote Mirror and Copy volume pair 9F01:C404 relationship successfully withdrawn.
CMUC00155I rmpprc: Remote Mirror and Copy volume pair E798:C502 relationship successfully withdrawn.
CMUC00155I rmpprc: Remote Mirror and Copy volume pair E799:C501 relationship successfully withdrawn.
CMUC00155I rmpprc: Remote Mirror and Copy volume pair EA01:C901 relationship successfully withdrawn.
```

We create the PPRC relationships for all volume pairs, now with the new storage, as shown in Example 8-152.

Example 8-152 Establishing PPRC for volume pairs with the new storage

```
mkpprc -remotedev IBM.2107-75TL771 -type mmir 7f01:a204 7f02:a205 9f01:2f02 e799:3700 e798:3701
dscli> mkpprc -remotedev IBM.2107-75TL771 -type mmir 7f01:a204 7f02:a205 9f01:2f02 e799:3700 e798:3701
ea01:2e01
Date/Time: February 19, 2014 3:37:09 AM CST IBM DSCLI Version: 6.6.0.305 DS: IBM.2107-75LY981
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 7F01:A204 successfully created.
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 7F02:A205 successfully created.
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship 9F01:2F02 successfully created.
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship E799:3700 successfully created.
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship E798:3701 successfully created.
CMUC00153I mkpprc: Remote Mirror and Copy volume pair relationship EA01:2E01 successfully created.
```

Before configuring disks to be HyperSwap-capable, we must wait while the disks are copied to the new storage system. You can monitor the process by using the **lspprc** command at the storage level, as shown in Example 8-153.

Example 8-153 Monitoring the disk copying process

```
dscli> lspprc -l 7f01 7f02 9f01 e799 e798 ea01
Date/Time: February 19, 2014 3:37:38 AM CST IBM DSCLI Version: 6.6.0.305 DS: IBM.2107-75LY981
ID        State        Reason Type         Out Of Sync Tracks Tgt Read Src Cascade Tgt Cascade Date Suspended
SourceLSS Timeout (secs) Critical Mode First Pass Status Incremental Resync Tgt Write GMIR CG PPRC CG isTgtSE
DisableAutoResync
===================================================================================================================
===================================================================================================================
===
7F01:A204 Copy Pending -      Metro Mirror 58677            Disabled Disabled    Invalid     -             7F
60            Disabled      Invalid          Disabled          Disabled  N/A    Enabled Unknown -
7F02:A205 Copy Pending -      Metro Mirror 61433            Disabled Disabled    Invalid     -             7F
60            Disabled      Invalid          Disabled          Disabled  N/A    Enabled Unknown -
9F01:2F02 Copy Pending -      Metro Mirror 39901            Disabled Disabled    Invalid     -             9F
60            Disabled      Invalid          Disabled          Disabled  N/A    Enabled Unknown -
E798:3701 Copy Pending -      Metro Mirror 68698            Disabled Disabled    Invalid     -             E7
60            Disabled      Invalid          Disabled          Disabled  N/A    Enabled Unknown -
E799:3700 Copy Pending -      Metro Mirror 2326456          Disabled Disabled    Invalid     -             E7
60            Disabled      Invalid          Disabled          Disabled  N/A    Enabled Unknown -
EA01:2E01 Copy Pending -      Metro Mirror 12677            Disabled Disabled    Invalid     -             EA
2E            Disabled      Invalid          Disabled
```

When the **lspprc** command indicates that the disks are in Full Duplex state, we proceed with the next configuration steps and run **cfgmgr** on all nodes. We verify the copy status, as shown in Example 8-154.

Example 8-154 Verify copying data

```
dscli> lspprc -l 7f01 7f02 9f01 e799 e798
Date/Time: February 19, 2014 3:55:28 AM CST IBM DSCLI Version: 6.6.0.305 DS: IBM.2107-75LY981
ID        State        Reason Type         Out Of Sync Tracks Tgt Read Src Cascade Tgt Cascade Date Suspended
SourceLSS Timeout (secs) Critical Mode First Pass Status Incremental Resync Tgt Write GMIR CG PPRC CG isTgtSE
DisableAutoResync
===================================================================================================================
===================================================================================================================
===
7F01:A204 Full Duplex  -      Metro Mirror 0                Disabled Disabled    Invalid     -             7F
60            Disabled      Invalid          Disabled          Disabled  N/A    Enabled Unknown -
7F02:A205 Full Duplex  -      Metro Mirror 0                Disabled Disabled    Invalid     -             7F
60            Disabled      Invalid          Disabled          Disabled  N/A    Enabled Unknown -
```

```
9F01:2F02 Full Duplex  -      Metro Mirror 0                  Disabled Disabled    Invalid     -             9F
60            Disabled      Invalid          Disabled          Disabled  N/A    Enabled Unknown -
E798:3701 Full Duplex  -      Metro Mirror 0                  Disabled Disabled    Invalid     -             E7
60            Disabled      Invalid          Disabled          Disabled  N/A    Enabled Unknown -
E799:3700 Copy Pending -      Metro Mirror 956                Disabled Disabled    Invalid     -             E7
60            Disabled      Invalid          Disabled          Disabled  N/A    Enabled Unknown -
EA01:2E01 Full Duplex  -      Metro Mirror 0                  Disabled Disabled    Invalid     -             EA
60            Disabled      Invalid          Disabled          Disabled  N/A    Enabled Unknown -
```

We validate the new disk attributes with the desired ones (`reserve_policy, rw_timeout`).

We start the disk configurations on all nodes by updating the san_rep_cfg disk attributes, as shown in Example 8-155.

Example 8-155 Changing disk attributes for a single node

```
root@r6r4m51:/work> for i in  hdisk49 hdisk50 hdisk97 hdisk99 hdisk100; do chdev
-l $i -a san_rep_cfg=migrate_disk -U;done
hdisk49 changed
hdisk50 changed
hdisk97 changed
hdisk99 changed
hdisk100 changed
```

The disk configuration after updating the disk attributes is shown in Example 8-156.

Example 8-156 HyperSwap disk configuration

```
root@r6r4m51:/work> lspprc -v hdisk49

HyperSwap lun unique
identifier.............352037354c5939383137463031005304 83a707210790003IBMfcp

hdisk49 Secondary       MPIO IBM 2107 FC Disk

        Manufacturer..................IBM
        Machine Type and Model........2107900
        ROS Level and ID..............2E393330
        Serial Number.................75TL771A
        Device Specific.(Z7)..........A204
        Device Specific.(Z0)..........000005329F101002
        Device Specific.(Z1)..........204
        Device Specific.(Z2)..........075
        Unique Device Identifier......200B75TL771A20407210790003IBMfcp
        Logical Subsystem ID..........0xa2
        Volume Identifier.............0x04
        Subsystem Identifier(SS ID)...0xFFA2
        Control Unit Sequence Number..00000TL771
        Storage Subsystem WWNN........500507630affc16b
        Logical Unit Number ID........40a2400400000000

hdisk49 Primary         MPIO IBM 2107 FC Disk

        Manufacturer..................IBM
        Machine Type and Model........2107900
        ROS Level and ID..............2E313336
        Serial Number.................75LY9817
        Device Specific.(Z7)..........7F01
        Device Specific.(Z0)..........000005329F101002
```

```
        Device Specific.(Z1)..........F01
        Device Specific.(Z2)..........075
        Unique Device Identifier......200B75LY9817F0107210790003IBMfcp
        Logical Subsystem ID..........0x7f
        Volume Identifier.............0x01
        Subsystem Identifier(SS ID)...0xFF7F
        Control Unit Sequence Number..00000LY981
        Storage Subsystem WWNN........5005076308ffc6d4
        Logical Unit Number ID........407f400100000000
```

We validate the disk configurations and add the new storage definition in PowerHA SystemMirror configuration, as shown in Example 8-157.

Example 8-157 Adding new storage

```
                            Add a Storage System

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

[Entry Fields]
* Storage System Name                                    [STG_C]
* Site Association                                        SITE_A +
* Vendor Specific Identifier                              IBM.2107-00000TL771 +
* WWNN                                                    500507630AFFC16B +
```

We start the cluster services without bringing up the resource groups, as shown in Example 8-158.

Example 8-158 Starting PowerHA SystemMirror services on all nodes

```
                            Start Cluster Services

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

                                                           [Entry Fields]
* Start now, on system restart or both                    now                  +
  Start Cluster Services on these nodes                  [satsspc4,r6r4m51,sats> +
* Manage Resource Groups                                  Manually             +
  BROADCAST message at startup?                           true                 +
  Startup Cluster Information Daemon?                     true                 +
  Ignore verification errors?                             false                +
  Automatically correct errors found during               Interactively        +
  cluster start?


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

We modify the mirror group and the resource groups, and re-adding all hdisks in the configurations. We verify and synchronize the cluster configuration and validate the operation log (in this example: `/var/hacmp/clverify/clverify.log`).

We start bringing the resource group online, as shown in Example 8-159 on page 333.

Example 8-159 Bring the RG online

```
                          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                      ORARG
  Node on Which to Bring Resource Group Online        All_Nodes_in_Group


............................<<snippet>>....................................
```

The operation is shown with the “failed” status (Example 8-160). But in reality, the RG has been brought online.

Example 8-160 RG online command status

```
COMMAND STATUS

Command: failed        stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

[TOP]
Attempting to bring group ORARG online on node ORARG:NONE:satsspc2.
Attempting to bring group ORARG online on node r6r4m51.
Attempting to bring group ORARG online on node ORARG:NONE:satsspc4.
Attempting to bring group ORARG online on node ORARG:NONE:r6r4m52.
No HACMPnode class found with name = ORARG:NONE:satsspc2
Usage: clRMupdate operation [ object ] [ script_name ] [ reference ]
Failed to queue resource group movement event in the cluster manager.
No HACMPnode class found with name = ORARG:NONE:satsspc4
No HACMPnode class found with name = ORARG:NONE:r6r4m52
Usage: clRMupdate operation [ object ] [ script_name ] [ reference ]
Failed to queue resource group movement event in the cluster manager.
Usage: clRMupdate operation [ object ] [ script_name ] [ reference ]
Failed to queue resource group movement event in the cluster manager.

Waiting for the cluster to process the resource group movement request....

Waiting for the cluster to stabilize........

Resource group movement successful.
Resource group ORARG is online on node r6r4m51.

ERROR: Resource Group ORARG did not move to node ORARG:NONE:r6r4m52. It is
currently on node r6r4m51
[MORE...15]
.......................<<snippet>>.................................


root@r6r4m51:/work> clRGinfo
-----------------------------------------------------------------------------
Group Name     State                        Node
```

```
------------------------------------------------------------------------------
ORARG          ONLINE                        r6r4m51@SITE_A
               ONLINE                        r6r4m52@SITE_A
               ONLINE                        satsspc4@SITE_
               ONLINE                        satsspc2@SITE_
```

We validate the paths for the HyperSwap disks, as shown in Example 8-161.

Example 8-161 Validating the HyperSwap disks paths

```
                          Manage User Mirror Group(s)

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

                                                           [Entry Fields]
* Mirror Group(s)                                        ORA_MG                  +
* Operation                                              Show active path        +


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

................................................................................
COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

r6r4m51: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
r6r4m51: ORA_MG:SITE_B:SITE_A:STG_B
r6r4m52: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
r6r4m52: ORA_MG:SITE_B:SITE_A:STG_B
satsspc4: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
satsspc4: ORA_MG:SITE_B:SITE_A:STG_B
satsspc2: MG_NAME:ACTIVE_SITE:SECONDARY_SITE:STORAGE_SYSTEM_ON_ACTIVE_SITE
satsspc2: ORA_MG:SITE_B:SITE_A:STG_B
```

We swap the disks in Storage C in Site A and validate the swap operation log, as shown in Example 8-162.

Example 8-162 Swap to Site A, Storage C

```
INFO      |2014-02-19T04:25:42.521830|Received XD CLI request = '' (0x1d)
INFO      |2014-02-19T04:25:43.522008|Received XD CLI request = 'Swap Mirror
Group' (0x1c)
INFO      |2014-02-19T04:25:43.522037|Request to Swap Mirror Group 'ORA_MG',
Direction 'SITE_A', Outfile ''
INFO      |2014-02-19T04:25:43.523748|No VG found for MG=ORA_MG
INFO      |2014-02-19T04:25:43.523771|No of VG found for MG ORA_MG
INFO      |2014-02-19T04:25:43.523792|Not able to find any VG disks for MG=ORA_MG
INFO      |2014-02-19T04:25:43.663809|Calling sfwGetRepGroupInfo()
INFO      |2014-02-19T04:25:43.663888|sfwGetRepGroupInfo() completed
```

```
INFO     |2014-02-19T04:25:43.663939|Calling sfwGetRepGroupInfo()
INFO     |2014-02-19T04:25:43.663979|sfwGetRepGroupInfo() completed
INFO     |2014-02-19T04:25:43.664010|Calling sfwGetRepGroupInfo()
INFO     |2014-02-19T04:25:43.664056|sfwGetRepGroupInfo() completed
INFO     |2014-02-19T04:25:43.693769|Calling DO_SWAP
INFO     |2014-02-19T04:25:44.863738|DO_SWAP completed
INFO     |2014-02-19T04:25:44.864022|Swap Mirror Group 'ORA_MG' completed.
```

We validate the disk paths, as shown in Example 8-163.

Example 8-163 Validating the disk paths

```
root@r6r4m51:/work> dsh /work/"asm_disks_n.sh" |dshbak -c
HOSTS -------------------------------------------------------------------------
r6r4m52.austin.ibm.com
-------------------------------------------------------------------------------
hdisk59   Active   0(s)         1           500507630affc16b  5005076308ffc6d4
hdisk98   Active   0(s)         1           500507630affc16b  5005076308ffc6d4
hdisk99   Active   0(s)         1           500507630affc16b  5005076308ffc6d4
hdisk101  Active   0(s)         1           500507630affc16b  5005076308ffc6d4
hdisk102  Active   0(s)         1           500507630affc16b  5005076308ffc6d4

HOSTS -------------------------------------------------------------------------
r6r4m51.austin.ibm.com
-------------------------------------------------------------------------------
hdisk49   Active   0(s)         1           500507630affc16b  5005076308ffc6d4
hdisk50   Active   0(s)         1           500507630affc16b  5005076308ffc6d4
hdisk97   Active   0(s)         1           500507630affc16b  5005076308ffc6d4
hdisk99   Active   0(s)         1           500507630affc16b  5005076308ffc6d4
hdisk100  Active   0(s)         1           500507630affc16b  5005076308ffc6d4

HOSTS -------------------------------------------------------------------------
satsspc2.austin.ibm.com
-------------------------------------------------------------------------------
hdisk101  Active   0(s)         1           500507630affc16b  5005076308ffc6d4
hdisk102  Active   0(s)         1           500507630affc16b  5005076308ffc6d4
hdisk103  Active   0(s)         1           500507630affc16b  5005076308ffc6d4
hdisk105  Active   0(s)         1           500507630affc16b  5005076308ffc6d4
hdisk106  Active   0(s)         1           500507630affc16b  5005076308ffc6d4

HOSTS -------------------------------------------------------------------------
satsspc4.austin.ibm.com
-------------------------------------------------------------------------------
hdisk98   Active   0(s)         1           500507630affc16b  5005076308ffc6d4
hdisk99   Active   0(s)         1           500507630affc16b  5005076308ffc6d4
hdisk101  Active   0(s)         1           500507630affc16b  5005076308ffc6d4
hdisk103  Active   0(s)         1           500507630affc16b  5005076308ffc6d4
hdisk104  Active   0(s)         1           500507630affc16b  5005076308ffc6d4

root@r6r4m51:/work> lspprc -v hdisk49

HyperSwap lun unique
identifier.............352037354c593938313746303100530483a707210790003IBMfcp

hdisk49 Primary         MPIO IBM 2107 FC Disk
```

```
        Manufacturer..................IBM
        Machine Type and Model........2107900
        ROS Level and ID..............2E393330
        Serial Number.................75TL771A
        Device Specific.(Z7)..........A204
        Device Specific.(Z0)..........000005329F101002
        Device Specific.(Z1)..........204
        Device Specific.(Z2)..........075
        Unique Device Identifier......200B75TL771A20407210790003IBMfcp
        Logical Subsystem ID..........0xa2
        Volume Identifier.............0x04
        Subsystem Identifier(SS ID)...0xFFA2
        Control Unit Sequence Number..00000TL771
        Storage Subsystem WWNN........500507630affc16b
        Logical Unit Number ID........40a2400400000000

hdisk49 Secondary       MPIO IBM 2107 FC Disk

        Manufacturer..................IBM
        Machine Type and Model........2107900
        ROS Level and ID..............2E313336
        Serial Number.................75LY9817
        Device Specific.(Z7)..........7F01
        Device Specific.(Z0)..........000005329F101002
        Device Specific.(Z1)..........F01
        Device Specific.(Z2)..........075
        Unique Device Identifier......200B75LY9817F0107210790003IBMfcp
        Logical Subsystem ID..........0x7f
        Volume Identifier.............0x01
        Subsystem Identifier(SS ID)...0xFF7F
        Control Unit Sequence Number..00000LY981
        Storage Subsystem WWNN........5005076308ffc6d4
        Logical Unit Number ID........407f400100000000
```

We have now finished the storage migration, so we validate how the database behaved during the entire configuration process. The Enterprise Manager Console graphic is shown in Figure 8-16.

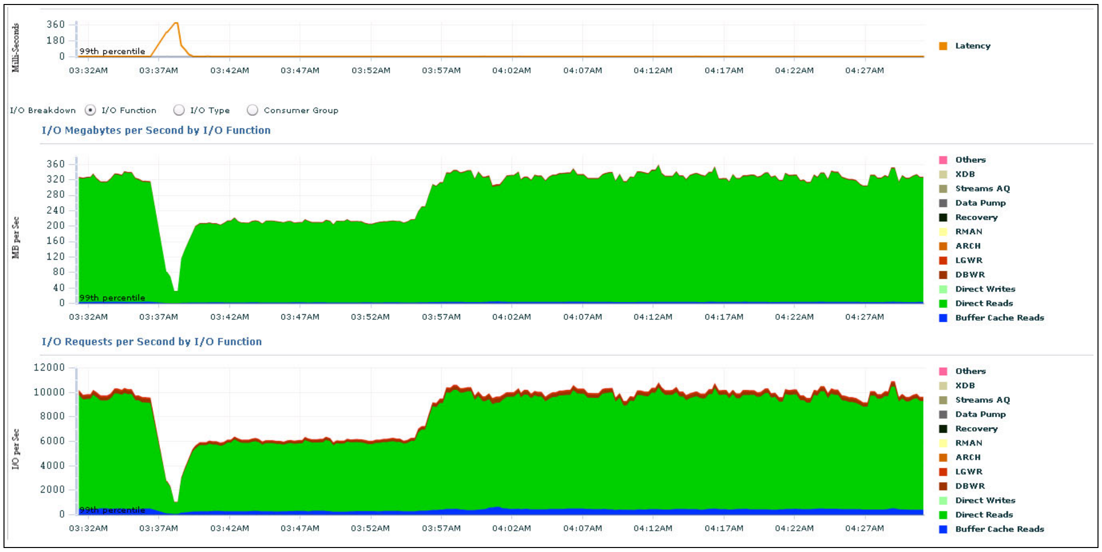


Figure 8-16 Graphical display of the cluster behavior using the Enterprise Manager Console

At 3:37 AM, an event was observed that decreased the load activity. This can easily be associated with the disk reconfiguration operation that was underway at that time. The results of this event are similar to a planned HperSwap and removal of a PPRC relationship. The decrease in the database load activity is directly related to the copy operations, as shown in Example 8-154 on page 330. The database load reverts back to the original start value after the copy operations have completed, as reflected by the Full Duplex state of the disk pairs.

8.22 Troubleshooting HyperSwap

HyperSwap events and various messages are logged in the following files:

- hacmp.out (`/var/hacmp/log/hacmp.out`)

  Displays messages that are related to detection, migration, termination, and execution of a PowerHA SystemMirror cluster for an application.

- clutils.log (`/var/hacmp/log/clutils.log`)

  Displays the results of the automatic verification that runs on a specified PowerHA SystemMirror cluster node every 24 hours.

- clxd_debug.log (`/var/hacmp/xd/log/clxd_debug.log`)

  Displays information about the **`clxd`** daemon.

You can also configure the kernel extension to create debug logs in the `/etc/syslog.conf` file by completing the following steps:

1. In the `/etc/syslog.conf` file, add the line shown in Example 8-164.

   Example 8-164 Logging HyperSwap by kern.debug extension

```
kern.debug /var/hacmp/xd/log/syslog.phake rotate size 500k files 7
```

2. Create a file called `syslog.phake` in the `/var/hacmp/xd/log` directory.

3. Refresh the **syslogd** daemon.

Note: The debug logs are also logged in the console. For unplanned operations, all events appear in the /var/hacmp/xd/log/syslog.phake file.

HyperSwap configurations use kernel extensions. Therefore, you can view error or warning messages from the kernel extensions by using the **errpt** command, as shown in Example 8-165.

Example 8-165 Errpt events

```
EA94555F   0209143814 I H pha_1065481201 PPRC Failover Completed
EA94555F   0209140214 I H pha_1065481201 PPRC Failover Completed
EA94555F   0209140214 I H hdisk100       PPRC Failover Completed
F31FFAC3   0209140114 I H hdisk70        PATH HAS RECOVERED
F31FFAC3   0209140114 I H hdisk69        PATH HAS RECOVERED
EA94555F   0209140114 I H pha_9654771971 PPRC Failover Completed
EA94555F   0209140114 I H hdisk41        PPRC Failover Completed
EA94555F   0209140114 I H hdisk42        PPRC Failover Completed
EA94555F   0209140114 I H pha_9654761964 PPRC Failover Completed
```

9

RBAC integration and implementation

In this chapter, we describe role-based access control (RBAC), which is a major component of federated security for IBM PowerHA SystemMirror. We cover the following topics:

- PowerHA SystemMirror federated security
- Components and planning
- Installation and configuration
- Testing and administration
- Customized method to achieve basic RBAC functions

After reading this chapter, you will understand how to integrate RBAC into a PowerHA SystemMirror environment from scratch.

9.1 PowerHA SystemMirror federated security

The IBM AIX operating system provides a rich set of security capabilities. However, similar capabilities were previously missing in clustered environments. The PowerHA SystemMirror federated security feature was introduced in PowerHA SystemMirror 7.1.1. The goal of federated security is to enable the security administration of AIX security features across the cluster.

Federated security is a centralized tool that addresses Lightweight Directory Access Protocol (LDAP), role-based access control (RBAC), and Encrypted File System (EFS) integration into cluster management.

With federated security, you can complete the following tasks:

- Configure and manage an IBM or non-IBM LDAP server as a centralized information base.
- Configure and manage a peer-to-peer IBM LDAP server.
- Configure and manage the LDAP client for all of the nodes of the cluster.
- Create and manage a highly available EFS file system.
- Create and manage RBAC roles for users and groups. You can use these roles to control which commands can be executed by different sets of users of PowerHA SystemMirror.

Through the federated security cluster, users can manage roles and the encryption of data across the cluster.

9.2 Components and planning

The major components of federated security are LDAP, RBAC, and EFS. Because the PowerHA SystemMirror roles and the EFS keystore are stored in the LDAP server, in order to use the feature, your environment must have an LDAP server. Two types of LDAP servers are supported:

- IBM Security Directory Server (formerly called IBM Tivoli Directory Server)
- Microsoft Windows Server Active Directory

In this book, we focus on the IBM Tivoli Directory Server software. PowerHA SystemMirror includes an option to configure the LDAP server on cluster nodes for which at least two cluster nodes are required for peer-to-peer replicated LDAP server setup. You can find the detailed steps for this configuration in 9.3.1, "Peer-to-peer replicated LDAP server scenario" on page 341. Depending on your environment, you can also configure the LDAP server on a node outside of the cluster.

For external LDAP server, cluster nodes need be configured only as an LDAP client. For detailed steps for this configuration, see 9.3.2, "External LDAP server scenario" on page 345.

9.2.1 Components

LDAP enables centralized security authentication, access to user and group information, and common authentication, user, and group information across the cluster.

RBAC enables assignment of daily cluster activities to specific predefined roles (administrator, operator, monitor, and viewer). These roles can then be associated with specific users to permit them to perform these activities.

EFS enables users to encrypt their data through a keystore that is specific to that user. When a process opens an EFS-protected file, these credentials are tested to verify that they match the file protection. If successful, the process is able to decrypt the file key and, therefore, the file content.

9.2.2 Planning

Before you can use the features of federated security, you must plan for its implementation in your environment.

In the example in following sections of this chapter, we are using a two-node cluster to illustrate the setup. The environment must meet the following requirements:

- The AIX operating system must be at one of the following technology levels:
  - IBM AIX 6.1 with Technology Level 7 or later
  - IBM AIX 7.1 with Technology Level 1 or later
- PowerHA SystemMirror Version 7.1.1 or later
- IBM Tivoli Directory Server 6.2 or later

Note: IBM Tivoli Directory Server is included with AIX base media.

9.3 Installation and configuration

This section explains the detailed steps of setting up IBM DB2, Global Secure Toolkit (GSKit), and Tivoli Directory Server. It also shows the configuration of each component. DB2 V9.7 installation files, GSKit file sets, and Tivoli Directory Server 6.3 are in the AIX Expansion Pack. In our example, we are using AIX Enterprise Edition V7.1 Expansion Pack (112013). It contains DB2 V9.7 Fix Pack 8, GSKit 8.0.50.10, and Tivoli Directory Server 6.3.0.24.

9.3.1 Peer-to-peer replicated LDAP server scenario

Follow these steps to install and configure peer-to-peer replicated LDAP servers:

1. Install the DB2 V9.7 package on two cluster nodes.
2. Install the GSKit on all of the cluster nodes.
3. Install the Tivoli Directory Server server and client on two cluster nodes.
4. Configure peer-to-peer replicated LDAP servers on two cluster nodes.
5. Configure LDAP clients on all of the cluster nodes.

Install DB2 on two cluster nodes

The DB2 installation steps are shown in Example 9-1 on page 342.

Example 9-1 DB2 installation steps

```
# ./db2_install

Default directory for installation of products - /opt/IBM/db2/V9.7
***********************************************************
Do you want to choose a different directory to install [yes/no] ?
no


Specify one of the following keywords to install DB2 products.
  AESE
  ESE
  CONSV
  WSE
  CLIENT
  RTCL

Enter "help" to redisplay product names.
Enter "quit" to exit.
***********************************************************
ESE
DB2 installation is being initialized.

 Total number of tasks to be performed: 46
Total estimated time for all tasks to be performed: 1890

Task #1 start
Description: Enable IOCP
Estimated time 1 second(s)
Task #1 end
...
...
Task #46 start
Description: Updating global profile registry
Estimated time 3 second(s)
Task #46 end

The execution completed successfully.

For more information see the DB2 installation log at
"/tmp/db2_install.log.8126548".

# /usr/local/bin/db2ls

Install Path                       Level   Fix Pack   Special Install Number
Install Date                 Installer UID
---------------------------------------------------------------------------------
-----------------------------------
/opt/IBM/db2/V9.7            9.7.0.8      8 Mon Nov 25 04:02:13 2013 EST
0
```

Install GSKit on all of the cluster nodes

The GSKit installation steps are shown in Example 9-2.

Example 9-2 GSKit installation

```
# installp -acgXd . GSKit8.gskcrypt32.ppc.rte
# installp -acgXd . GSKit8.gskssl32.ppc.rte
# installp -acgXd . GSKit8.gskcrypt64.ppc.rte
# installp -acgXd . GSKit8.gskssl64.ppc.rte

# lslpp -l | grep GSKit
  GSKit8.gskcrypt32.ppc.rte
                           8.0.50.10  COMMITTED  IBM GSKit Cryptography Runtime
  GSKit8.gskcrypt64.ppc.rte
                           8.0.50.10  COMMITTED  IBM GSKit Cryptography Runtime
  GSKit8.gskssl32.ppc.rte  8.0.50.10  COMMITTED  IBM GSKit SSL Runtime With
  GSKit8.gskssl64.ppc.rte  8.0.50.10  COMMITTED  IBM GSKit SSL Runtime With
```

Install the Tivoli Directory Server server and client on two cluster nodes

The Tivoli Directory Server server and client installation steps are shown in Example 9-3.

Example 9-3 Tivoli Directory Server server and client file sets installation

```
Install idsLicense in the /license directory from the AIX Expansion DVD.
# /license/idsLicense
International Program License Agreement

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, CLICKING ON AN "ACCEPT"
BUTTON, OR OTHERWISE USING THE PROGRAM, LICENSEE AGREES TO THE TERMS OF THIS
AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF LICENSEE, YOU
REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND LICENSEE TO THESE
TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

* DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, CLICK ON AN "ACCEPT" BUTTON, OR USE
THE PROGRAM; AND

* PROMPTLY RETURN THE UNUSED MEDIA, DOCUMENTATION, AND PROOF OF ENTITLEMENT TO
THE PARTY FROM WHOM IT WAS OBTAINED FOR A REFUND OF THE AMOUNT PAID. IF THE
PROGRAM WAS DOWNLOADED, DESTROY ALL COPIES OF THE PROGRAM.

Press Enter to continue viewing the license agreement, or, Enter "1" to accept
the agreement, "2" to decline it or "99" to go back to the previous screen,
"3" Print.
1

Install ldap server and client filesets.
# installp -acgXd . idsldap.license63
# installp -acgXd . idsldap.srvbase64bit63
# installp -acgXd . idsldap.srv64bit63
# installp -acgXd . idsldap.srv_max_cryptobase64bit63
# installp -acgXd . idsldap.msg63.en_US
# installp -acgXd . idsldap.srvproxy64bit63
# installp -acgXd . idsldap.cltbase63
```

```
# installp -acgXd . idsldap.clt32bit63
# installp -acgXd . idsldap.clt64bit63
# installp -acgXd . idsldap.clt_max_crypto32bit63
# installp -acgXd . idsldap.clt_max_crypto64bit63
# installp -acgXd . idsldap.cltjava63

# lslpp -l | grep ldap
  idsldap.clt32bit63.rte    6.3.0.24  COMMITTED  Directory Server - 32 bit
  idsldap.clt64bit63.rte    6.3.0.24  COMMITTED  Directory Server - 64 bit
  idsldap.clt_max_crypto32bit63.rte
  idsldap.clt_max_crypto64bit63.rte
  idsldap.cltbase63.adt     6.3.0.24  COMMITTED  Directory Server - Base Client
  idsldap.cltbase63.rte     6.3.0.24  COMMITTED  Directory Server - Base Client
  idsldap.cltjava63.rte     6.3.0.24  COMMITTED  Directory Server - Java Client
  idsldap.license63.rte     6.3.0.24  COMMITTED  Directory Server - License
  idsldap.msg63.en_US       6.3.0.24  COMMITTED  Directory Server - Messages -
  idsldap.srv64bit63.rte    6.3.0.24  COMMITTED  Directory Server - 64 bit
  idsldap.srv_max_cryptobase64bit63.rte
  idsldap.srvbase64bit63.rte
  idsldap.srvproxy64bit63.rte
  idsldap.clt32bit63.rte    6.3.0.24  COMMITTED  Directory Server - 32 bit
  idsldap.clt64bit63.rte    6.3.0.24  COMMITTED  Directory Server - 64 bit
  idsldap.cltbase63.rte     6.3.0.24  COMMITTED  Directory Server - Base Client
  idsldap.srvbase64bit63.rte
  idsldap.srvproxy64bit63.rte
```

Configure peer-to-peer replicated LDAP servers on two cluster nodes

The peer-to-peer replicated LDAP server configuration steps are shown in Example 9-4.

Example 9-4 Using cl_ldap_server_config to configure peer-to-peer replicated LDAP servers

```
Create a directory called /newkeys on both cluster nodes.
# mkdir /newkeys
# /usr/es/sbin/cluster/cspoc/cl_ldap_server_config -h lpar0104,lpar0204 -a
cn=admin -w adminpwd -s rfc2307aix -d cn=aixdata,o=ibm -p 636 -S
/newkeys/serverkey.kdb -W serverpwd -V 6.3 -X db2pwd -E 123456789012
INFO: Running ldap server configuration on lpar0104, please wait...
Machine Hardware is 64 bit.
Kernel is 64 bit enabled.
DB2 Version 9.7.0.8 installed on this system, continuing configuration...
ITDS server version 6.3.0.24 is compatible, continuing configuration...
ITDS client version 6.3.0.24 is compatible, continuing configuration...
INFO: Running mksecldap on lpar0104, it may take quite a bit of time...
Keys and certificates exists...
INFO: Running ldap server configuration on lpar0204, please wait...
Machine Hardware is 64 bit.
Kernel is 64 bit enabled.
DB2 Version 9.7.0.8 installed on this system, continuing configuration...
ITDS server version 6.3.0.24 is compatible, continuing configuration...
ITDS client version 6.3.0.24 is compatible, continuing configuration...
INFO: Running mksecldap on lpar0204, it may take quite a bit of time...
Keys and certificates exists...
Restarting server on lpar0104 node, please wait...
Restarting server on lpar0204 node, please wait...
```

```
Operation completed successfully. Details: 1 servers replicated successfully out
of 1 attempts.
The PowerHA SystemMirror configuration has been changed - LDAP Server configure
has been done.  The configuration must be synchronized to make this change
effective across the cluster. Run verification and Synchronization.

Run cluster verification and synchronization:
# smitty sysmirror
```

Configure LDAP clients on all of the cluster nodes

The LDAP client configuration steps are shown in Example 9-5.

Example 9-5 Using cl_ldap_client_config command to configure LDAP clients

```
# /usr/es/sbin/cluster/cspoc/cl_ldap_client_config -h lpar0104,lpar0204 -a
cn=admin -w adminpwd -d cn=aixdata,o=ibm -p 636 -S /newkeys/clientkey.kdb -W
clientpwd
INFO: Running ldap client configuration on lpar0104, please wait...
ITDS client version 6.3.0.24 is compatible, continuing configuration...
The secldapclntd daemon is not running.
Starting the secldapclntd daemon.
The secldapclntd daemon started successfully.
INFO: Running ldap client configuration on lpar0204, please wait...
ITDS client version 6.3.0.24 is compatible, continuing configuration...
The secldapclntd daemon is not running.
Starting the secldapclntd daemon.
The secldapclntd daemon started successfully.
INFO: Running RBAC configuration, it may take quite a bit of time, please wait...
Authorization "PowerHASM" exists.
Authorization "PowerHASM.admin" exists.
Authorization "PowerHASM.mon" exists.
Authorization "PowerHASM.op" exists.
Authorization "PowerHASM.view" exists.
The PowerHA SystemMirror configuration has been changed - LDAP Client configure
has been done.  The configuration must be synchronized to make this change
effective across the cluster. Run verification and Synchronization.

Run cluster verification and synchronization:
# smitty sysmirror
```

9.3.2 External LDAP server scenario

Follow these steps to install and configure RBAC by using an external LDAP server:

1. Install the DB2 V9.7 package on the LDAP server.
2. Install the GSKit on the LDAP server and all of the cluster nodes.
3. Install the Tivoli Directory Server server and client on the LDAP server.
4. Install the Tivoli Directory Server client on all of the cluster nodes.
5. Create the server keys.
6. Create the client keys.
7. Configure the LDAP server.

8. Configure the LDAP server and client on all the cluster nodes.

Install DB2 on the LDAP server

The DB2 installation steps are shown in Example 9-1 on page 342.

Install the GSKit on the LDAP server and all of the cluster nodes

The GSKit installation steps are shown in Example 9-2 on page 343.

Install the Tivoli Directory Server server and client on the LDAP server

The Tivoli Directory Server server and client installation steps are shown in Example 9-3 on page 343.

Install the Tivoli Directory Server client on all of the cluster nodes

The Tivoli Directory Server client installation steps are shown in Example 9-6.

Example 9-6 Tivoli Directory Server client file sets installation

```
Install idsLicense in the /license directory from the AIX Expansion DVD.
# /license/idsLicense
International Program License Agreement

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, CLICKING ON AN "ACCEPT"
BUTTON, OR OTHERWISE USING THE PROGRAM, LICENSEE AGREES TO THE TERMS OF THIS
AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF LICENSEE, YOU
REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND LICENSEE TO THESE
TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

* DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, CLICK ON AN "ACCEPT" BUTTON, OR USE
THE PROGRAM; AND

* PROMPTLY RETURN THE UNUSED MEDIA, DOCUMENTATION, AND PROOF OF ENTITLEMENT TO
THE PARTY FROM WHOM IT WAS OBTAINED FOR A REFUND OF THE AMOUNT PAID. IF THE
PROGRAM WAS DOWNLOADED, DESTROY ALL COPIES OF THE PROGRAM.

Press Enter to continue viewing the license agreement, or, Enter "1" to accept
the agreement, "2" to decline it or "99" to go back to the previous screen,
"3" Print.
1

# installp -acgXd . idsldap.license63
# installp -acgXd . idsldap.cltbase63
# installp -acgXd . idsldap.clt32bit63
# installp -acgXd . idsldap.clt64bit63
# installp -acgXd . idsldap.clt_max_crypto32bit63
# installp -acgXd . idsldap.clt_max_crypto64bit63
# installp -acgXd . idsldap.cltjava63

# lslpp -l | grep ldap
  idsldap.clt32bit63.rte    6.3.0.24  COMMITTED  Directory Server - 32 bit
  idsldap.clt64bit63.rte    6.3.0.24  COMMITTED  Directory Server - 64 bit
  idsldap.clt_max_crypto32bit63.rte
  idsldap.clt_max_crypto64bit63.rte
```

```
  idsldap.cltbase63.adt     6.3.0.24  COMMITTED  Directory Server - Base Client
  idsldap.cltbase63.rte     6.3.0.24  COMMITTED  Directory Server - Base Client
  idsldap.cltjava63.rte     6.3.0.24  COMMITTED  Directory Server - Java Client
  idsldap.license63.rte     6.3.0.24  COMMITTED  Directory Server - License
  idsldap.clt32bit63.rte    6.3.0.24  COMMITTED  Directory Server - 32 bit
  idsldap.clt64bit63.rte    6.3.0.24  COMMITTED  Directory Server - 64 bit
  idsldap.cltbase63.rte     6.3.0.24  COMMITTED  Directory Server - Base Client
```

Create the server keys

The steps for creating the server keys are shown in Example 9-7.

Example 9-7 Using the gsk8capicmd_64 command to create server keys

```
# mkdir /newkeys
# /usr/bin/gsk8capicmd_64 -keydb -create -db /newkeys/serverkey.kdb -pw serverpwd
-type cms -stash
# ls -l /newkeys
total 32
-rw-------    1 root     system           88 Nov 28 21:40 serverkey.crl
-rw-------    1 root     system           88 Nov 28 21:40 serverkey.kdb
-rw-------    1 root     system           88 Nov 28 21:40 serverkey.rdb
-rw-------    1 root     system          129 Nov 28 21:40 serverkey.sth
# /usr/bin/gsk8capicmd_64 -cert -create -db /newkeys/serverkey.kdb -pw serverpwd
-label SERVER_CERT -dn "cn=`hostname`" -default_cert yes
# ls -l /newkeys
total 40
-rw-------    1 root     system           88 Nov 28 21:40 serverkey.crl
-rw-------    1 root     system         5088 Nov 28 22:01 serverkey.kdb
-rw-------    1 root     system           88 Nov 28 21:40 serverkey.rdb
-rw-------    1 root     system          129 Nov 28 21:40 serverkey.sth
# /usr/bin/gsk8capicmd_64 -cert -extract -db /newkeys/serverkey.kdb -pw serverpwd
-label SERVER_CERT -target /newkeys/serverkey.arm -format binary
# ls -l /newkeys
total 48
-rw-r--r--    1 root     system          408 Nov 28 22:06 serverkey.arm
-rw-------    1 root     system           88 Nov 28 21:40 serverkey.crl
-rw-------    1 root     system         5088 Nov 28 22:01 serverkey.kdb
-rw-------    1 root     system           88 Nov 28 21:40 serverkey.rdb
-rw-------    1 root     system          129 Nov 28 21:40 serverkey.sth
```

Create the client keys

The steps for creating the client keys are shown in Example 9-8.

Example 9-8 Using the gsk8capicmd_64 command to create client keys

```
# /usr/bin/gsk8capicmd_64 -keydb -create -db /newkeys/clientkey.kdb -pw clientpwd
-type cms -stash
# ls -l /newkeys
total 32
-rw-------    1 root     system           88 Nov 28 22:44 clientkey.crl
-rw-------    1 root     system           88 Nov 28 22:44 clientkey.kdb
-rw-------    1 root     system           88 Nov 28 22:44 clientkey.rdb
-rw-------    1 root     system          129 Nov 28 22:44 clientkey.sth
```

```
Copy the serverkey.kdb and serverkey.arm from LDAP server to client under
/newkeys.
# /usr/bin/gsk8capicmd_64 -cert -add -db /newkeys/clientkey.kdb -pw clientpwd
-label SERVER_CERT -file /newkeys/serverkey.arm -format binary
# ls -l /newkeys
total 48
-rw-------    1 root     system           88 Nov 28 22:37 clientkey.crl
-rw-------    1 root     system         5088 Nov 28 22:48 clientkey.kdb
-rw-------    1 root     system           88 Nov 28 22:37 clientkey.rdb
-rw-------    1 root     system          129 Nov 28 22:37 clientkey.sth
-rw-r--r--    1 root     system          408 Nov 28 22:40 serverkey.arm
```

Configure the LDAP server

The LDAP server configuration steps are shown in Example 9-9.

Example 9-9 Using the mksecldap -s command to configure LDAP server

```
# mksecldap -s -a cn=admin -p adminpwd -S rfc2307aix -d cn=aixdata,o=ibm -k
/newkeys/serverkey.kdb -w serverpwd
The user "ldapdb2" has an invalid lastupdate attribute.
ldapdb2's New password: db2pwd
Enter the new password again: db2pwd
Enter an encryption seed to generate key stash files: 123456789012
Error opening toollibs.cat
GLPWRP123I The program '/opt/IBM/ldap/V6.3/sbin/64/idsicrt' is used with the
following arguments 'idsicrt -I ldapdb2 -s 636 -e ***** -n'.
You have chosen to perform the following actions:

# ps -eaf | grep ldap
    root 3997778 4128848   0 23:44:24  pts/0  0:00 /bin/ksh /usr/sbin/mksecldap -s
-a cn=admin -p adminpwd -S rfc2307aix -d cn=aixdata,o=ibm -k
/newkeys/serverkey.kdb -w serverpwd
 ldapdb2 5898366 4063482   0 00:01:30      -  0:00 db2fmp (C) 0
 ldapdb2 2097654       1   0 23:54:41  pts/1  0:00
/opt/IBM/ldap/V6.3/sbin/64/ibmdiradm -I ldapdb2
 ldapdb2 2425236 4063482   0 23:56:27      -  0:01 db2acd 0
 ldapdb2 3408200       1   0 23:57:14  pts/1  0:13
/opt/IBM/ldap/V6.3/sbin/64/ibmslapd -I ldapdb2 -f
/home/ldapdb2/idsslapd-ldapdb2/etc/ibmslapd.conf
 ldapdb2 3801502 4063482   0 23:56:26      -  0:03 db2sysc 0
```

Configure the LDAP server and client on all of the cluster nodes

Steps for configuring the LDAP server and client are shown in Example 9-10.

Example 9-10 Using cl_ldap_client_config command to configure LDAP clients

```
To test the communication between LDAP server and client:
# gsk8capicmd_64 -cert -list -db /newkeys/clientkey.kdb -pw clientpwd
Certificates found
* default, - personal, ! trusted
!       SERVER_CERT

To define LDAP server on cluster node, you can do it from C-SPOC or command line:
# /usr/es/sbin/cluster/cspoc/cl_ldap_server_existing -h a3 -a cn=admin -w adminpwd
-d cn=aixdata,o=ibm -p 636 -S /newkeys/serverkey.kdb -W serverpwd
```

```
ITDS client version 6.3.0.24 is compatible, continuing configuration...
RSH service failed with an error on a3, continuing assuming server already updated
with relevant schemas and data...
The PowerHA SystemMirror configuration has been changed - LDAP server configure
has been done.  The configuration must be synchronized to make this change
effective across the cluster. Run verification and Synchronization.

Run cluster verification and synchronization:
# smitty sysmirror

# /usr/es/sbin/cluster/cspoc/cl_ldap_client_config -h a3 -a cn=admin -w adminpwd
-d cn=aixdata,o=ibm -p 636 -S /newkeys/clientkey.kdb -W clientpwd
INFO: Running ldap client configuration on a6, please wait...
ITDS client version 6.3.0.24 is compatible, continuing configuration...
Keys and certificates exists...
The secldapclntd daemon is not running.
Starting the secldapclntd daemon.
The secldapclntd daemon started successfully.
INFO: Running ldap client configuration on b6, please wait...
ITDS client version 6.3.0.24 is compatible, continuing configuration...
Keys and certificates exists...
The secldapclntd daemon is not running.
Starting the secldapclntd daemon.
The secldapclntd daemon started successfully.
INFO: Running RBAC configuration, it may take quite a bit of time, please wait...
Authorization "PowerHASM" exists.
Authorization "PowerHASM.admin" exists.
Authorization "PowerHASM.mon" exists.
Authorization "PowerHASM.op" exists.
Authorization "PowerHASM.view" exists.
Role "ha_admin" exists.
Role "ha_op" exists.
Role "ha_mon" exists.
Role "ha_view" exists.
The PowerHA SystemMirror configuration has been changed - LDAP Client configure
has been done.  The configuration must be synchronized to make this change
effective across the cluster. Run verification and Synchronization.

Run cluster verification and synchronization:
# smitty sysmirror
```

9.4 Testing and administration

During LDAP client configuration, four roles defined by PowerHA are created in LDAP:

ha_op (for operations): Provides *operator* authorization for the relevant cluster functions. For example, “move cluster resource group” is under operator authorization.

ha_admin (for administrator): Provides *admin* authorization for the relevant cluster functions. For example, “creating a cluster snapshot” is under administrator authorization.

ha_view (for viewer)	Provides *view* authorization. It has all read permissions for the cluster functions. For example, “read hacmp.out file” is under viewer authorization.
ha_mon (for monitor)	Provides *monitor* authorization for the relevant cluster functions. For example, the **clRGinfo** command is under monitor authorization.

These roles can be assigned to the user to provide restricted access to the cluster functions, based on the role.

User management is in the PowerHA SystemMirror Cluster Single Point of Control (C-SPOC), which is shown in Figure 9-1. To reach user management, enter **smitty sysmirror** and select **System Management (C-SPOC)** → **Security and Users** → **Users in an PowerHA SystemMirror cluster**.

```
                     Users in an PowerHA SystemMirror cluster

Move cursor to desired item and press Enter.

  Add a User to the Cluster
  Change / Show Characteristics of a User in the Cluster
  Remove a User from the Cluster
  List Users in the Cluster

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

Figure 9-1 PowerHA SystemMirror user management

To create a user, you can set the authentication and registry mode to either LOCAL(FILES) or LDAP, as shown in Figure 9-2 on page 351.

```
                     Users in an PowerHA SystemMirror cluster

Move cursor to desired item and press Enter.

  Add a User to the Cluster
  Change / Show Characteristics of a User in the Cluster
  Remove a User from the Cluster
  List Users in the Cluster


  +--------------------------------------------------------------------------+
  |                Select an Authentication and registry mode                |
  |                                                                          |
  | Move cursor to desired item and press Enter.                             |
  |                                                                          |
  |   LOCAL(FILES)                                                           |
  |   LDAP                                                                   |
  |                                                                          |
  | F1=Help                F2=Refresh             F3=Cancel                  |
  | F8=Image               F10=Exit               Enter=Do                   |
F1| /=Find                 n=Find Next                                       |
F9+--------------------------------------------------------------------------+
```

Figure 9-2 Add a user to the cluster

You can assign the PowerHA SystemMirror RBAC roles to the new user as shown in Figure 9-3 on page 352.

```
                              Add a User to the LDAP

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

[TOP]                                                  [Entry Fields]
* User NAME                                         []
  User ID                                           []                     #
* Roles                                             [ha_admin]            +
* Registry                                           LDAP
* Login Authentication Grammar                       LDAP
  Keystore Access                                    LDAP
  ADMINISTRATIVE USER?                               false                +
  Primary GROUP                                     []                    +
  Group SET                                         []                    +
  ADMINISTRATIVE GROUPS                             []                    +
  Another user can SU TO USER?                       true                 +
  SU GROUPS                                         [ALL]                 +
  HOME directory                                    []
[MORE...38]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 9-3 Adding a new user with ha_admin role

In this section, we create four non-root users and assign these different RBAC roles to them:

- haOp - ha_op
- haAdmin - ha_admin
- haView - ha_view
- haMon - ha_mon

We use the following four examples to illustrate how the four RBAC roles can be used for some PowerHA SystemMirror functions.

ha_op role example

Moving cluster resource group by a non-root user with ha_op role is shown in Example 9-11.

Example 9-11 Moving cluster resource group by a non-root user with ha_op role

```
# lsuser haOp
haOp id=208 pgrp=staff groups=staff home=/home/haOp shell=/usr/bin/ksh login=true
su=true rlogin=true telnet=true daemon=true admin=false sugroups=ALL admgroups=
tpath=nosak ttys=ALL expires=0 auth1=SYSTEM auth2=NONE umask=22 registry=LDAP
SYSTEM=LDAP logintimes= loginretries=0 pwdwarntime=0 account_locked=false minage=0
maxage=0 maxexpired=-1 minalpha=0 minloweralpha=0 minupperalpha=0 minother=0
mindigit=0 minspecialchar=0 mindiff=0 maxrepeats=8 minlen=0 histexpire=0
histsize=0 pwdchecks= dictionlist= default_roles= fsize=2097151 cpu=-1 data=262144
stack=65536 core=2097151 rss=65536 nofiles=2000 roles=ha_op

# su - haOp
$ whoami
```

```
haOp
$ swrole ha_op
haOp's Password:
$ rolelist -e
ha_op
```

To move a cluster resource group, enter **smitty sysmirror** and select **System Management (C-SPOC)** → **Resource Group and Applications** → **Move Resource Groups to Another Node** and select the resource group and destination node. The task completes successfully with the result as shown in Figure 9-4.

```
                                COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

[TOP]
Attempting to move resource group rg04 to node lpar0204.

Waiting for the cluster to process the resource group movement request....

Waiting for the cluster to stabilize......

Resource group movement successful.
Resource group rg04 is online on node lpar0204.



Cluster Name: cluster04
[MORE...7]

F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next
```

Figure 9-4 Moving cluster resource group result

ha_admin role example

An example of creating a cluster snapshot by a non-root user with ha_admin role is shown in Example 9-12.

Example 9-12 Creating a cluster snapshot by a non-root user with ha_admin role

```
# lsuser haAdmin
haAdmin id=207 pgrp=staff groups=staff home=/home/haAdmin shell=/usr/bin/ksh
login=true su=true rlogin=true telnet=true daemon=true admin=false sugroups=ALL
admgroups= tpath=nosak ttys=ALL expires=0 auth1=SYSTEM auth2=NONE umask=22
registry=LDAP SYSTEM=LDAP logintimes= loginretries=0 pwdwarntime=0
account_locked=false minage=0 maxage=0 maxexpired=-1 minalpha=0 minloweralpha=0
minupperalpha=0 minother=0 mindigit=0 minspecialchar=0 mindiff=0 maxrepeats=8
minlen=0 histexpire=0 histsize=0 pwdchecks= dictionlist= default_roles=
fsize=2097151 cpu=-1 data=262144 stack=65536 core=2097151 rss=65536 nofiles=2000
roles=ha_admin
```

```
# su - haAdmin
$ whoami
haAdmin
$ swrole ha_admin
haAdmin's Password:
$ rolelist -e
ha_admin
```

To create a cluster snapshot, enter **smitty sysmirror** and select **Cluster Nodes and Networks** → **Manage the Cluster** → **Snapshot Configuration** → **Create a Cluster Snapshot of the Cluster Configuration** as shown in Figure 9-5.

```
                  Create a Snapshot of the Cluster Configuration

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

                                                          [Entry Fields]
* Cluster Snapshot Name                                [testCluster]           /
  Custom Defined Snapshot Methods                      []                     +
* Cluster Snapshot Description                         [To test ha_admin role]


F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 9-5 Create a cluster snapshot

ha_view role example

An example of reading the `hacmp.out` file by a non-root user with the ha_view role is shown in Example 9-13.

Example 9-13 Reading the hacmp.out file by a non-root user with the ha_view role

```
# lsuser haView
haView id=210 pgrp=staff groups=staff home=/home/haView shell=/usr/bin/ksh
login=true su=true rlogin=true telnet=true daemon=true admin=false sugroups=ALL
admgroups= tpath=nosak ttys=ALL expires=0 auth1=SYSTEM auth2=NONE umask=22
registry=LDAP SYSTEM=LDAP logintimes= loginretries=0 pwdwarntime=0
account_locked=false minage=0 maxage=0 maxexpired=-1 minalpha=0 minloweralpha=0
minupperalpha=0 minother=0 mindigit=0 minspecialchar=0 mindiff=0 maxrepeats=8
minlen=0 histexpire=0 histsize=0 pwdchecks= dictionlist= default_roles=
fsize=2097151 cpu=-1 data=262144 stack=65536 core=2097151 rss=65536 nofiles=2000
roles=ha_view

# su - haView
$ whoami
haView
$ swrole ha_view
haView's Password:
$ rolelist -e
ha_view
$ pvi /var/hacmp/log/hacmp.out
```

```
Warning: There is no cluster found.
Warning: There is no cluster found.
Warning: There is no cluster found.
Warning: There is no cluster found.
HACMP: Starting cluster services at Tue Dec  3 01:10:48 2013

HACMP: Additional messages will be logged here as the cluster events are run

                        HACMP Event Preamble
----------------------------------------------------------------------------

Enqueued rg_move acquire event for resource group rg04.

Node Up Completion Event has been enqueued.
----------------------------------------------------------------------------
:check_for_site_up[+54] [[ high = high ]]
:check_for_site_up[+54] version=1.4
:check_for_site_up[+55] :check_for_site_up[+55] cl_get_path
HA_DIR=es
:check_for_site_up[+57] STATUS=0
:check_for_site_up[+59] set +u
:check_for_site_up[+61] [ ]
"/var/hacmp/log/hacmp.out" [Read only] 19847 lines, 1335877 characters
```

Note: You cannot use the vi editor or **cat** command to read or write a privileged file. You can use only the pvi editor to do so.

ha_mon role example

An example of monitoring resource group information using **clRGinfo** by a non-root user with the *ha_mon* role is shown in Example 9-14.

Example 9-14 Monitoring RS information using clRGinfo by a non-root user with the ha_mon role

```
# lsuser haMon
haMon id=209 pgrp=staff groups=staff home=/home/haMon shell=/usr/bin/ksh
login=true su=true rlogin=true telnet=true daemon=true admin=false sugroups=ALL
admgroups= tpath=nosak ttys=ALL expires=0 auth1=SYSTEM auth2=NONE umask=22
registry=LDAP SYSTEM=LDAP logintimes= loginretries=0 pwdwarntime=0
account_locked=false minage=0 maxage=0 maxexpired=-1 minalpha=0 minloweralpha=0
minupperalpha=0 minother=0 mindigit=0 minspecialchar=0 mindiff=0 maxrepeats=8
minlen=0 histexpire=0 histsize=0 pwdchecks= dictionlist= default_roles=
fsize=2097151 cpu=-1 data=262144 stack=65536 core=2097151 rss=65536 nofiles=2000
roles=ha_mon

# su - haMon
$ whoami
haMon
$ swrole ha_mon
haMon's Password:
$ rolelist -e
ha_mon
$ /usr/es/sbin/cluster/utilities/clRGinfo
-----------------------------------------------------------------------------
Group Name     State                        Node
-----------------------------------------------------------------------------
```

```
rg04          OFFLINE                      lpar0104
              ONLINE                       lpar0204
```

9.5 Customized method to achieve basic RBAC functions

Some organizations, especially those with few servers and clusters in their environments, might not have an existing LDAP server. To use some of the basic RBAC functions, for example, to enable a non-root user to run **clRGinfo**, they might not need to configure PowerHA SystemMirror federated security to take advantage of the whole set of security features. Instead, they can customize the AIX built-in RBAC to do that.

In this section, we use the **clRGinfo** example to illustrate the customization. To enable a non-root user to run **clRGinfo**, complete the following steps:

1. Check whether Enhanced RBAC is enabled by running the following command:

   ```
   lsattr -El sys0 -a enhanced_RBAC
   ```

   "True" means that it is enabled. If it is not, enable it by running this command:

   ```
   chdev -l sys0 -a enhanced_RBAC=true
   ```

2. Create a user-defined authorization hierarchy:

   ```
   mkauth dfltmsg='IBM custom' ibm
   mkauth dfltmsg='IBM custom application' ibm.app
   mkauth dfltmsg='IBM custom application execute' ibm.app.exec
   ```

3. Assume that the command is not listed in /etc/security/privcmds. If you want to find out what privileges are necessary to run the command, use **tracepriv**, as Example 9-15 shows. Otherwise, skip this step.

   Example 9-15 Using tracepriv to find the necessary privileges to run a command

   ```
   # tracepriv -ef /usr/es/sbin/cluster/utilities/clRGinfo
   -----------------------------------------------------------------------------
   Group Name     State                        Node
   -----------------------------------------------------------------------------
   rg04           ONLINE                       lpar0104
                  OFFLINE                      lpar0204

   4128894: Used privileges for /usr/es/sbin/cluster/utilities/clRGinfo:
     PV_NET_CNTL                        PV_NET_PORT
   ```

4. Add the command to the privileged command database:

   ```
   setsecattr -c accessauths=ibm.app.exec innateprivs=PV_NET_CNTL,PV_NET_PORT
   /usr/es/sbin/cluster/utilities/clRGinfo
   ```

Note: If the command being added is a shell script, you might have to add an authorization to change your effective user ID (EUID) to match the owner of the shell script by using the `euid` attribute.

For example, clsnapshot is a shell script:

```
# ls -l  /usr/es/sbin/cluster/utilities/clsnapshot
-r-x------    1 root     system       115020 Nov 08 00:18
/usr/es/sbin/cluster/utilities/clsnapshot
```

Add the command to the privileged command database by using this command:

```
# setsecattr -c euid=0 accessauths=ibm.app.exec
innateprivs=PV_DAC_GID,PV_NET_CNTL,PV_NET_PORT
/usr/es/sbin/cluster/utilities/clsnapshot
```

5. Verify that the command has been added successfully:

   ```
   lssecattr -F -c /usr/es/sbin/cluster/utilities/clRGinfo
   ```

6. Create a role that contains the authorization necessary to run the command:

   ```
   mkrole authorizations='ibm.app.exec' dfltmsg='Custom role to run PowerHA exec'
   ha_exec
   ```

7. Update the kernel security tables (KST) by using the **`setkst`** command.
8. Assign the role to a non-root user:

   ```
   chuser roles=ha_exec haExec
   ```

Now, a non-root user can run **`clRGinfo`** as the example in Example 9-16 shows.

Example 9-16 A non-root user running clRGinfo

```
# lsuser haExec
haExec id=205 pgrp=staff groups=staff home=/home/haExec shell=/usr/bin/ksh
auditclasses=general login=true su=true rlogin=true daemon=true admin=false
sugroups=ALL admgroups= tpath=nosak ttys=ALL expires=0 auth1=SYSTEM auth2=NONE
umask=22 registry=files SYSTEM=compat logintimes= loginretries=5 pwdwarntime=5
account_locked=false minage=0 maxage=52 maxexpired=8 minalpha=2 minloweralpha=0
minupperalpha=0 minother=2 mindigit=0 minspecialchar=0 mindiff=4 maxrepeats=8
minlen=8 histexpire=26 histsize=4 pwdchecks= dictionlist= default_roles=
fsize=2097151 cpu=-1 data=262144 stack=65536 core=2097151 rss=65536 nofiles=2000
roles=ha_exec
# su - haExec
$ whoami
haExec
$ swrole ha_exec
haExec's Password:
$ rolelist -e
ha_exec         Custom role to run PowerHA exec
$ /usr/es/sbin/cluster/utilities/clRGinfo
-----------------------------------------------------------------------------
Group Name     Group State                  Node
-----------------------------------------------------------------------------
ha71_rg        ONLINE                       aixtnha105
               OFFLINE                      aixtnha155
```

10

Dynamic host name change (host name takeover)

This chapter describes the dynamic host name change support in the cluster. It includes the following topics:

- Why changing the host name might be necessary
- Choosing the dynamic host name change type
- Changing the host name
- Initial setup and configuration
- Temporary host name change
- Permanent host name change
- Changing the host name in earlier PowerHA 7.1 versions
- Migrating a host name takeover environment
- PowerHA hostname change script

10.1 Why changing the host name might be necessary

In most cluster environments, there might not be a need to change the host name after the cluster is deployed. However, due to some existing applications, there might be a need to change the host name when a failover occurs. Most common middleware products do not need the host name to be changed after a failover. However, verify with the middleware provider whether the host name needs to be changed during a failover of a clustered environment.

> **Note:** If it is done incorrectly, changing the host name can lead to multiple nodes having the same host name. That could cause confusion in the TCP/IP networking in the environment.

Before looking into this solution, check with your application specialist about whether a dynamic host name is really needed. Most applications can be configured not to be host name-dependent.

Older versions of IBM Systems Director, SAP, or Oracle applications might have a host name dependency requirement and require that the host name is acquired by the backup system. For information about the latest versions and requirements of those applications, check the following websites:

http://www.ibm.com/systems/director/
http://www.sap.com
http://www.oracle.com

If a middleware product needs the host name to be changed when a failover is happening, the most common method of accomplishing this host name change is to use the IBM AIX **hostname** command in the start script for the middleware. Also, it is necessary to restore the host name to its original name when the application is being stopped in the stop script to avoid multiple nodes having the same host name accidentally.

There are two supported solutions in IBM PowerHA 7.1.3: *Temporary* and *permanent* host name changes. Which of these two solutions work for you depends on your application.

The main question here is: How does the application get the host name information? For details, see 10.5, “Temporary host name change” on page 370 and 10.6, “Permanent host name change” on page 372.

If you are using a PowerHA version before 7.1.3, check whether the solution described in 10.7, “Changing the host name in earlier PowerHA 7.1 versions” on page 375, might be an option for you.

10.2 Choosing the dynamic host name change type

AIX supports the two types of host name modification that are described in this section. AIX stores the host name information in two important locations:

- First, in the Object Data Manager (ODM), which is the permanent or persistent information and is used by AIX during boot to set the host name of the node.

- Second, in the host name location, which is in the AIX kernel memory. During boot, this variable is set to the same value as in the ODM. This value can be modified temporarily and is valid for the life of the AIX LPAR. After the LPAR reboots, the host name is reset back to the value from the ODM.

In summary, AIX stores two kinds of host names:

- *Permanent host name:* Stored persistently in the ODM
- *Temporary host name:* Kernel memory variable

AIX provides various commands and APIs to work with these two types of host names:

- Interfaces that set or get the permanent host name. This ODM attribute is read directly using the ODM-related primitives:

  ```
  lsattr -El inet0 | grep hostname
  odmget -q "attribute=hostname" CuAt
  ```

  The host name can also be permanently changed by using the SMIT panel.
- Interfaces that set or get the host name temporarily:

  ```
  hostname
  uname -n or uname -a
  hostid
  The function gethostname()
  ```

If your application is using scripts to set or get the host name temporarily, that is easy to determine by searching for one of the options listed above. If your application is based on binaries, the AIX gethostname() function is probably used. But because you have only a binary file, you must test it to find out.

10.3 Changing the host name

This section describes the different options to change the host name of an AIX system. These options are described from an AIX point of view only. What it means to your application, Cluster Aware AIX (CAA), or PowerHA are discussed in 10.5, “Temporary host name change” on page 370 and 10.6, “Permanent host name change” on page 372.

There are two ways to change the host name of an AIX system:

- By using the command line. See 10.3.1, “Using the command line to change the host name” on page 361. This method gives you the most flexibility.
- By using the System Management Interface Tool (SMIT). See 10.3.2, “Using SMIT to change the host name information” on page 365.

Be sure to read 10.3.3, “Cluster Aware AIX (CAA) dependencies” on page 367 for details about the effects of the different commands on the CAA.

10.3.1 Using the command line to change the host name

As listed in section 10.2, “Choosing the dynamic host name change type” on page 360, there are several ways to get the host name by using the command line. Similar commands can be used to change it also. The following sections contain a list of these commands and some details about them.

hostname command

If only the **hostname** command is used, the type of dynamic host name that you need is *temporary host name change*. For details on how to set this up, see 10.5, “Temporary host name change” on page 370.

Using the **hostname <name>** command changes the host name in the running environment. The *hostname* kernel variable will be changed to the new name. Therefore, if you use host name without any options or arguments, you get the name that you specified for the <name> variable.

The AIX gethostname() function is also reading from the *hostname* kernel variable. Therefore, it also returns the name that you specified under <name>.

None of the other values considered to list the host name change. Example 10-1 shows this behavior. The first part shows the output of the different commands before using the **hostname <name>** command, and the second half shows the output after the change.

Example 10-1 hostname command

```
root@asterix(/)# hostname
asterix
root@asterix(/)# uname -n
asterix
root@asterix(/)# lsattr -El inet0 | grep hostname
hostname      asterix                       Host Name                        True
root@asterix(/)# hostid
0xac1e77cd
root@asterix(/)# host $(hostid)
asterix is 172.30.119.205,  Aliases:   pokbc.lpar0103
root@asterix(/)#
root@asterix(/)# hostname fred
root@asterix(/)# hostname
fred
root@asterix(/)# uname -n
asterix
root@asterix(/)# lsattr -El inet0 | grep hostname
hostname      asterix                       Host Name                        True
root@asterix(/)# hostid
0xac1e77cd
root@asterix(/)# host $(hostid)
asterix is 172.30.119.205,  Aliases:   pokbc.lpar0103
root@asterix(/)#
```

uname command

If the **uname -n** or **uname -a** command is used, then *temporary hostname change* is the better choice. For details in how to set this up, see section 10.5, “Temporary host name change” on page 370.

Using the **uname -S <name>** command changes the uname information in the running environment. The kernel variable *utsname* is changed to the new name. Therefore, if you use **uname -n** or **-a**, you get the name that you specified for the <name> variable.

None of the other values considered to list the host name change. Example 10-2 on page 363 shows this behavior. The first part shows the output of the different commands before using the **uname -S <name>** command, and the second part shows the output after the change.

Example 10-2 uname command

```
root@asterix(/)# hostname
asterix
root@asterix(/)# uname -n
asterix
root@asterix(/)# lsattr -El inet0 | grep hostname
hostname      asterix                       Host Name                        True
root@asterix(/)# hostid
0xac1e77cd
root@asterix(/)# host $(hostid)
asterix is 172.30.119.205,  Aliases:   pokbc.lpar0103
root@asterix(/)#
root@asterix(/)# uname -S fred
root@asterix(/)# hostname
asterix
root@asterix(/)# uname -n
fred
root@asterix(/)# lsattr -El inet0 | grep hostname
hostname      asterix                       Host Name                        True
root@asterix(/)# hostid
0xac1e77cd
root@asterix(/)# host $(hostid)
asterix is 172.30.119.205,  Aliases:   pokbc.lpar0103
root@asterix(/)#
```

hostid command

The **hostid** command returns a hex value of the IP label that is normally associated with the host name.

If the **hostid** command is used, then *temporary hostname change* is appropriate. For details on how to set this up, see 10.5, “Temporary host name change” on page 370.

Using the **hostid <name>** command changes the hostid information in the running environment. Keep in mind that the name that you used for `<name>` must be a resolvable name. You can use the IP address instead. If you use the **hostid** command without any options or arguments, you get the hex value for the specified information under `<name>`. To get readable information, you can use either the **host** or **ping** command:

```
host $(hostid)
```

or

```
ping $(hostid)
```

None of the other values considered to list the host name change. Example 10-3 on page 364 shows this behavior. The first part shows the output of the different commands before using the **hostid <name>** command, and the second half shows the output after the change.

Example 10-3 hostid command

```
root@asterix(/)# hostname
asterix
root@asterix(/)# uname -n
asterix
root@asterix(/)# lsattr -El inet0 | grep hostname
hostname      asterix                       Host Name                      True
root@asterix(/)# hostid
0xac1e77cd
root@asterix(/)# host $(hostid)
asterix is 172.30.119.205,  Aliases:   pokbc.lpar0103
root@asterix(/)#
root@asterix(/)# hostid paris
root@asterix(/)# hostname
asterix
root@asterix(/)# uname -n
asterix
root@asterix(/)# lsattr -El inet0 | grep hostname
hostname      asterix                       Host Name                      True
root@asterix(/)# hostid
0xac1e77ef
root@asterix(/)# host $(hostid)
paris is 172.30.119.239,  Aliases:   test-svc1
root@asterix(/)#
```

odmget and lsattr commands

If the **odmget** or **lsattr -El inet0** command is used, then *permanent hostname change* is appropriate. For details on how to set this up, see section 10.6, “Permanent host name change” on page 372.

Using the **chdev -l inet0 -a hostname=<name>** command changes two components. Like using the **hostname** command, it changes the *hostname* kernel variable. It also changes the host name information in the CuAt ODM class. Therefore, if you use the **lsattr -El inet0 | grep hostname**, you get the name you specified for <name>. In this case, you get the same result when you use the **hostname** command.

Important: Using the **chdev** command makes the change persistent across a reboot, so this change can create problems, potentially.

None of the other values considered to list the host name change. Example 10-4 on page 365 show this behavior. The first part shows the output of the different commands before using the **chdev -l inet0 -a hostname=<name>** command, and the second half shows the output we get after the change.

Example 10-4 lsattr command

```
root@asterix(/)# hostname
asterix
root@asterix(/)# uname -n
asterix
root@asterix(/)# lsattr -El inet0 | grep hostname
hostname      asterix                        Host Name                        True
root@asterix(/)# hostid
0xac1e77cd
root@asterix(/)# host $(hostid)
asterix is 172.30.119.205,  Aliases:   pokbc.lpar0103
root@asterix(/)#
root@asterix(/)# chdev -l inet0 -a hostname=london
root@asterix(/)# hostname
london
root@asterix(/)# uname -n
asterix
root@asterix(/)# lsattr -El inet0 | grep hostname
hostname      london                         Host Name                        True
root@asterix(/)# hostid
0xac1e77cd
root@asterix(/)# host $(hostid)
asterix is 172.30.119.205,  Aliases:   pokbc.lpar0103
root@asterix(/)#
```

gethostname function

The gethostname() C function gets its value from the running *hostname* kernel variable. Therefore, if you use the **hostname** or the **chdev** commands to change the host name, the function returns the new host name.

> **Note:** The examples for this section are illustrated in "hostname command" on page 362, and "odmget and lsattr commands" on page 364.

10.3.2 Using SMIT to change the host name information

Using the SMIT menu is a good idea only if you need to make a permanent change. If this your aim, the recommended way is to select **smitty mkhostname** → **Communications Applications and Services** → **TCP/IP** → **Further Configuration** → **Hostname** → **Set the Hostname** (see Figure 10-1 on page 366).

```
                                 Set Hostname

                     Please refer to Help for information
                  concerning hostname / INTERNET address mapping

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

                                                        [Entry Fields]
* HOSTNAME (symbolic name of your machine)           [obelix]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 10-1 smitty mkhostname

It is important to keep in mind that this step makes several changes to your system. When you run your changes, the system performs the **chdev** and **hostid** commands, so most of the host name-related information gets updated in one action. The only exception is the uname information. To get the uname-related *utsname* kernel variable updated also, you have two options: You can reboot the system or use the **uname -S $(hostname)** command.

Example 10-5 shows the information from our test systems. For this example, we used the value listed in Figure 10-1. The first part shows the output of the different commands before using SMIT, and the second half shows the output after the change.

Example 10-5 smitty mkhostname

```
root@asterix(/)# hostname
asterix
root@asterix(/)# uname -n
asterix
root@asterix(/)# lsattr -El inet0 | grep hostname
hostname      asterix                       Host Name                       True
root@asterix(/)# hostid
0xac1e77cd
root@asterix(/)# host $(hostid)
asterix is 172.30.119.205,  Aliases:   pokbc.lpar0103
root@asterix(/)#
root@asterix(/)# smitty mkhostname
...
...
root@asterix(/)# hostname
obelix
root@asterix(/)# uname -n
asterix
root@asterix(/)# lsattr -El inet0 | grep hostname
hostname      obelix                        Host Name                       True
root@asterix(/)# hostid
0xac1e77e3
root@asterix(/)# host $(hostid)
obelix is 172.30.119.227,  Aliases:   pokbc.lpar0203
root@asterix(/)#
```

Theoretically, there is another path to get the host name defined: **smitty mktcpip** or “Minimum Configuration and Startup.” However, you should never use this path on a configured system, because it does a lot more that just defining the host name. In the worst case, it can create severe problems in your existing setup.

> **Attention:** Never use **smitty mktcpip** on an existing environment only to change the host name.

10.3.3 Cluster Aware AIX (CAA) dependencies

CAA starts with the output of the **hostname** command to check whether the host name can be used or not. CAA does several additional checks based on this information.

From a host name change point of view, you should know about the commands described in 10.3.1, “Using the command line to change the host name” on page 361, because they affect the CAA. The following list gives you a brief summary:

uname -S <name> The uname information is ignored by CAA.

hostid <name> The hostid information is ignored by CAA.

hostname <name> This is the primary information used by CAA during setup. If it changes, it is ignored if you use the default setting. However, if you use **clmgr** to change the TEMP_HOSTNAME variable from disallow (which is the default) to allow, the CAA changes the “CAA Node Name” to the new host name.

chdev -l inet0 -a hostname=<name> In this case, you are initiating a permanent host name takeover, so you need to synchronize the cluster.

10.4 Initial setup and configuration

To set up your virtual systems to be able to make use of the dynamic host name change, you must complete the steps in the subsections that follow. The descriptions are based on the assumption that you already have the operating system and PowerHA installed.

Depending on your environment, there are different sequences, which are explained in these sections:

- “New system setup”
- “Adding and configuring PowerHA in an existing environment” on page 369

10.4.1 New system setup

A brief summary of the key steps follows. If PowerHA is new to you, you can find a detailed description of general installation steps in Chapter 2, “Basic concepts” on page 9.

> **Note:** Keep in mind that some of the tasks listed here must be done on all cluster nodes.

Before starting with these steps, make sure that you have the scripts that manage your host name takeover available. An example of what we used is listed in 10.9, “PowerHA hostname change script” on page 378.

1. Install AIX and the PowerHA components that you need for your environment.
2. Configure your AIX environment by defining all necessary variables, user IDs, group IDs, TCP/IP settings, and disks.
3. Verify that you have all necessary PowerHA TCP/IP addresses defined in your `/etc/hosts` file, and make sure that you have all shared volume groups (VGs) known to all cluster nodes.
4. Add your boot IP addresses to the `/etc/cluster/rhosts` file. That file must have the same content on all cluster nodes.
5. Configure all your boot or base addresses in all your cluster nodes.
6. Check that the host name is equal to one of your IP-Labels used for the boot address(es).
7. If you already know your storage configuration details, you can do this at this step. If not just continue with the next step.
8. Start configuring PowerHA:
   a. Configure the Power Cluster name and nodes.
   b. Configure your repository disk.
   c. Synchronize your cluster.
   d. Define the application script to set the host name.
   e. Add your service IP address.
   f. Configure a resource group with your service IP only.
   g. Synchronize your cluster.
   h. Start only one cluster node, for instance your primary note.
   This makes the host name and your service address available to install the application.
9. Configure your VGs and file systems and mount them if not already done as part of step 7.
10. Install your applications. Depending on your application, you might need to varyon and mount your application-related VGs first.
11. Stop your application and stop the resource group or move the resource group to your next system.
12. Activate the resource group on your backup system (still IP and maybe VG only) if not already done as part of step 11.
13. Install you application on the backup system.
14. Stop your application and your resource group.
15. Add your application start/stop scripts to your resource group. Check that your resource group now contains all components that are necessary for starting your application.
16. Synchronize you PowerHA cluster.
17. Continue with 10.5, “Temporary host name change” on page 370 or 10.6, “Permanent host name change” on page 372 and start testing.

10.4.2 Adding and configuring PowerHA in an existing environment

Keep in mind this is a disruptive process. So before you start making your existing application highly available, check the maintenance window you have is enough to perform all needed tasks.

As in the section above, we list a brief summary of the key steps:

1. If you have not already done so, install PowerHA.
2. Check that all needed PowerHA TCP/IP addresses are defined in your `/etc/hosts` file.
3. Add your boot IP addresses, persistent IP addresses, and your service IP address to the `/etc/cluster/rhosts` file. The /etc/cluster/rhosts file must have the same content on all cluster nodes.
4. Stop your application.
5. Change your existing host name to be equal to the new boot IP label (use **`smitty`** or **`chdev`**), and change your interface to reflect the new boot address.

   In our test environment, we used **`smitty mkhostname`** and **`smitty chinet`**. To be on the safe side, we also used **`uname -S <name>`**.

   > **Important:** *Do not* use **`smitty mktcpip`** here.

6. Now you might must migrate data from your local VG to a shared VG.

   If moving data from the local VG to the shared VG is necessary, the safest way is to use the following commands:

   ```
   cd <source_dir>
   find . | backup -if - | (cd <target_dir>; restore -xdqf -)
   umount or delete <source_dir>
   ```

7. Start to configure PowerHA.
   a. Configure the Power Cluster name and nodes.
   b. Configure your repository disk.
   c. Synchronize your cluster.
   d. Define the application script to set the host name.
   e. Add your service IP address.
   f. Configure a resource group with your service IP only. Optionally, you can add the shared VG.
   g. Synchronize your cluster.
8. Test whether your application is still able to run on your primary system. If yes, continue with the next step. Otherwise, continue with step 13.
9. Activate your resource group on your primary system.
10. Test whether your application works.
11. Stop your application.
12. Stop or move your resource group to the backup system. If you move the resource group, you can continue with step 14 on page 370.
13. Activate the resource group on the backup system. This will make the host name and your service address available to install the application on the backup system.

14. Install your application.
15. Stop your application and your resource group.
16. Add your application start/stop scripts to your resource group. Make sure that your resource group now contains all components that are necessary to start your application.
17. Synchronize your PowerHA cluster.
18. Continue with either “Temporary host name change” or 10.6, “Permanent host name change” on page 372, and then start testing.

10.5 Temporary host name change

When your application requires a host name takeover, the temporary host name change should be the preferred one. It is rare that an application checks the content of the AIX `CuAt` ODM class.

If you are already using host name takeover with an earlier PowerHA version and you are planning for migration to PowerHA 7.1.3, you must check your existing host name change scripts first. As explained in more detail in section 10.8, “Migrating a host name takeover environment” on page 376, these scripts should *not* contain a **`chdev -l inet0`** command.

To get the temporary host name takeover to work, follow the description in section 10.4, “Initial setup and configuration” on page 367. Also check that the start/stop scripts handle the change of the host name.

> **Note:** A goal of the temporary host name takeover is that this change does not persist across a reboot.

The following section shows example output from our test system. Before starting the resource group, check some of the system settings. Example 10-6 shows that our initial host name is *asterix* and that the CAA node names are *asterix* and *obelix*.

Example 10-6 Check hostname information before starting the resource group

```
root@asterix(/)# hostname; uname -n ; host $(hostid); lsattr -El inet0 | grep host
asterix
asterix
asterix is 129.40.119.205,  Aliases:   pokbc.lpar0103
hostname      asterix                            Host Name                     True
root@asterix(/)#lscluster -m
Calling node query for all nodes...
Node query number of nodes examined: 2

        Node name: asterix
        Cluster shorthand id for node: 1
        UUID for node: 400aa068-5b4e-11e3-8b2f-2a6f38485e04
        State of node: UP  NODE_LOCAL
        Smoothed rtt to node: 0
        Mean Deviation in network rtt to node: 0
        Number of clusters node is a member in: 1
        CLUSTER NAME       SHID         UUID
        france_cluster     0            400dcee6-5b4e-11e3-8b2f-2a6f38485e04
        SITE NAME          SHID         UUID
        LOCAL              1            51735173-5173-5173-5173-517351735173
```

```
        Points of contact for node: 0

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

        Node name: obelix
        Cluster shorthand id for node: 2
        UUID for node: 400aa0b8-5b4e-11e3-8b2f-2a6f38485e04
        State of node: UP
        Smoothed rtt to node: 23
        Mean Deviation in network rtt to node: 21
        Number of clusters node is a member in: 1
        CLUSTER NAME       SHID         UUID
        france_cluster     0            400dcee6-5b4e-11e3-8b2f-2a6f38485e04
        SITE NAME          SHID         UUID
        LOCAL              1            51735173-5173-5173-5173-517351735173

        Points of contact for node: 1
        ------------------------------------------------------------------------
        Interface     State  Protocol    Status     SRC_IP->DST_IP
        ------------------------------------------------------------------------
        tcpsock->02   UP     IPv4         none   129.40.119.205->129.40.119.227
root@asterix(/)#
```

Next, start PowerHA or the resource group. Wait until the cluster is back in a stable state. Then, use the commands shown in Example 10-6 on page 370.

Example 10-7 shows the output after that. As expected, the hostname changes when you use one of the following commands: **hostname**, **uname -n**, or **host $(hostid)**. The information in CAA and CuAt does not change.

Example 10-7 Check hostname information after resource group start

```
root@asterix(/)# hostname; uname -n ; host $(hostid); lsattr -El inet0 | grep host
paris
paris
paris is 129.40.119.239,  Aliases:   bb-svc1
hostname      asterix                          Host Name
True
root@asterix(/)# lscluster -m
Calling node query for all nodes...
Node query number of nodes examined: 2

        Node name: asterix
        Cluster shorthand id for node: 1
        UUID for node: 400aa068-5b4e-11e3-8b2f-2a6f38485e04
        State of node: UP  NODE_LOCAL
        Smoothed rtt to node: 0
        Mean Deviation in network rtt to node: 0
        Number of clusters node is a member in: 1
        CLUSTER NAME       SHID         UUID
        france_cluster     0            400dcee6-5b4e-11e3-8b2f-2a6f38485e04
        SITE NAME          SHID         UUID
        LOCAL              1            51735173-5173-5173-5173-517351735173

        Points of contact for node: 0
```

```
---------------------------------------------------------------------------

        Node name: obelix
        Cluster shorthand id for node: 2
        UUID for node: 400aa0b8-5b4e-11e3-8b2f-2a6f38485e04
        State of node: UP
        Smoothed rtt to node: 7
        Mean Deviation in network rtt to node: 7
        Number of clusters node is a member in: 1
        CLUSTER NAME       SHID         UUID
        france_cluster     0            400dcee6-5b4e-11e3-8b2f-2a6f38485e04
        SITE NAME          SHID         UUID
        LOCAL              1            51735173-5173-5173-5173-517351735173

        Points of contact for node: 1
        ------------------------------------------------------------------------
        Interface     State  Protocol    Status     SRC_IP->DST_IP
        ------------------------------------------------------------------------
        tcpsock->02   UP     IPv4         none   129.40.119.205->129.40.119.227
root@asterix(/)#
```

Now, when you move the resource group to the backup system, you see that the system *asterix* gets back its original host name. Also, the host name of the backup system is now showing the host name *paris* rather than *obelix*.

10.6 Permanent host name change

In this section, we demonstrate the permanent host name change in two scenarios. In both scenarios, the IP address associated with the host name is used as the boot IP address. In the first scenario, we change the host name but not its IP address. In the second scenario, we change both the host name and its IP address.

10.6.1 Scenario 1: Host name changes but IP address does not

1. Stop cluster services on all nodes by using the **Bring Resource Group Offline** option.
2. Change the `/etc/hosts` file for each node in the cluster to the new host name. If your environment is using a DNS, you must update the DNS with the new host name.
3. Change node 1 host name with the #**`smitty mkhostname`** command.
4. Verify and synchronize the cluster configuration from node 1. This updates the COMMUNICATION_PATH of node 1 on both nodes.

> **Note:** This action restores the previous host name in the `/etc/hosts` directory and appends it into the new entry as this syntax shows:
>
> `x.x.x.x <new host name> <old host name>`

5. Change node 2 host name with the #**`smitty mkhostname`** command.
6. Verify and synchronize the cluster configuration from node 2. This updates the COMMUNICATION_PATH of node 2 on both nodes.

7. On node 1, update the boot IP label, and edit `tmp1` with the new host name in the ip_label field.

```
#odmget HACMPadapter > /tmp/tmp1
edit tmp1
#odmdelete -o HACMPadapter
odmadd /tmp/tmp1
```

8. Change the `/etc/hosts` file on both nodes to remove the old host name that was added in step 4 on page 372.
9. *Optional:* To change a PowerHA node name, execute #**`smitty cm_manage_nodes`**. Update the node name (be sure that you do not alter the communication path), and press Enter.
10. Verify and synchronize the cluster configuration from node 1.
11. Start cluster services.

10.6.2 Scenario 2: Both the host name and IP address change

1. Stop cluster services on all nodes by using the Bring Resource Group Offline option.
2. Add the new host name entries into the `/etc/hosts` file for each node in the cluster. If your environment is using a DNS, you must update the DNS with the new host name.
3. Bring up the new node 1 IP address on node 1 by using an IP alias:

```
# ifconfig en# <new IP address> netmask <new netmask> alias up
```

4. Change node 1 host name with the #**`smitty mkhostname`** command.
5. Verify and synchronize the cluster configuration from node 1. This updates the COMMUNICATION_PATH of node 1 on both nodes.
6. Bring up the new node 2 IP address on node 2 using the IP alias:

```
# ifconfig en# <new IP address> netmask <new netmask> alias up
```

7. Change node 2 host name with # **`smitty mkhostname`**.
8. Verify and synchronize the cluster configuration from node 2. This updates the COMMUNICATION_PATH of node 2 on both nodes.
9. On node 1, update the boot IP label and boot IP address:

```
# odmget HACMPadapter > /tmp/tmp1
```

   Edit `tmp1` with the new host name in the ip_label field and the new IP address in the corresponding identifier field:

```
# odmdelete -o HACMPadapter
# odmadd /tmp/tmp1
```

10. Use **`smitty chinet`** to change the boot IP to the new IP address on both nodes, and remove old host name entries from `/etc/hosts`.

> **Note:** Issue the **`smitty chinet`** command from the console to avoid losing the connection if you are logged in through the old IP address.

11. *Optional:* To change the node name in PowerHA, complete the following steps on one of the cluster nodes (assuming the same node as the one in step 9).
    a. Update the new node name with **`smitty cm_manage_nodes`**.
    b. Update only the new node name. Do not select the communication path again.

12. Update /etc/cluster/rhosts to the new boot IP addresses. Then stop and restart clcomd:

```
# stopsrc -s clcomd
# startsrc -s clcomd
```

13. Verify and synchronize the cluster configuration from node 1.
14. Start cluster services.

> **Note:** To minimize downtime, the previous steps can be tuned without stopping the cluster service, but it still requires two short downtime periods during the resource group movement. Follow the actions in step 2 on page 235 through step 8 on page 235, and then follow these steps:

15. Move the resource group from node 1 to node 2.
16. On node 1, update its boot IP label and boot IP address, and then edit `tmp1` with the new host name in the ip_label field and the new IP address in the corresponding identifier field:

```
# odmget HACMPadapter > /tmp/tmp1
# odmdelete -o HACMPadapter
# odmadd /tmp/tmp1
```

17. Change the boot IP to the new IP address on node 1 by using **smitty chinet**.
18. Verify and synchronize the cluster configuration from node 1.
19. Move the resource group from node 2 to node 1.
20. On node 2, update its boot IP label and boot IP address, and edit tmp1 with the new host name in the ip_label field, and the new IP address in the corresponding identifier field:

```
# odmget HACMPadapter > /tmp/tmp1
# odmdelete -o HACMPadapter
# odmadd /tmp/tmp1
```

21. Change the boot IP to the new IP address on node 2 with **smitty chinet**.
22. Verify and synchronize the cluster configuration from node 2.
23. Remove the old host name entries from `/etc/hosts` and the DNS.
24. Update `/etc/cluster/rhosts` to the new boot IP addresses, and then stop and restart clcomd:

```
# stopsrc -s clcomd
# startsrc -s clcomd
```

> **Note:** It is not allowed to change the node name in PowerHA when the node is active. However, you can change it later after bringing down the cluster service. To change the node name in PowerHA, complete the following steps on one of the cluster nodes:
>
> - Update the new node name with **smitty cm_manage_nodes**.
>
>   Update only the new node name, do not select communication path again at this step.
> - Then, verify and synchronize the cluster configuration from the same node.

If the application start script has a command such as **#chdev -l inet0 -a hostname=<service IP label>** after the application is started on a node, you must run the cluster verification and synchronization from the same node. This updates the new COMMUNICATION_PATH of that node to all of the nodes in the cluster. If the application starts in this way, its stop script usually contains a command such as **#chdev -l inet0 -a hostname=<old host name>**. This is to change the source node host name back to the original.

When there is a planned resource group movement, the resource group is brought down on the source node, and then it is brought up on the destination node. The stop script changes the source node host name back, and the start script changes the destination node host name to the service IP label. At this time, the destination node still remembers the source node from its previous communication path as the service IP address. But the IP address is now on the destination node, so it is not possible to synchronize the cluster configuration from the destination node. Synchronize it from the source node. Similarly, when you move the resource group back to the source node, you must synchronize the cluster from the destination node.

If the stop script has been run when there is an unplanned resource group movement, such as the result of a source node network failure, the host name change behavior and actions are similar to the planned resource group movement. However, if it is a node failure, the stop script is not run, so the failure node host name does not change back to its original, but remain as the service IP label after restart. In this case, after the failed node is up again, you must manually change its host name by using **smitty mkhostname**. However, you are not able to synchronize the cluster configuration from the failure node because PowerHA SystemMirror does not allow synchronization from an inactive node to an active node. You must manually update the COMMUNICATION_PATH in the ODM of the node that the application is now running on. The commands are shown in Example 10-8.

Example 10-8 Manually update the COMMUNICATION_PATH in the ODM

```
# odmget HACMPnode > /tmp/tmp1, edit tmp1 with the new host name in
COMMUNICATION_PATH value field. # odmdelete -o HACMPnode; odmadd /tmp/tmp1.
```

Then, you can synchronize the cluster configuration from the node to the failed node and start the cluster service on the failed node again.

When you stop the cluster service on the source node with the option to move the resource group, it behaves like the planned resource group movement. So you must synchronize the cluster configuration from the source node after the movement.

You can then start the cluster service on the source node again. Then, move the resource group back according to the planned resource group movement.

10.7 Changing the host name in earlier PowerHA 7.1 versions

Dynamic host name change is not supported in PowerHA SystemMirror 7.1.2 or earlier versions.

> **Note:** Generally, after the cluster is configured, you should not need to change the host name of any cluster nodes.

To change the host name of a cluster node, you must first remove the Cluster Aware AIX (CAA) cluster definition, update PowerHA SystemMirror and the AIX operating system configurations, and then synchronize the changes to re-create the CAA cluster with the new host name.

To change the host name for a cluster node in PowerHA SystemMirror 7.1.2 or earlier versions, complete the following steps:

1. Stop the cluster services on all nodes by using the Bring Resource Group Offline option.

2. To remove the CAA cluster, complete the following steps on all nodes:

   a. Get the name of the CAA cluster:

   ```
   # lscluster -i | grep Name
   ```

   b. Get the disk of the repository disk:

   ```
   # lspv | grep caavg_private
   # clusterconf -ru <repository disk>
   ```

   c. CAA_FORCE_ENABLED=1 ODMDIR=/etc/objrepos /usr/sbin/rmcluster -f -n <CAA cluster name> -v

   d. CAA_FORCE_ENABLED=1 ODMDIR=/etc/objrepos /usr/sbin/rmcluster -f -r <repository disk> -v

   e. Reboot the node to clean up the CAA repository information.

3. To update the AIX operating system configuration, complete the following steps on all the nodes with the new host name:

   a. Change the `/etc/hosts` file for each node in the cluster with the new host name. If your environment is using a DNS, you must update the DNS with the new host name.

   b. Change the `/etc/cluster/rhosts` file on all cluster nodes.

   c. Run **smitty mktcpip** to change the host name and IP address.

   d. Stop and restart clcomd:

   ```
   # stopsrc -s clcomd; startsrc -s clcomd
   ```

4. To update the PowerHA SystemMirror configuration, complete the following steps on one of the cluster nodes:

   a. Update the communication path with **smitty cm_manage_nodes**. Select only the new communication path. Do not update the new node name at this step.

   b. Update the boot IP label, and then edit `tmp1` with the new host name in the ip_label field and the new IP address in the corresponding identifier field:

   ```
   # odmget HACMPadapter > /tmp/tmp1
   # odmdelete -o HACMPadapter
   # odmadd /tmp/tmp1
   ```

   c. Discover the network interfaces and disks:

   ```
   # smitty cm_cluster_nodes_networks
   ```

   d. Verify and synchronize the cluster configuration. This process creates the CAA cluster configuration with the updated host name.

5. *Optional:* To change the node name in PowerHA, complete the following steps on one of the cluster nodes:

   a. Update the new node name using **smitty cm_manage_nodes**. Update only the new node name, do not select the communication path again at this step.

   b. Verify and synchronize the cluster configuration.

6. Start the cluster services.

10.8 Migrating a host name takeover environment

In this section, we cover only the migration considerations related to the host name takeover. For information about migration in general, see Chapter 4, “Migration” on page 49.

The main question here is: Does your script use the **chdev** command?

- If the answer is yes, continue reading this section.
- If the answer is no, great, you can make use of the temporary host name takeover. Now check for other migration dependencies.

Now that you know that your scripts are using the **chdev** command, as shown in Example 10-9, you need to test whether the **chdev** command is needed.

Example 10-9 Existing hostname takeover script

```
case $SysName in
     alpha) echo "changing hostname to alpha ..."
               chdev -l inet0 -a hostname=alpha
               /usr/sbin/hostid `hostname`
               /bin/uname -S`hostname|sed 's/\..*$//'`
               # Compacts the ODM for the printer menues and rename links
               /usr/lib/lpd/pio/etc/piodmgr -h
               ;;
     beta) echo "changing hostname to beta ..."
               chdev -l inet0 -a hostname=beta
               /usr/sbin/hostid `hostname`
               /bin/uname -S`hostname|sed 's/\..*$//'`
               # Compacts the ODM for the printer menues and rename links
               /usr/lib/lpd/pio/etc/piodmgr -h
               ;;
esac
```

It is rare that an application checks for the content of the AIX `CuAT` ODM class. Therefore, in most cases it is not necessary to use the **chdev** command.

If you have a test environment or a wide maintenance window, an easy way to test it is by replacing the command **chdev** in our example with **hostname <name>**. Example 10-10 shows the change that we did in comparison to the part shown in Example 10-9.

Example 10-10 Modified hostname takeover script

```
case $SysName in
     alpha) echo "changing hostname to alpha ..."
              # chdev -l inet0 -a hostname=alpha
               hostname alpha
               /usr/sbin/hostid `hostname`
               /bin/uname -S`hostname|sed 's/\..*$//'`
               # Compacts the ODM for the printer menues and rename links
               /usr/lib/lpd/pio/etc/piodmgr -h
               ;;
     beta) echo "changing hostname to beta ..."
              # chdev -l inet0 -a hostname=beta
               hostname alpha
               /usr/sbin/hostid `hostname`
               /bin/uname -S`hostname|sed 's/\..*$//'`
               # Compacts the ODM for the printer menues and rename links
               /usr/lib/lpd/pio/etc/piodmgr -h
               ;;
esac
```

If your application still works with the modification shown in Example 10-10 on page 377, you can use the temporary host name takeover.

Now, check for other migration dependencies.

If your application does not work, you have one of the rare cases, so you must use the permanent host name takeover option.

10.9 PowerHA hostname change script

Example 10-11 shows the script for changing the host name. Appendix B shows PowerHA-related monitoring scripts that we used while writing this book.

Example 10-11 Change hostname script

```
#!/usr/bin/ksh
########################################################################
#
# Script to manage hostname take over
#
# this script expects one argument which is start or stop
########################################################################

#VERBOSE_LOGGING=high
[[ "$VERBOSE_LOGGING" == "high" ]] && set -x

# Variables

Task="$1"
SystemA="asterix"
SystemB="obelix"
ServiceHostname="paris"
PathUtils="/usr/es/sbin/cluster/utilities"
ActualHAnodename=$(${PathUtils}/get_local_nodename)

case $Task in
  start)        hostname $ServiceHostname;
                hostid $ServiceHostname;
                uname -S $ServiceHostname;
                RC=0;;
  stop)         hostname $ActualHAnodename;
                hostid $ActualHAnodename;
                uname -S $ActualHAnodename;
                RC=0;;
  *)            echo "Unknown Argument used";
                RC=1;;
esac

exit $RC
```

11

PowerHA cluster monitoring

The following sections in this chapter describe various approaches to monitoring the status of an IBM PowerHA cluster:

- Obtaining the cluster status
- Custom monitoring
- PowerHA cluster log monitoring
- PowerHA cluster SNMP trap monitoring
- SNMPv1 daemon support for PowerHA trap monitoring

11.1 Obtaining the cluster status

In a clustered environment, it is critical that you can gain timely and accurate status information about the cluster topology and application resources. It is also critical that application monitors are configured for each application that is to be made highly available in the cluster[1]. Without application monitors, PowerHA has no mechanism to determine whether your applications are actually up, available, and performing as you would expect.

PowerHA provides commands such as **`cldump`** and **`clstat`** for monitoring the status of the cluster. There are also IBM Tivoli file sets that provide support for existing version 5 monitoring environments. There is no *specific* cluster monitoring function for Tivoli Monitoring version 6 or other OEM enterprise monitoring products. For more information, see 11.3.1, "IBM Tivoli Monitoring agent for UNIX logs" on page 390.

The SNMP protocol is the crux of obtaining the status of the cluster. The SNMP protocol is used by network management software and systems for monitoring network applications and devices for conditions that warrant administrative attention. The SNMP protocol is composed of a database and a set of data objects. The set of data objects forms a Management Information Base (MIB). The standard SNMP agent is the *snmpd* daemon. A SMUX (SNMP Multiplexing protocol) subagent allows vendors to add product-specific MIB information.

The *clstrmgr* daemon in PowerHA acts as a SMUX subagent. The SMUX peer function, which is in clstrmgrES, maintains cluster status information for the PowerHA MIB. When the clstrmgrES starts, it registers with the SNMP daemon, snmpd, and continually updates the MIB with cluster status in real time. PowerHA implements a private MIB branch that is maintained by a SMUX peer subagent to SNMP that is contained in the clstrmgrES daemon, as shown in Figure 11-1.

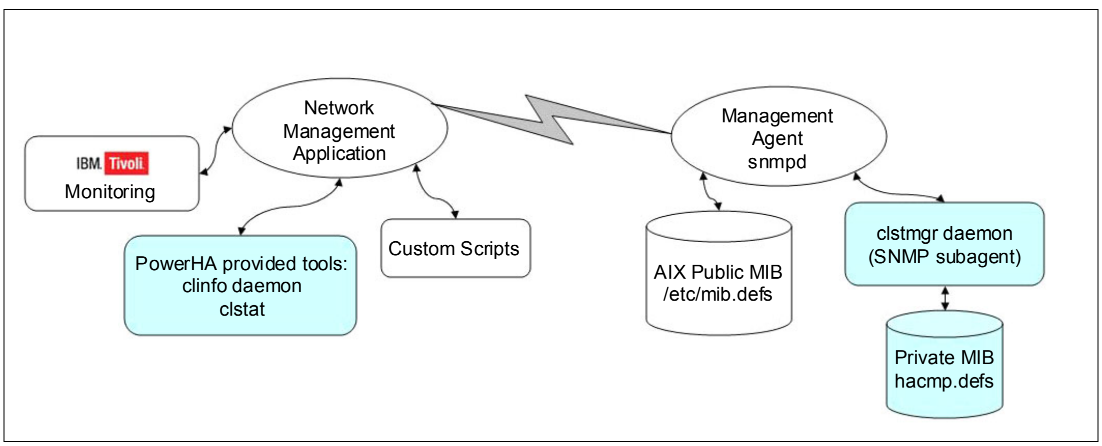


Figure 11-1 PowerHA private Management Information Base

[1] Application monitoring is a feature of PowerHA which aides the cluster in determining whether the application is alive and well. Further information about application monitoring is beyond the scope of this chapter,

PowerHA participates under the IBM Enterprise SNMP MIB (Figure 11-2):

ISO (1) → Identified Organization (3) → Department of Defense (6) → Internet (1) → Private (4) → Enterprise (1) → IBM (2) → IBM Agents (3) → AIX (1) → aixRISC6000 (2) → risc6000agents (1) → risc6000clsmuxpd (5)

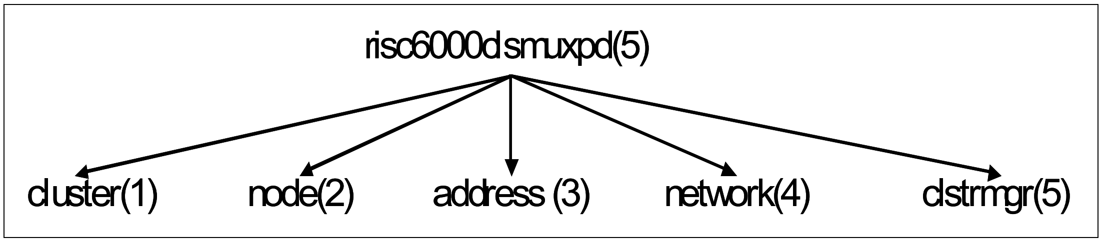


Figure 11-2 PowerHA cluster MIB structure

The resultant MIB for PowerHA **cluster** would be 1.3.6.1.4.1.2.3.1.2.1.5.1. The data held within this MIB can be pulled by using the **snmpinfo** command shown in Example 11-1.

Example 11-1 snmpinfo command

```
# snmpinfo -v -m dump -o /usr/es/sbin/cluster/hacmp.defs cluster
clusterId.0 = 1120652512
clusterName.0 = "sapdemo71_cluster"
clusterConfiguration.0 = ""
clusterState.0 = 2
clusterPrimary.0 = 1
clusterLastChange.0 = 1386133818
clusterGmtOffset.0 = 21600
clusterSubState.0 = 32
clusterNodeName.0 = "mhoracle1"
clusterPrimaryNodeName.0 = "mhoracle1"
clusterNumNodes.0 = 2
clusterNodeId.0 = 1
clusterNumSites.0 = 0
```

Individual elements, such as the cluster state and cluster substate, can be pulled as shown in Example 11-2.

Example 11-2 Showing the cluster state

```
# snmpinfo -v -o /usr/es/sbin/cluster/hacmp.defs ClusterState.0
clusterState.0 = 2

# snmpinfo -v -o /usr/es/sbin/cluster/hacmp.defs ClusterSubState.0
clusterSubState.0 = 32

Note: the -v translates the numbered MIB branch path to reable variable name.

# snmpinfo -o /usr/es/sbin/cluster/hacmp.defs ClusterState.0
1.3.6.1.4.1.2.3.1.2.1.5.1.4.0 = 2
```

In Example 11-2, the cluster has a state of 2 and a substate of 32. To determine the meaning of these values, see the `/usr/es/sbin/cluster/hacmp.my` file, which contains a description of each HACMP MIB variable (Example 11-3 on page 382).

Example 11-3 Snapshot of the HACMP MIB definition file

```
clusterState    OBJECT-TYPE
                SYNTAX  INTEGER { up(2), down(4),
                                  unknown(8), notconfigured(256) }
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "The cluster status"

clusterSubState OBJECT-TYPE
                SYNTAX  INTEGER { unstable(16), error(64),
                                  stable(32), unknown(8), reconfig(128),
                                  notconfigured(256), notsynced(512) }
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "The cluster substate"
```

You can conclude from Example 11-3 that the cluster status is UP and STABLE. This is the mechanism that **clinfo**/**clstat** uses to display the cluster status.

The **clstat** utility uses clinfo library routines (via the clinfo daemon) to display all node, interface, and resource group information for a selected cluster. The **cldump** does likewise, as a one-time command, by interrogating the private MIB directly within the cluster node. Both rely solely on the SNMP protocol and the mechanism described above.

11.2 Custom monitoring

When it comes to monitoring a PowerHA clustered environment, what cluster status information is reported and how it is reported often varies (for example, command-line output, web browser, enterprise SNMP software). The IBM **clstat** facility is provided as a compiled binary. Therefore, it cannot be customized in any way, can be run only from an AIX OS partition, and provides basic information regarding node, adapter, and resource group status. Enterprise monitoring solutions are often complex, have cost implications, and might not provide the information that you require in a format you require. A simple and effective solution is to write your own custom monitoring scripts tailored for your environment.

The examples that follow are templates that have been written for customer environments and can be customized. The scripts are included in Appendix B, “Custom monitoring scripts” on page 415.

11.2.1 Custom example 1: Query HA (qha)

Query HA was written around 2001 for IBM High Availability Cluster Multiprocessing (HACMP) version 4, a predecessor of PowerHA. It has been updated over the years since to support the latest code levels up to version 7.1.3, the version that was current at the time of writing. Query HA primarily provides an in-cluster status view, which is not reliant on the SNMP protocol or clinfo infrastructure. It can also be easily customized to run remotely over an SSH connection from any UNIX or Linux based OS. Both in-cluster and remote cluster versions are included in Appendix B, “Custom monitoring scripts” on page 415.

Rather than simply report whether the cluster is up and running or unstable, the focus is on the internal status on the cluster manager. Although not officially documented, the internal *clstrmgr* status provides an essential understanding of what is happening within the cluster, especially during event processing (cluster changes such as start, stop, resource groups moves, application failures, and so on). When viewed alongside other information, such as the running event, the resource group status, online network interfaces, and varied on volume groups, it provides an excellent overall status view of the cluster. It also helps with problem determination as to understand PowerHA event flow during node_up or failover events, for example, and when searching through cluster and `hacmp.out` files.

PowerHA version 7 uses the Cluster Aware AIX (CAA) infrastructure for heartbeat functions across all IP and SAN-based interfaces. With versions 7.1.1 and 7.1.2, heartbeats across IP interfaces are via a special IP multicast (class D) address. In certain environments, multicasting is disabled within the Ethernet switch infrastructure; in others, multicast communications might not be allowed by the network team as a corporate policy. As such, starting with version 7.1.3, the administrator can switch to unicast for heartbeats. This is similar to previous versions that used Reliable Scalable Cluster Technology (RSCT).

From a status perspective, be sure that you know whether IP communications are multicast or unicast. If you are using multicasting and multicasting is disabled within your switch environment, the IP interfaces appear up to AIX but down to CAA. This is a particularly bad situation. **`Query HA -c`** reports the communication method (multicast or unicast) and the active status, from a CAA perspective, of all IP interfaces by using the **`lscluster -m`** command.

SAN and repository disk communications are a way of providing a *non-IP*-based network. In previous releases, the communication was handled via RSCT topology services (topsvcs) with heartbeats over disk. Now it is handled by CAA, and **`clstat`** no longer provides the status of this non-IP heartbeat communication. It is critical that this status is known and active in a running cluster. Effective with AIX 7.1 TL3, **`Query HA -c`** also provides this status via the **`clras`** command. See Example 11-4.

Example 11-4 Internal cluster manager states

```
ST_INIT: cluster configured and down
ST_JOINING: node joining the cluster
ST_VOTING: Inter-node decision state for an event
ST_RP_RUNNING: cluster running recovery program
ST_BARRIER: clstrmgr waiting at the barrier statement
ST_CBARRIER: clstrmgr is exiting recovery program
ST_UNSTABLE: cluster unstable
NOT_CONFIGURED: HA installed but not configured
RP_FAILED: event script failed
ST_STABLE: cluster services are running with managed resources (stable cluster) or
cluster services have been "forced" down with resource groups potentially in the
UNMANAGED state (from HACMP/PowerHA 5.4)
```

In addition to the default reporting status of the clstrmgr manager and the resource groups, Query HA can show the status of network interfaces, non-IP disk heartbeat networks (in version 5 and 6), online volume groups, running events, application monitoring and CAA IP/SAN/Disk communication (in version 7). It uses the various flag options that are shown in Example 11-5 on page 384.

Example 11-5 qha syntax

```
/> qha -?

Usage: qha [-n] [-N] [-v] [-l] [-e] [-m] [-1] [-c]
                -n displays network interfaces
                -N displays network interfaces + nonIP heartbeat disk
                -v shows online VGs
                -l logs entries to /tmp/qha.out
                -e shows running event
                -m shows appmon status
                -1 single interation  <<CORRECT spelling? Should be iteration.>>
                -c shows CAA IP/SAN/Disk Status (AIX7.1 TL3 min.)
```

Example 11-6 shows **qha -nevmc** running and displaying the network interfaces, such as running events, online volume groups, application monitors, and the CAA communication status.

Example 11-6 qha running example

```
# qha -nevmc
```

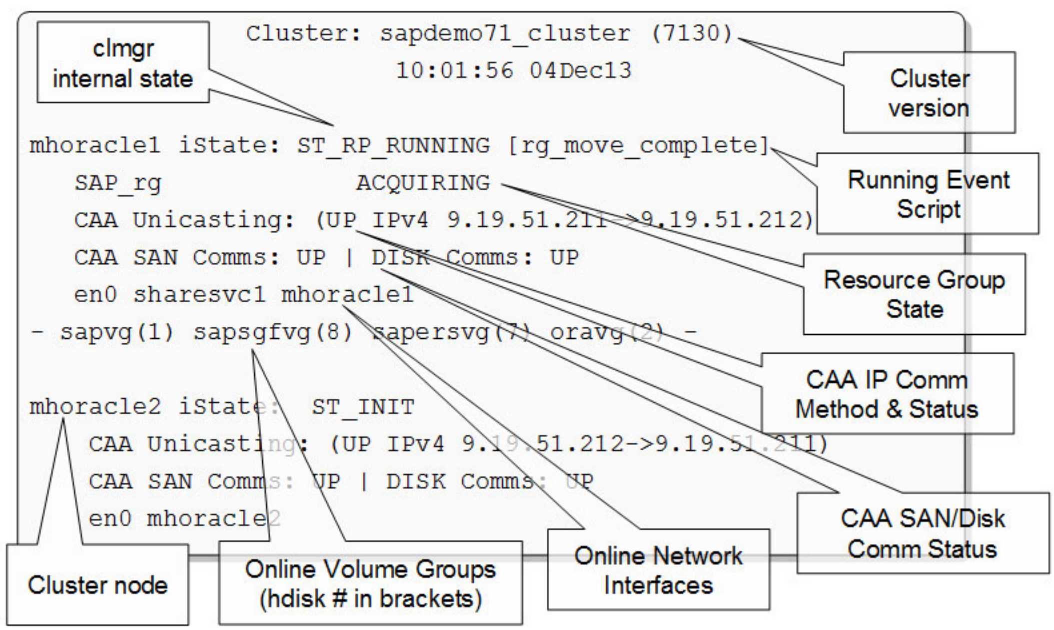


Figure 11-3 shows **qha -nvm** highlighting the application monitoring, which is a critical part of the PowerHA configuration.

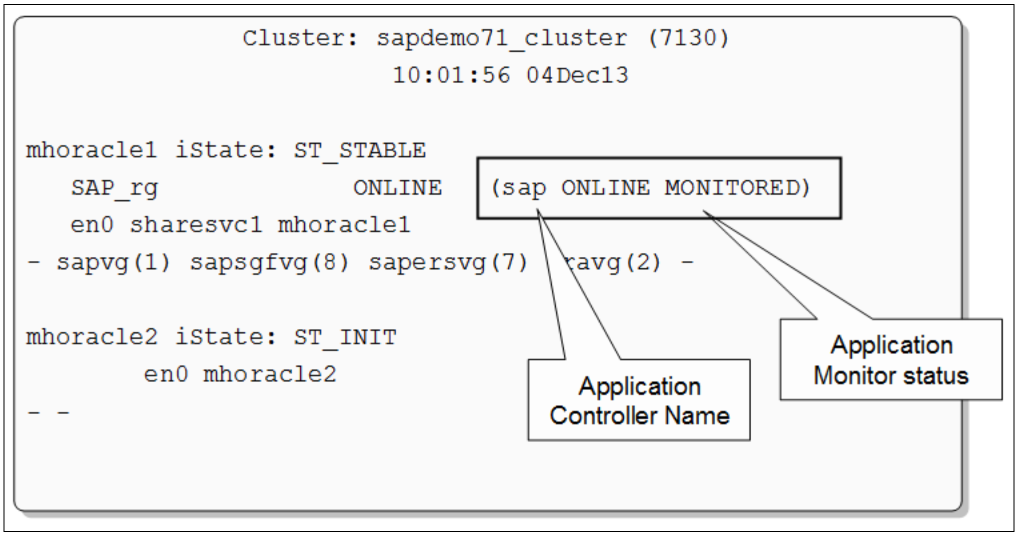


Figure 11-3 qha -nvm highlights application monitoring

In Figure 11-4, the running **qha -nvm** shows a failed application monitor.

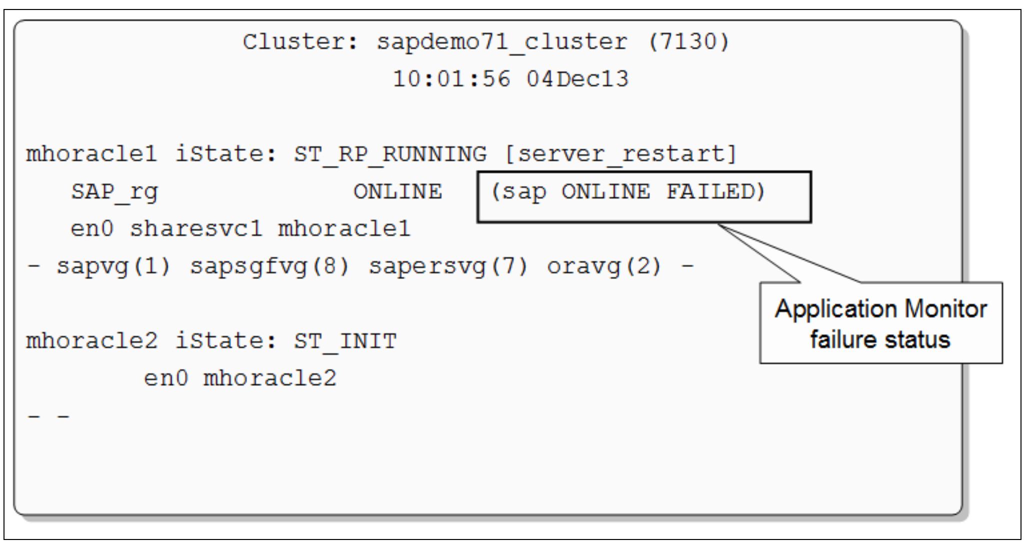


Figure 11-4 qha -nvm shows a failed application monitor

To set up Query HA, copy the script into any directory in the root's path, for example: `/usr/sbin`. Also, **qha** can be modified to send SNMP traps to a monitoring agent upon state change. To enable this feature, invoke **qha** with the **-l** flag and edit the script at the specific point as shown in Example 11-7.

Example 11-7 Adding snmp traps on state change

```
# Note, there's been a state change, so write to the log
# Alternatively, do something additional, for example: send an snmp trap
alert, using the snmptrap command. For example:
# snmptrap -c <community> -h <anmp agent> -m "appropriate message"
```

11.2.2 Custom example 2: Remote multi-cluster status monitor (qha_rmc)

The second example is based upon a customer request to provide an instant cluster overview of the node and resource group status for 32 two-node Oracle clusters. The output is displayed on the screen (if run manually from the command line) and also in HTML format, which is continually refreshed and updated. The update time default value is 5 seconds, and this value can be tuned by the administrator. The front-end HTML report was intended for first-level support personnel. An Apache (or equivalent) web server is required on the operating system that executes **qha_rmc**. See Figure 11-5.

The core of the code is based on the previous **qha** example and amended accordingly. It is recommended that **qha_rmc** is invoked from cron.

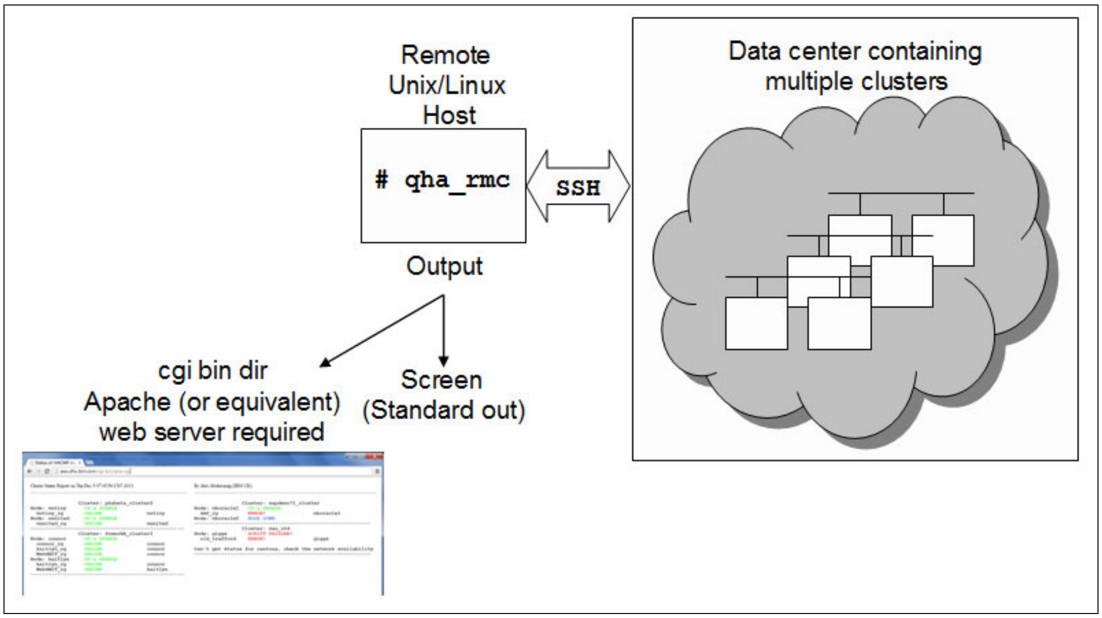


Figure 11-5 qha_rmc overview

Figure 11-6 on page 387 shows the **qha_rmc** snapshot run manually from the command line, which shows the output from our test clusters.

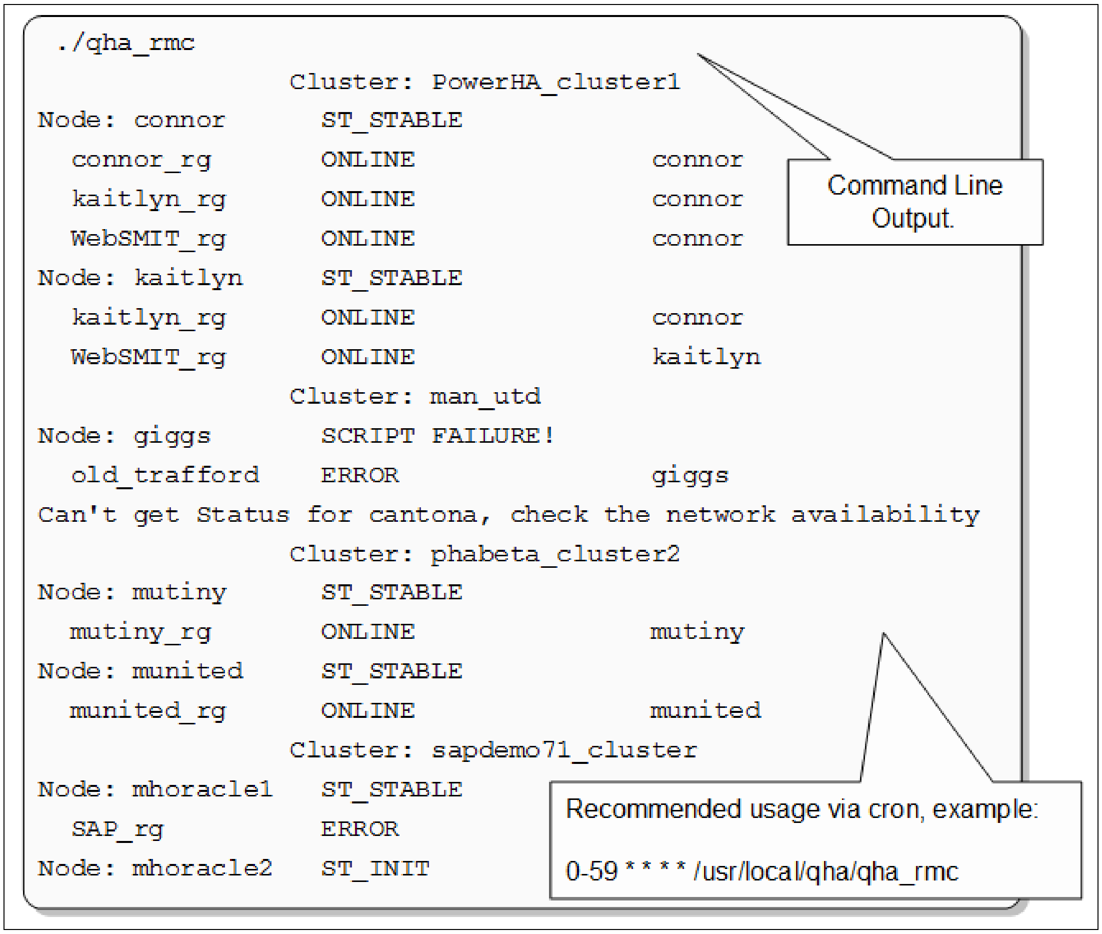
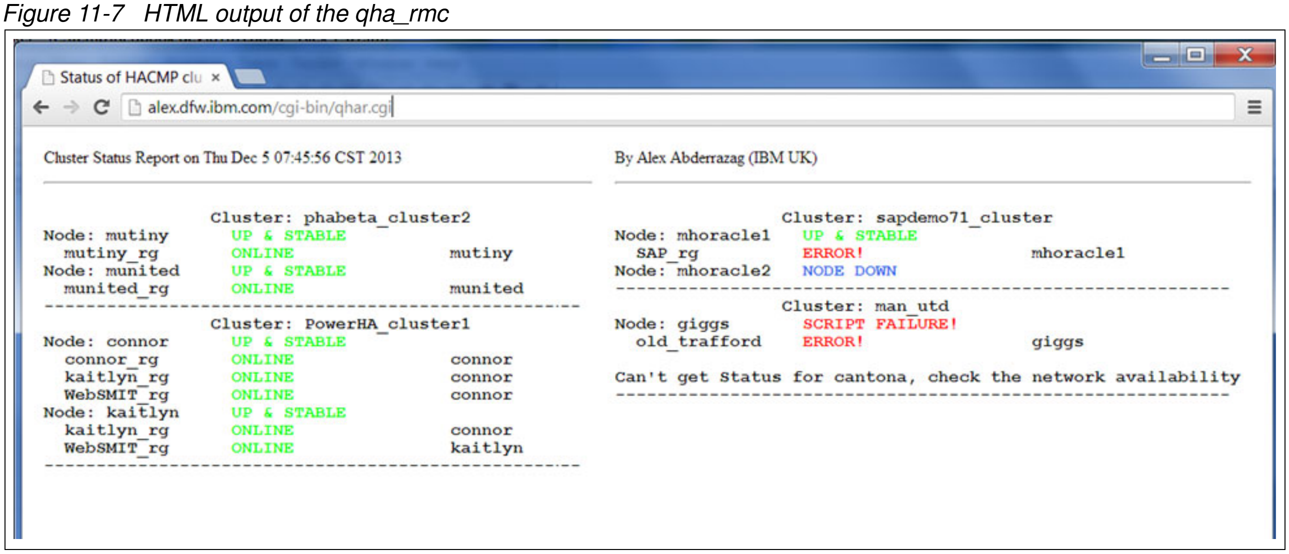


Figure 11-6 qha_rmc snapshot runs manually from the command line

The HTML output is shown in Figure 11-7.

Figure 11-7 HTML output of the qha_rmc



To set up **qha_rmc**, copy the script to a suitable location in the users path. Make sure that unprompted SSH access is configured for each cluster node. To create a cluster definition file, use this file format, as shown in Example 11-8 on page 388:

```
Cluster: <name of the cluster>:<resolvable cluster node names or IP addresses,
space delimited.>
```

Example 11-8 Cluster definition file for qha_rmc (`CLUSTERfile=/alex/QHAhosts`*)*

```
cluster:sapdemo71_cluster:mhoracle1 mhoracle2
cluster:PowerHA_cluster1:connor kaitlyn
```

Now, edit the script and adjust the global variables as appropriate, as shown in Example 11-9.

Example 11-9 Adjusting the global variables

```
CLUSTERfile=/alex/QHAhosts
CGIPATH=/opt/freeware/apache/share/cgi-bin #Path to Web server cgi-bin
CGIFILE="$CGIPATH/qhar.cgi"
```

Depending on the number of clusters to be monitored, you might have to adjust the `SLEEPTIME` variable in the global variables at the start of the script.

11.2.3 Custom example 3: Remote SNMP status monitor (liveHA)

The third example is called *liveHA*. It is similar to **`clstat`** but fully customizable by the user. Architecturally, it is the same as 11.2.2, “Custom example 2: Remote multi-cluster status monitor (qha_rmc)” on page 386, but it focuses on a single cluster rather than a multiple one. It is possible to have multiple instances running, each reporting the status of different clusters. liveHA is intended to run remotely, outside of the cluster. It obtains the cluster status by interrogating the SNMP MIB over SSH without the need for the clinfo daemon. Further, it uses the same standard AIX and PowerHA commands, such as **`lssrc`**, **`clRGinfo`**, and **`clras`**, to display information that is not held in the SNMP MIB (such as the resource group status, for example).

liveHA is invoked from the command line (as shown in Example 11-10) and produces both text and CGI outputs over SSH (in the same operation), in a way that is similar to **`qha_rmc`**, as shown in 11.2.2, “Custom example 2: Remote multi-cluster status monitor (qha_rmc)” on page 386. liveHA runs from any OS that supports Korn Shell. In addition to **`clstat`**, liveHA shows the active node being queried, the internal cluster manager status, and the status of the CAA SAN communications.

Example 11-10 liveHA syntax

```
./liveHA -?

Usage: liveHA [-n] [-1] [-i]
        -n Omit Network info
        -1 Display 1 report rather than loop
        -i Displays the internal state of cluster manager
        -c Displays the state of SAN Communications
```

Example 11-11 on page 389 shows the running liveHA.

Example 11-11 liveHA in operation: `# liveHA -ic`

```
                Status for sapdemo71_cluster on 09 Dec 13 05:43:07
                     Cluster is (UP & STABLE)    qn: mhoracle1

Node : mhoracle1       State: UP (ST_STABLE)

  Network : net_ether_01       State: UP
        9.19.51.211     mhoracle1           UP
        9.19.51.239     sharesvc1           UP
  CAA SAN Comms          State: UP

        Resource Group(s) active on mhoracle1:
          SAP_rg          ONLINE
          alexRG          ONLINE

Node : mhoracle2       State: DOWN (ST_INIT)

  Network : net_ether_01       State: DOWN
        9.19.51.212     mhoracle2           DOWN
  CAA SAN Comms          State: UP
```

Figure 11-8 shows the SMIT screen while monitoring a remote cluster via SSH/SNMP.

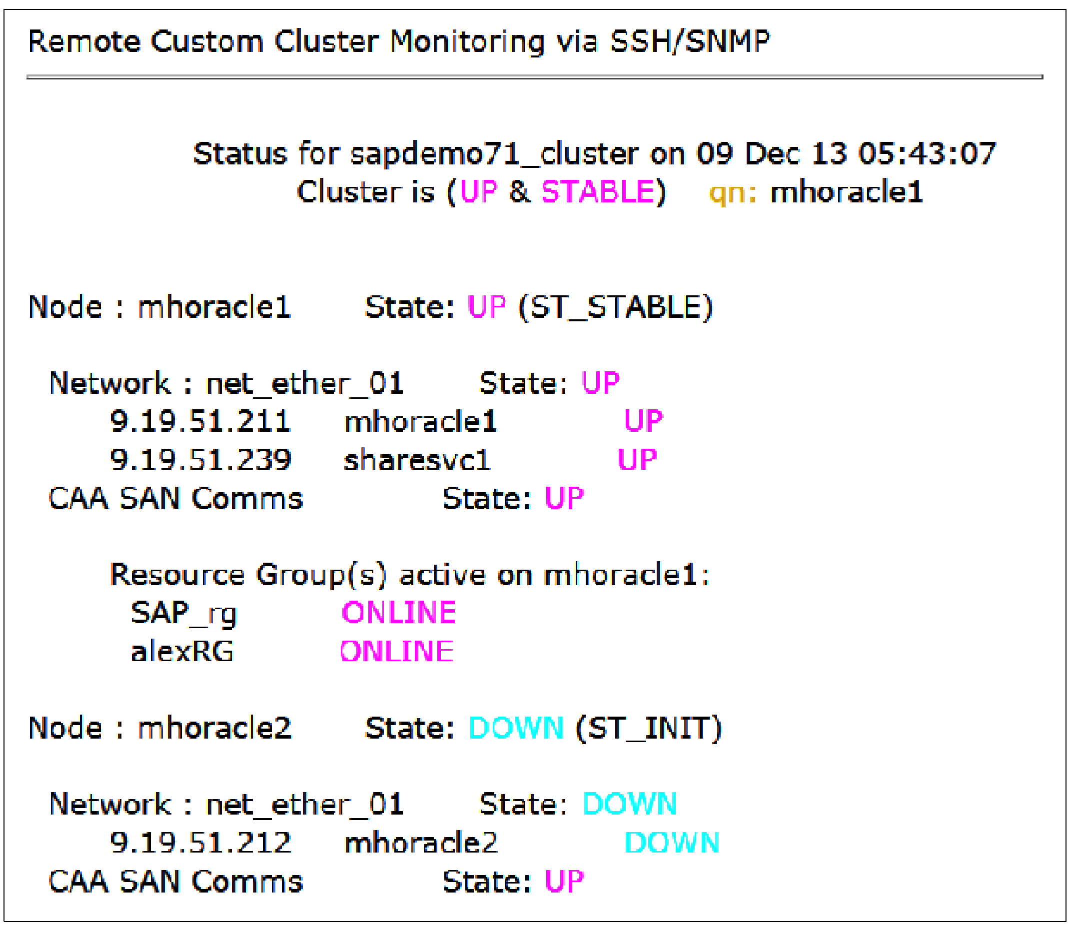


Figure 11-8 Remote customer cluster monitoring via SSH/SNMP

To use liveHA, first place the script in a directory contained within the users path. Then configure unprompted SSH access between the machine running liveHA and the cluster. Edit the liveHA script and change the global variables to suit your environment, and run the script. The `clhost` file must contain a resolvable name of each node in the cluster. See Example 11-12.

Example 11-12 liveHA global variables and clhost file example

```
LOGFILE="/tmp/.qhaslog.$$" #General log file
HTMLFILE="/tmp/.qhashtml.$$" #HTML output file
CGIPATH=/opt/freeware/apache/share/cgi-bin #Path to Web server cgi-bin
CGIFILE="$CGIPATH/qhasA.cgi" #CGI file to be displayed in the web browser
CLHOSTS="/alex/clhosts" #Populate this file with the resolvable names of each
cluster node
USER=root # to be used for ssh access
SNMPCOMM=public #SNMP community name

#cat clhosts
mhoracle1
mhoracle2
```

11.3 PowerHA cluster log monitoring

This section focuses on monitoring the PowerHA cluster log (`/var/hacmp/adm/cluster.log`) for various events that are associated with cluster operations. Although there are many monitoring tools available in the market for log monitoring, this section highlights using IBM Tivoli Monitoring to monitor the cluster log.

Note: This section requires an understanding of IBM Tivoli Monitoring v6.1.x or later and the concept of the IBM Tivoli Monitoring agent for UNIX logs.

11.3.1 IBM Tivoli Monitoring agent for UNIX logs

The monitoring agent for UNIX logs provides the capability to monitor UNIX logs effectively. It performs the following actions:

- Creates situations that are triggered when specific messages are written to a log so that you can take a more proactive approach to managing the systems. This means that you can respond to events as soon as they occur and take action to prevent potential problems from developing.
- Eliminates the need to manually analyze large log files because the monitoring agent for UNIX logs screens all log entries and forwards only selected entries to the Tivoli Enterprise Portal, which is the interface for the monitoring software.
- Shifts the emphasis of management from post-mortem diagnosis to real-time response. The monitoring agent enables you to increase the amount of log data that is collected by system daemons and user applications and decrease the amount of data that is stored for historical debugging and analysis.
- Retrieves log entries that occurred within a certain time span from any monitored log. Data from different log types can be presented in a common format within a Tivoli Enterprise Portal workspace.

> **Note:** Tivoli Monitoring v6.1.x or later is the base software for the monitoring agent for UNIX logs.

11.3.2 PowerHA cluster log

PowerHA SystemMirror writes the messages that it generates to the system console and to several log files. Because each log file contains a different subset of the types of messages generated by PowerHA SystemMirror, you can get different views of the cluster status by viewing different log files.

The `/var/hacmp/adm/cluster.log` file is the main PowerHA SystemMirror log file. PowerHA SystemMirror error messages and messages about PowerHA SystemMirror-related events are appended to this log with the time and date when they occurred.

11.3.3 Installing and configuring cluster monitoring

The following sections provide detailed steps to set up and configure cluster log monitoring through IBM Tivoli Monitoring agent for UNIX logs.

Set up the IBM Tivoli Monitoring infrastructure

Follow these steps to set up the IBM Tivoli Monitoring infrastructure:

1. Design and plan your IBM Tivoli Monitoring environment for your enterprise.
2. Install the required IBM Tivoli Monitoring components:
   a. Install the hub Tivoli Enterprise Monitoring Server as a collection and control point for alerts received from the monitoring agents.
   b. Install any remote monitoring servers that are required, based on your environment size and requirements.
   c. Install the Tivoli Enterprise Portal Server to enable retrieval, manipulation, and analysis of data from the monitoring agents.

> **Note:** See the IBM Tivoli Monitoring Information Center for detailed Installation and configuration guide:
>
> `http://ibm.co/UOQxNf`

Add application support in IBM Tivoli Monitoring Infrastructure

Application support includes the necessary workspaces and situations for each agent. Therefore, install the respective application support for Tivoli Monitoring agent for UNIX logs (ul) on the monitoring server and portal server.

> **Note:** *ul* is the agent code for the monitoring agent for UNIX logs.

You can ensure that the required application support is installed, as Example 11-13 on page 392 shows.

Example 11-13 Verification of application support for UNIX logs

```
[root:/opt/IBM/ITM/bin:] ./cinfo -i
ul      Monitoring Agent for UNIX Logs
         tms     Version: 06.22.08.00
         tps     Version: 06.22.02.00
         tpw     Version: 06.22.02.00root@asterix(/)
```

Install Tivoli Monitoring agent for UNIX logs in the cluster nodes

Follow these steps to install IBM Tivoli Monitoring Agent in the PowerHA cluster nodes:

1. Install the Tivoli Monitoring agent for UNIX logs (ul) in all the nodes of the PowerHA cluster.
2. Configure the ul agent to establish connectivity to the monitoring server.
3. Ensure the installation of the agent, as shown in Example 11-14.

Example 11-14 Monitoring Agent for UNIX logs Installation

```
[root:/opt/IBM/ITM/bin:] ./cinfo -i
ul      Monitoring Agent for UNIX Logs
         aix526  Version: 06.22.08.00
```

Enable log monitoring in the PowerHA cluster nodes

After installing and configuring the Tivoli Monitoring agent for UNIX logs in the cluster nodes, enable the `cluster.log` file monitoring:

1. Open the `$CANDLEHOME$/config/kul_configfile` and ensure that the following line is present:

   `KUL_CONFIG_FILE=$CANDLEHOME$/config/kul_configfile`

> **Note:** `$CANDLEHOME` refers to the directory where the IBM Tivoli Monitoring components are installed. Typically, it is this path: `/opt/IBM/ITM`

2. Append the following line to the `$CANDLEHOME$/config/kul_configfile` to enable the `cluster.log` file monitoring:

   ```
   /var/hacmp/adm/cluster.log      ;n      ;u      ;a,"%s %d %d:%d:%d %s %[^\n]" ,
   month day hour min sec source desc
   ```

3. Save the `kul_configfile` file.
4. Restart the ul agent.
5. You must be able to see the log entries in Tivoli Enterprise Portal Server workspaces.

   As the `cluster.log` is updated by PowerHA, you see the appropriate updates in the Tivoli Enterprise Portal.

6. Configure the situations (events) in the Tivoli Enterprise Portal Server for event alerts and integration with the Event Management console.

11.3.4 IBM Tivoli Monitoring situations for PowerHA event monitoring

This section lists some of the recommended situations that may be implemented for the PowerHA cluster monitoring, as shown in Table 11-1. You may extend to monitor a large number of situations.

Table 11-1 Recommended monitoring situations

Situation name	Description	Formula
sit_acq_serviceaddr	This situation is triggered when the local node joins the cluster or a remote node leaves the cluster.	`*IF *SCAN Log_Entries.Log_Name *EQ 'cluster.log' *AND *SCAN Log_Entries .Description *EQ 'acquire_service_addr'`
sit_acq_takeoveraddr	This situation is triggered when a remote node leaves the cluster.	`*IF *SCAN Log_Entries.Log_Name *EQ 'cluster.log' *AND *SCAN Log_Entries.Description *EQ 'acquire_takeover_addr'`
sit_fail_interface	This situation is triggered when an adapter goes down.	`*IF *SCAN Log_Entries.Log_Name *EQ 'cluster.log' *AND *SCAN Log_Entries.Description *EQ 'fail_interface'`
sit_fail_stdby	This situation is triggered when restoring the route for the remaining standby on subnet.	`*IF *SCAN Log_Entries.Log_Name *EQ 'cluster.log' *AND *SCAN Log_Entries.Description *EQ 'fail_standby'`
sit_join_interface	This event script is called when an adapter comes up.	`*IF *SCAN Log_Entries.Log_Name *EQ 'cluster.log' *AND *SCAN Log_Entries.Description *EQ 'join_interface'`
sit_join_stdby	This event is triggered when trying to restore the route for the remaining standby on the subnet.	`*IF *SCAN Log_Entries.Log_Name *EQ 'cluster.log' *AND *SCAN Log_Entries.Description *EQ 'join_standby'`
sit_nodedown	This event is triggered when a node leaves the cluster.	`*IF *SCAN Log_Entries.Log_Name *EQ 'cluster.log' *AND *SCAN Log_Entries.Description *EQ 'node_down'`
sit_nodeup	This event is triggered when a node joins the cluster.	`*IF *SCAN Log_Entries.Log_Name *EQ 'cluster.log' *AND *SCAN Log_Entries.Description *EQ 'node_up'`

11.4 PowerHA cluster SNMP trap monitoring

This section focuses on monitoring the PowerHA cluster events through SNMP trap-based monitoring. Although any monitoring tool that is capable of processing SNMP traps can be used, it shows you how IBM Tivoli Monitoring can be used for PowerHA monitoring through SNMP traps.

IBM Tivoli Monitoring v6.1 and later supports a type of agent called *IBM Tivoli Universal Agent*, which is a generic agent of IBM Tivoli Monitoring. In the next sections, monitoring PowerHA SNMP traps through the Tivoli Universal Agent is explained.

> **Note:** This section requires an understanding of IBM Tivoli Monitoring v6.1.x or later and the concept of the IBM Tivoli Universal Agent.

11.4.1 IBM Tivoli Universal Agent

You can configure the IBM Tivoli Universal Agent to monitor any data that you collect. You can view the data in real-time and historical workspaces on the Tivoli Enterprise Portal and manage with Tivoli Enterprise Portal monitoring situations and automation policies, the same as data from other Tivoli Enterprise Monitoring agents.

The IBM Tivoli Universal Agent extends the performance and availability management capabilities of IBM Tivoli Monitoring to applications and operating systems not covered by other IBM Tivoli Monitoring agents. It gives you a single point to manage all of your enterprise resources and protects your investment in applications and resources.

The IBM Tivoli Universal Agent provides the following benefits:

- Integrates data from virtually any operating system and any source, including custom applications, databases, systems, subsystems, and networks.
- Monitors only the data attributes of interest.
- Responds quickly to changing monitoring and management scenarios.
- Gives you control of attributes and surfacing of data.

11.4.2 Tivoli Universal Agent data provider

Data providers are the interfaces of the Tivoli Universal Agent. They handle these functions:

- Collect data from data sources, such as log files, client programs, URLs, scripts, relational tables, or SNMP agents.
- Pass the collected data and the information about the data definition metafiles to the IBM Tivoli Universal Agent.

This scenario is based on using the SNMP Data Provider. It brings the functionality of Simple Network Management Protocol (SNMP) management capability to IBM Tivoli Monitoring, which enables you to integrate network management with systems and applications management. This includes network discovery and trap monitoring.

Through the SNMP Data Provider, the Universal Agent can monitor any industry standard MIB or any MIB that you supply. Tivoli Monitoring creates Universal Agent applications for you by converting the MIBs into data definition metafiles. You can then monitor any MIB variable as an attribute and monitor any SNMP traps that are sent to the data provider.

Note: This method supports only SNMPv1 traps.

11.4.3 PowerHA SNMP support

The cluster manager provides SNMP support to client applications. SNMP is an industry-standard specification for monitoring and managing TCP/IP-based networks. It includes a protocol, a database specification, and a set of data objects. This set of data objects forms a Management Information Base (MIB). SNMP provides a standard MIB that includes information such as IP addresses and the number of active TCP connections. The standard SNMP agent is the snmpd daemon.

The cluster manager maintains cluster status information in a special PowerHA SystemMirror MIB (`/usr/es/sbin/cluster/hacmp.my`). When the cluster manager starts on a cluster node, it registers with the SNMP snmpd daemon, and then continually gathers cluster information. The cluster manager maintains an updated topology of the cluster in the PowerHA SystemMirror MIB as it tracks events and the resulting states of the cluster.

Important: The default `hacmp.my` that is installed with the PowerHA cluster file sets for V7.1.3 has errors that are corrected with the installation of PowerHA V7.1.3 SP1. See Appendix B, "Custom monitoring scripts" on page 415 for the file to use in the earlier versions of PowerHA.

11.4.4 Installing and configuring PowerHA SNMP trap monitoring

The following sections provide detailed steps to set up and configure PowerHA SNMP monitoring through the IBM Tivoli Universal Agent.

Set up the IBM Tivoli Monitoring infrastructure

Follow these steps to set up the IBM Tivoli Monitoring Infrastructure:

1. Design and plan your IBM Tivoli Monitoring environment for your enterprise.
2. Install the required IBM Tivoli Monitoring components:
   a. Install the Hub Tivoli Enterprise Monitoring Server, which acts as a collection and control point for alerts received from the monitoring agents.
   b. Install any remote monitoring servers required, based on your environment size and requirements.
   c. Install the Tivoli Enterprise Portal Server, which enables retrieval, manipulation, and analysis of data from the monitoring agents.

Note: See the IBM Tivoli Monitoring Information Center for detailed Installation and Configuration guide:

`http://ibm.co/UOQxNf`

Add application support for IBM Tivoli Monitoring infrastructure

Application support includes the necessary workspaces and situations for each agent. Install the support for IBM Tivoli Universal Agent (um) on the monitoring server and portal server.

Note: *um* is the agent code for IBM Tivoli Universal Agent.

You can ensure that the required application support is installed as shown in Example 11-15.

Example 11-15 Application support for IBM Tivoli Universal Agent

```
[root:/opt/IBM/ITM/bin:] ./cinfo -i
um      Universal Agent
         tms     Version: 06.22.08.00
         tps     Version: 06.22.02.00
         tpw     Version: 06.22.02.00
[root:/opt/IBM/ITM/bin:]
```

Configure SNMP in the PowerHA cluster nodes

Follow these steps to configure SNMP in all the nodes of a PowerHA cluster:

1. The latest version of AIX has the SNMPv3 daemon enabled by default. You can configure SNMP version 3 with the `/etc/snmpdv3.conf` file.

   Note: The **`ls -l /usr/sbin/snmpd`** command returns the version of snmpd that is running on the server.

2. A typical `/etc/snmpdv3.conf` that receives PowerHA SNMP traps has the entries as shown in Example 11-16.

Example 11-16 SNMP v3 daemon configuration (/etc/snmpdv3.conf)

```

VACM_GROUP group1 SNMPv1  MyCommunity  -

VACM_VIEW defaultView       1.3.6.1.4.1.2.2.1.1.1.0    - included -
VACM_VIEW defaultView       1.3.6.1.4.1.2.6.191.1.6    - included -

# exclude snmpv3 related MIBs from the default view
VACM_VIEW defaultView       snmpModules                - excluded -
VACM_VIEW defaultView       1.3.6.1.6.3.1.1.4          - included -
VACM_VIEW defaultView       1.3.6.1.6.3.1.1.5          - included -

# exclude aixmibd managed MIBs from the default view
VACM_VIEW defaultView       1.3.6.1.4.1.2.6.191        - excluded -

VACM_ACCESS  group1 - - noAuthNoPriv SNMPv1  defaultView - defaultView -

NOTIFY notify1 traptag trap -

TARGET_ADDRESS Target1 UDP 1.1.1.1 traptag trapparms1 - - -
#TARGET_ADDRESS Target1 UDP 127.0.0.1       traptag trapparms1 - - -

TARGET_PARAMETERS trapparms1 SNMPv1  SNMPv1  MyCommunity  noAuthNoPriv -

COMMUNITY MyCommunity    MyCommunity     noAuthNoPriv 0.0.0.0     0.0.0.0
-

DEFAULT_SECURITY no-access - -

logging         file=/usr/tmp/snmpdv3.log       enabled
logging         size=100000                     level=0
```

```
VACM_VIEW defaultView    internet       - included -
VACM_VIEW defaultView           1.3.6.1.4.1.2.3.1.2.1.5         - included -

smux 1.3.6.1.4.1.2.3.1.2.1.2         gated_password  # gated
smux     1.3.6.1.4.1.2.3.1.2.1.5 clsmuxpd_password # PowerHA SystemMirror clsmuxpd
smux 1.3.6.1.4.1.2.3.1.2.3.1.1 muxatmd_password #muxatmd
```

Note: MyCommunity is the community name used in Example 11-16 on page 396. You may replace the community name with your own community name or leave the default community name, *Public*.

In Example 11-16 on page 396, the target server, 1.1.1.1, is the server where the IBM Tivoli Universal Agent is installed, as explained in "Install the IBM Tivoli Universal Agent" on page 397.

3. Restart the snmpd daemon as shown in Example 11-17.

Example 11-17 Restart SNMP daemon

```
[root:/home/root:] stopsrc -s snmpd
0513-044 The snmpd Subsystem was requested to stop.
[root:/home/root:] startsrc -s snmpd
0513-059 The snmpd Subsystem has been started. Subsystem PID is 3604548.
[root:/home/root:]
```

4. Wait for a few seconds for the following line to appear in the `/var/hacmp/log/clstrmgr.debug` file:

   `"smux_simple_open ok, try smux_register()"`

5. Ensure the correctness of the SNMP configuration by running the **`cldump`** or the **`clstat`** command.

Install the IBM Tivoli Universal Agent

Follow these steps to install the Universal Agent:

1. Identify a centralized server in your environment where the Tivoli Universal Agent can be installed and the SNMP traps can be sent from the PowerHA cluster nodes.
2. Install the Tivoli Universal Agent (um) in the identified server.
3. Configure the UL agent to establish connectivity to the monitoring server.
4. Verify the installation of the agent, as shown in Example 11-18.

Example 11-18 IBM Tivoli Universal Agent installation

```
[root:/opt/IBM/ITM/bin:] ./cinfo -i
um      Universal Agent
         aix526  Version: 06.22.08.00
```

Configure and enable SNMP monitoring

The following steps are required to configure Tivoli Universal Agent and enable monitoring of SNMP traps received from the PowerHA cluster nodes:

1. By default, the SNMP data provider is not enabled with the default configuration of the Tivoli Universal Agent. Re-configure the Universal Agent to include the SNMP data provider by using this command:

   ```
   $CANDLEHOME/bin/itcmd config -A um
   ```

> **Note:** `$CANDLEHOME` refers to the directory where IBM Tivoli Monitoring components are installed, which is typically: `/opt/IBM/ITM`

2. Ensure that the SNMP data provider is enabled through the following line in um.config:

   ```
   KUMA_STARTUP_DP='ASFS,SNMP'
   ```

3. Define the IBM Tivoli Universal Agent application by building the appropriate data definition metafile.

   If you are well-versed in Universal Agent data definition control statements, you may use the default *hacmp.my* (`/usr/es/sbin/cluster/hacmp.my`) to build up the Universal Agent metafile manually.

   Alternatively, you may use MibUtility, which is available from OPAL, to convert the MIB file (`/usr/es/cluster/utilities/hacmp.my`) into an IBM Tivoli Monitoring Universal Agent application. Append the generated `trapcnfg_* file to TRAPCNFG` file, the location of which is defined in the **`$KUM_WORK_PATH`** environment variable.

> **Note:** The `PowerHA.mdl` metafile and the `trapcnfg` trap file are included in Appendix B, “Custom monitoring scripts” on page 415 for your reference.

4. Import the resultant metafile by using the **`$CANDLEHOME/bin/um_console`** command. When it prompts you to enter a command, enter **`validate PowerHA.MDL`**.

   After validating, it prompts for importing the MDL file. Type **`Import`** and press Enter to import the MDL file to the server.

5. Notice that the appropriate workspaces are created in Tivoli Enterprise Portal.
6. Simulate an event in PowerHA, for example bringing the resource group offline.

   You should see appropriate traps received in the Tivoli Enterprise Portal.

You can now proceed to create appropriate situations for automated event monitoring and subsequent event escalation.

11.5 SNMPv1 daemon support for PowerHA trap monitoring

This section focuses on monitoring the PowerHA cluster events through SNMP v1 daemon-based monitoring.

11.5.1 SNMP v1

Simple Network Management Protocol (SNMP) version 3 is the default version that is used in the latest releases of the AIX operating system. However, you can use SNMP version 1 and configure it with the `/etc/snmpd.conf` file for PowerHA cluster trap monitoring if you prefer.

You can switch from SNMP v3 to SNMP v1 by using the command shown in Example 11-19.

Example 11-19 IBM Tivoli Universal Agent installation

```
[root:/home/root:]  /usr/sbin/snmpv3_ssw -1
Stop daemon: snmpmibd
Stop daemon: snmpd
Make the symbolic link from /usr/sbin/snmpd to /usr/sbin/snmpdv1
Make the symbolic link from /usr/sbin/clsnmp to /usr/sbin/clsnmpne
Start daemon: dpid2
Start daemon: snmpd
[root:/home/root:]
[root:/home/root:] ls -l /usr/sbin/snmpd
lrwxrwxrwx    1 root     system          17 Dec 19 22:21 /usr/sbin/snmpd ->
/usr/sbin/snmpdv1
[root:/home/root:]
```

11.5.2 SNMP v1 daemon configuration

This section provides a sample SNMPv1 configuration file that can be used for sending PowerHA cluster traps to any SNMP manager or SNMP monitoring tool for trap processing or monitoring.

Additionally, you can use the configuration file as shown in Example 11-20 to integrate PowerHA cluster traps with the Tivoli Universal Agent as described in section 11.4, “PowerHA cluster SNMP trap monitoring” on page 394.

Example 11-20 SNMP v1 daemon configuration (/etc/snmpd.conf)

```

logging         file=/usr/tmp/snmpd.log         enabled
logging         size=100000                     level=0

community       MyCommunity
#community       private 127.0.0.1 255.255.255.255 readWrite
#community       system  127.0.0.1 255.255.255.255 readWrite 1.17.2

view            1.17.2          system enterprises view

trap            MyCommunity 2.2.2.2 1.2.3   fe      # loopback

#snmpd          maxpacket=1024 querytimeout=120 smuxtimeout=60

smux            1.3.6.1.4.1.2.3.1.2.1.2         gated_password  # gated
smux            1.3.6.1.4.1.2.3.1.2.2.1.1.2     dpid_password   #dpid

smux     1.3.6.1.4.1.2.3.1.2.1.5 clsmuxpd_password # HACMP/ES for AIX clsmuxpd
#
```

Note: MyCommunity is the community name used in Example 11-18 on page 397. You may replace the community name with your own community name or leave the default community name, which is *Public*.

In Example 11-20 on page 399, *2.2.2.2* is the SNMP Manager that is capable of monitoring SNMP v3 traps.

A

Repository disk recovery procedure

This appendix describes recreating the Cluster Aware AIX (CAA) repository disk after a multi-component outage. We explain a possible failure scenario and provide a procedure that re-creates the repository disk after a complete cluster failure when the repository disk is still missing. The example is based on a two-site stretched cluster environment. However, it is also valid in a single-site cluster environment.

This appendix covers the following topics:

- Outage scenario
- Recovering from a failure with PowerHA 7.1.3 and later
- Reintegrating a failed node

Outage scenario

Example A-1 shows an example of a two-site stretched cluster configuration with only one active and one backup repository disk. In the environment shown, assume that due to limitations in the storage infrastructure, a storage-based mirroring of the repository disk as recommended in 6.1, “Introduction to the example scenario” on page 104, is not possible.

The hdisk2 is defined as repository disk, so it is assigned to the caavg_private volume group, and hdisk5 is defined as backup_repository. Example A-1 shows the configuration.

Example A-1 Repository disk configuration of the cluster

```
root@a2:/> lspv
hdisk0          00f70c99540419ff                    rootvg          active
hdisk1          00f70c9975f30ff1                    None
hdisk2          00f70c9976cc355b                    caavg_private   active
hdisk3          00f70c9976cc35af                    sasapp_vg
hdisk4          00f70c9976cc35e2                    sasapp_vg
hdisk5          00f70c9976cc3613                    None
hdisk6          00f70c9976cc3646                    None
root@a2:/> odmget HACMPsircol

HACMPsircol:
        name = "sas_itso_cl_sircol"
        id = 1
        uuid = "0"
        ip_address = ""
        repository = "00f70c9976cc355b"
        backup_repository = "00f70c9976cc3613
```

In the configuration shown in Figure A-1 on page 403, all single failures of a component, node, or a whole data center could be covered by IBM PowerHA 7.1 within the standard procedures. In a rolling disaster or a multi-component outage, a situation might occur where PowerHA is not able to restart the service with standard procedures.

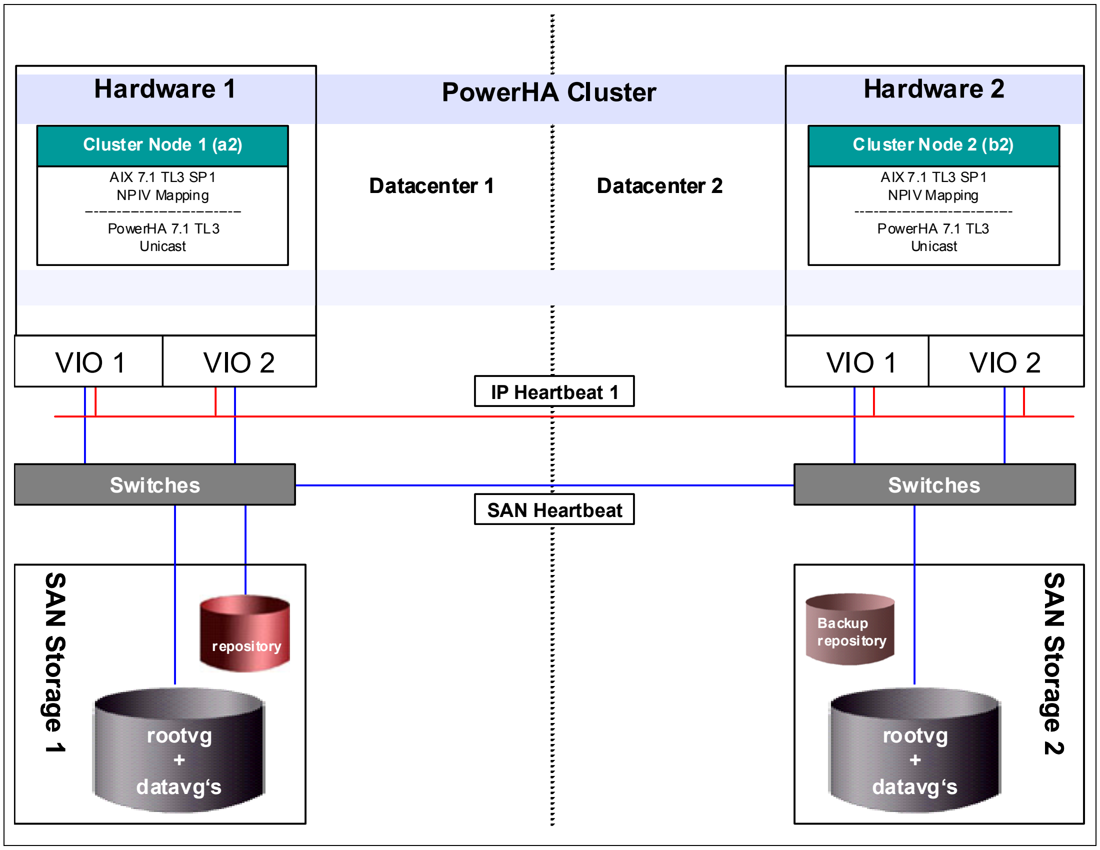


Figure A-1 Two-node stretched cluster with unmirrored repository disk

An example of an outage where multiple components are involved is described in Figure A-2 on page 405. Datacenter 1 completely fails due to a power outage caused by the network provider. Cluster node 1 fails, and cluster node 2 loses the repository disk. Example A-2 shows the entries logged in `/var/hacmp/log/hacmp.out`.

Example A-2 Entries logged in /var/hacmp/log/hacmp.out

```
ERROR: rep_disk_notify : Wed Dec 11 02:54:00 EST 2013 : Node b2 on Cluster
sas_itso_cl has lost access to repository disk hdisk2. Please recover from this
error or replace the repository disk using smitty.
clevmgrd: Wed Dec 11 02:54:24 2013 NODE_DOWN on node
0x5741F7C052EF11E3ABD97A40C9CE2704

                        HACMP Event Preamble
----------------------------------------------------------------------------

Node 'a2' is down.

Enqueued rg_move release event for resource group 'sasapp_rg'.

Enqueued rg_move acquire event for resource group 'sasapp_rg'.

Node Down Completion Event has been enqueued.
----------------------------------------------------------------------------
```

Example A-3 shows the errpt entry on node b2, which shows the failure of the repository disk.

Example A-3 errpt entry on node b2 shows failure of repository disk

```
LABEL:          OPMSG
IDENTIFIER:     AA8AB241

Date/Time:       Tue Apr 29 17:40:25 2014
Sequence Number: 3663
Machine Id:      00C0FB324C00
Node Id:         b2
Class:           O
Type:            TEMP
WPAR:            Global
Resource Name:   clevmgrd

Description
OPERATOR NOTIFICATION

User Causes
ERRLOGGER COMMAND

        Recommended Actions
        REVIEW DETAILED DATA

Detail Data
MESSAGE FROM ERRLOGGER COMMAND
INFORMATION: Invoked rep_disk_notify event with PID 15204572 +++
```

In the failure scenario where Datacenter 1 has failed completely (Figure A-2 on page 405), PowerHA fails over the workload to node 2 in Datacenter 2 and allows the business to continue. In this case, the cluster operates in the restricted mode, without any repository disk. It is expected that an administrator notices the repository disk failure or unavailability and uses the repository disk replacement menus to enable a new repository disk for the cluster.

However, assume that Datacenter 1 failed and, for some reason, when the workload is failing over or even after it has failed over, node 2 reboots (intentionally or otherwise) without recreating a new repository. After reboot in that situation, node 2 would not have any repository disk to start the cluster. This is related to the missing repository disk hosted in Datacenter 1. Then, it becomes necessary that a repository disk recovery process be initiated to re-create the repository disk and allow node 2 to start the cluster and workload.

After node 2 has started using a new disk as repository, certain steps are needed on node 1 also (after Datacenter 1 recovers) so that it can start using the new repository disk. All of these recovery steps are explained in the following sections.

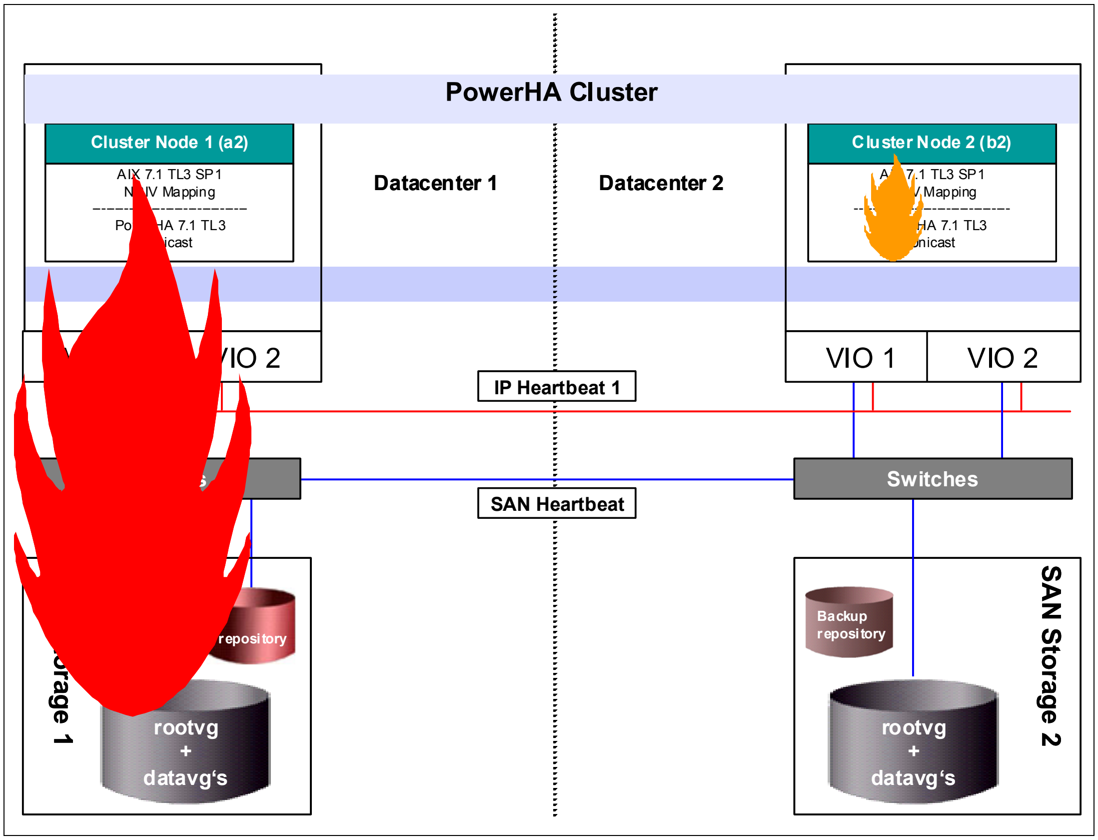


Figure A-2   Example outage with multiple failing components

In releases before PowerHA 7.1.3, a new CAA cluster definition setup with only the available node is required before PowerHA including service could be started.

Starting with PowerHA 7.1.3, a new process is available that re-creates the previous cluster configuration on a new disk. See Table A-1 for caavg_recreate support that is available with different versions of PowerHA.

Table A-1   Availability of caavg_recreate feature in different levels

Version	Re-create possible	Fix
< PowerHA 7.1.2	Special procedure during outage.	Please contact your support center.
= PowerHA 7.1.2	(YES)	Backport required. Please contact your support center.
= PowerHA 7.1.3 and AIX 7.1 TL3 SP1 or AIX 6.1 TL9 SP1	YES	AIX and PowerHA fixes are required, depending on level. Please open a problem record to request fixes.
>= PowerHA 7.1.3 SP1 and AIX 7.1 TL3 SP3 or AIX 6.1 TL9 SP3	YES	Included.

If you want the *ifix* for a release before PowerHA 7.1.3 SP1 and IBM AIX 7.1 TL3 SP3 or AIX 6.1 TL9 SP3 become available, contact IBM Support and refer to the following APARs:

- PowerHA 7.1.3:
  – IV54588: CLMGR IMPROVES CAA REPOSITORY DISK RECOVERY
- AIX 6.1 TL9 SP1:
  – IV53637: RECOVER CLUSTER FROM CACHE FILE
- AIX 7.1. TL3 SP1:
  – IV53656: RECOVER CLUSTER FROM CACHE FILE
- AIX 7.1 TL3 SP2:
  – IV56563: RECOVER CLUSTER FROM CACHE FILE.

> **Note:** The APAR file sets must be installed *before* the outage occurs to use the new process.

The following section describes the steps require to start the service on the remaining node.

Recovering from a failure with PowerHA 7.1.3 and later

The cluster status on node 2 after the reboot is shown in Example A-4. No cluster service is available due to the missing repository disk.

Example A-4 Cluster status after the reboot

```
root@b2:/> lscluster -m
lscluster: Cluster services are not active.

root@b2:/> lssrc -ls clstrmgrES | grep state
Current state: ST_INIT

root@b2:/> clRGinfo
Cluster IPC error: The cluster manager on node b2 is in ST_INIT or NOT_CONFIGURED
state and cannot process the IPC request.
```

Example A-5 shows that the caavg_privat hdisk2 is missing.

Example A-5 lspv command output showing a missing hdisk2

```
root@b2:/> lspv
hdisk0          00f6f5d056bafee2                    rootvg          active
hdisk1          00f6f5d076cce945                    None
hdisk3          00f70c9976cc35af                    sasapp_vg
hdisk4          00f70c9976cc35e2                    sasapp_vg
hdisk5          00f70c9976cc3613                    None
hdisk6          00f70c9976cc3646                    None
root@b2:/> odmget HACMPsircol

HACMPsircol:
        name = "sas_itso_cl_sircol"
        id = 1
        uuid = "0"
```

```
        ip_address = ""
        repository = "00f70c9976cc355b"
        backup_repository = "00f70c9976cc3613"
```

In this state, the new support in the **clmgr** command is able to re-create the repository disk on a new disk. The new disk must fulfill the same requirements as the old one and cannot be part of another volume group in the LPAR. The **clmgr** command (shown in Example A-6) can be issued to re-create the repository disk on hdisk5.

Example A-6 Re-creatiing the repository with the clmgr command

```
root@b2:/> clmgr replace repository hdisk5
root@b2:/>
```

> **Note:** The replacement of the repository disk might take a while, depending on the cluster and LPAR configuration.

Subsequently, the repository disk is available and the CAA cluster starts on cluster node b2, as Example A-7 shows.

Example A-7 Repository disk is changed to hdisk5 and the CAA cluster is available

```
root@b2:/> lspv
hdisk0          00f6f5d056bafee2                    rootvg          active
hdisk1          00f6f5d076cce945                    None
hdisk3          00f70c9976cc35af                    sasapp_vg
hdisk4          00f70c9976cc35e2                    sasapp_vg
hdisk5          00f70c9976cc3613                    caavg_private   active
hdisk6          00f70c9976cc3646                    None
root@b2:/> lscluster -i
Network/Storage Interface Query

Cluster Name: sas_itso_cl
Cluster UUID: 5741f7c0-52ef-11e3-abd9-7a40c9ce2704
Number of nodes reporting = 1
Number of nodes stale = 1
Number of nodes expected = 1

Node b2
Node UUID = 573ae868-52ef-11e3-abd9-7a40c9ce2704
Number of interfaces discovered = 3
        Interface number 1, en0
                IFNET type = 6 (IFT_ETHER)
                NDD type = 7 (NDD_ISO88023)
                MAC address length = 6
                MAC address = EE:AF:09:B0:26:02
                Smoothed RTT across interface = 0
                Mean deviation in network RTT across interface = 0
                Probe interval for interface = 990 ms
                IFNET flags for interface = 0x1E084863
                NDD flags for interface = 0x0021081B
                Interface state = UP
                Number of regular addresses configured on interface = 1
                IPv4 ADDRESS: 192.168.100.85 broadcast 192.168.103.255 netmask
255.255.252.0
```

```
                Number of cluster multicast addresses configured on interface = 1
                IPv4 MULTICAST ADDRESS: 228.168.100.75
        Interface number 2, sfwcom
                IFNET type = 0 (none)
                NDD type = 304 (NDD_SANCOMM)
                Smoothed RTT across interface = 0
                Mean deviation in network RTT across interface = 0
                Probe interval for interface = 990 ms
                IFNET flags for interface = 0x00000000
                NDD flags for interface = 0x00000009
                Interface state = UP
        Interface number 3, dpcom
                IFNET type = 0 (none)
                NDD type = 305 (NDD_PINGCOMM)
                Smoothed RTT across interface = 750
                Mean deviation in network RTT across interface = 1500
                Probe interval for interface = 22500 ms
                IFNET flags for interface = 0x00000000
                NDD flags for interface = 0x00000009
                Interface state = UP RESTRICTED AIX_CONTROLLED

Node a2
Node UUID = 573ae80e-52ef-11e3-abd9-7a40c9ce2704
Number of interfaces discovered = 0
```

PowerHA can now be started to make the service available again by using **smitty clstart** from the smitty menu or from the command line, using **clmgr online node b2 WHEN=now**.

Note: Keep in mind that, at this point, you have neither fixed the whole problem nor synchronized the cluster. After the power is up again in Datacenter 1, node 1 of the cluster will start with the old repository disk. You must clean up this situation before starting the service on cluster node 1.

Reintegrating a failed node

The procedure to clean up a stretched cluster with different repository disks after a repository disk replacement on one node is documented in APAR IV50788, “DOC HA 7.1 HOW TO HANDLE SIMULTANEOUSLY REP DISK AND NODE FAILURE.”

```
PowerHA APAR IV50788
APAR status
Closed as documentation error.
```

Error description

In the case of a simultaneous node and repository disk failure (for example, when a data center fails), it might be necessary to replace the repository disk before all nodes are up again.

Procedure to replace the repository disk

To replace the repository disk, select **smitty sysmirror** → **Problem Determination Tools** → **Replace the Primary Repository Disk**.

A node that is down while the repository disk is replaced continues to access the original repository disk after reboot.

If the original repository disk is available again, the CAA cluster services start using this disk. The node remains in the DOWN state. The **lscluster -m** output is shown in Example A-8.

Example A-8 lscluster -m command output

```
Calling node query for all nodes...
Node query number of nodes examined: 2
      Node name: ha1clA
      Cluster shorthand id for node: 1
      UUID for node: 1ab63438-d7ed-11e2-91ce-46fc4000a002
      State of node: DOWN  NODE_LOCAL
      ...

-----------------------------------------------------
      Node name: ha2clA
      Cluster shorthand id for node: 2
      UUID for node: 1ac309e2-d7ed-11e2-91ce-46fc4000a002
      State of node: UP
      ...
      Points of contact for node: 2
      ------------------------------------------
      Interface     State  Protocol    Status
      ------------------------------------------
      en0           UP     IPv4         none
      en1           UP     IPv4         none
```

To force a previously failed node to use the *new* repository disk, run these commands on the affected node:

```
$ export CAA_FORCE_ENABLED=true
$ clusterconf -fu
```

Use the **lscluster -c** command to verify that the CAA cluster services are inactive.

Wait up to 10 minutes for the node to join the CAA cluster again, using the *new* repository disk.

Execute the **lscluster -c** and **lscluster -m** commands to verify that the CAA has restarted.

Before restarting PowerHA on the affected node, the PowerHA configuration needs to be synchronized. The synchronization needs to be started at a node that was up while the repository disk was replaced. Select **smitty sysmirror** → **Cluster Nodes and Networks** → **Verify and Synchronize Cluster Configuration**.

If there are multiple nodes available to do so and PowerHA is not up and running on all of them, choose an active node to start the synchronization.

Afterward, it is possible to restart PowerHA at the previously failed node by selecting **smitty sysmirror** → **System Management (C-SPOC)** → **PowerHA SystemMirror Services** → **Start Cluster Services**.

The sequence to correct the repository disk mismatch for the cluster nodes is described in the workflow depicted in Figure A-3.

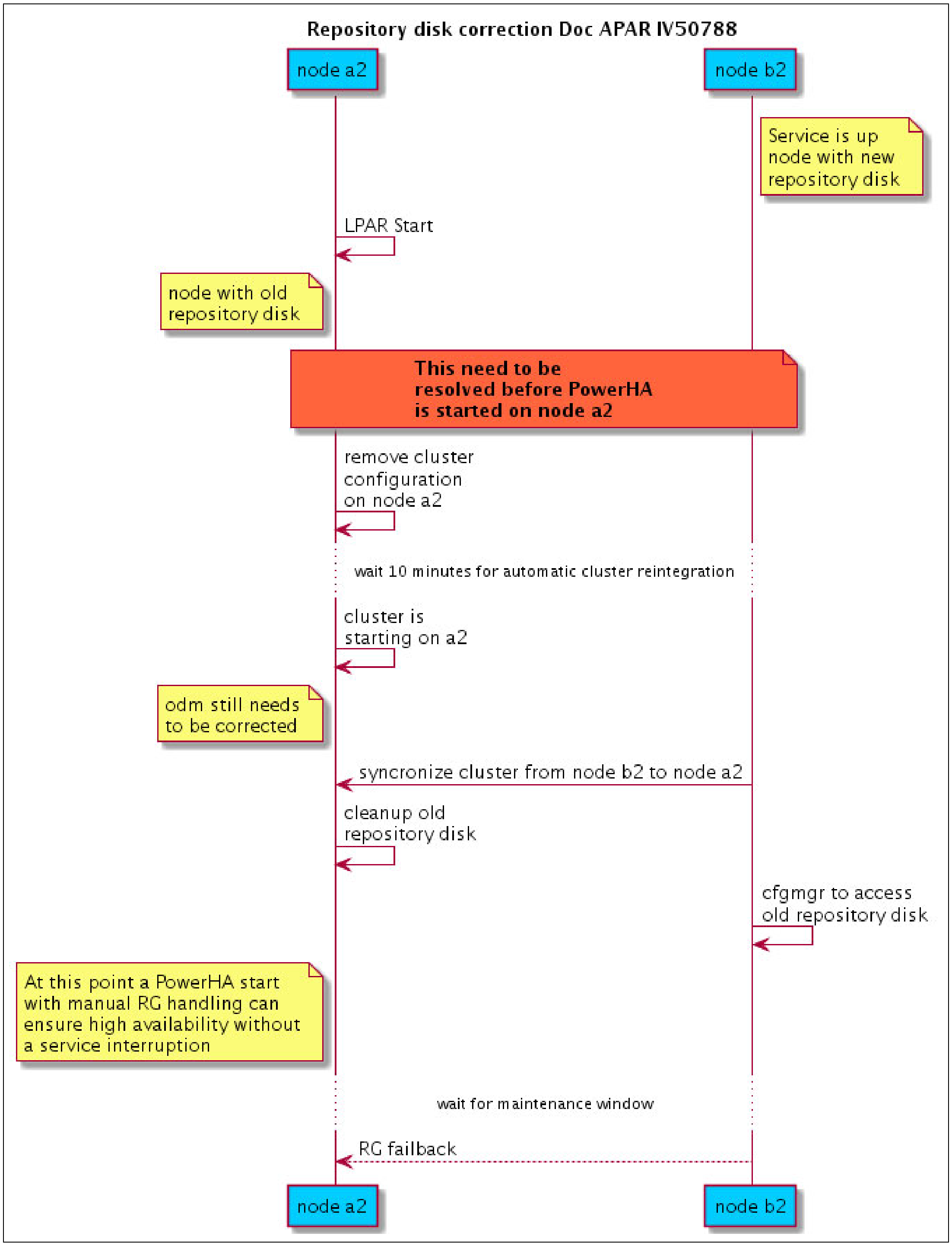


Figure A-3 Flowchart of how to correct the repository disk mismatch

Example A-9 through Example A-16 on page 414 show the correction of the repository mismatch between node a2 and node b2.

Example A-9 Cluster status after the start of LPAR a2

```
NODE a2 status is DOWN but RSCT is running
root@a2:/> ps -ef | grep rsct
    root 4784164 1507776   0 03:34:37      -  0:03
/usr/sbin/rsct/bin/IBM.ConfigRMd
    root 3211764 1507776   0 03:34:34      -  0:06 /usr/sbin/rsct/bin/rmcd -a
IBM.LPCommands -r -d all_but_msgs=4
    root 3408204 1507776   0 03:34:38      -  0:00 /usr/sbin/rsct/bin/IBM.DRMd
    root 3473726 1507776   0 03:34:38      -  0:00
/usr/sbin/rsct/bin/IBM.ServiceRMd

root@a2:/> lssrc -g rsct
Subsystem         Group            PID          Status
 ctrmc            rsct             3211764      active
 ctcas            rsct                          inoperative

root@a2:/> lscluster -m
Calling node query for all nodes...
Node query number of nodes examined: 2

        Node name: a2
        Cluster shorthand id for node: 2
        UUID for node: 573ae80e-52ef-11e3-abd9-7a40c9ce2704
        State of node: DOWN  NODE_LOCAL
        Smoothed rtt to node: 0
        Mean Deviation in network rtt to node: 0
        Number of clusters node is a member in: 1
        CLUSTER NAME       SHID         UUID
        sas_itso_cl        0            5741f7c0-52ef-11e3-abd9-7a40c9ce2704
        SITE NAME          SHID         UUID
        LOCAL              1            51735173-5173-5173-5173-517351735173

        Points of contact for node: 0
----------------------------------------------------------------------------

        Node name: b2
        Cluster shorthand id for node: 3
        UUID for node: 573ae868-52ef-11e3-abd9-7a40c9ce2704
        State of node: UP
        Smoothed rtt to node: 25
        Mean Deviation in network rtt to node: 18
        Number of clusters node is a member in: 1
        CLUSTER NAME       SHID         UUID
        sas_itso_cl        0            5741f7c0-52ef-11e3-abd9-7a40c9ce2704
        SITE NAME          SHID         UUID
        LOCAL              1            51735173-5173-5173-5173-517351735173

        Points of contact for node: 1
        -----------------------------------------------------------------------
        Interface     State  Protocol    Status     SRC_IP->DST_IP
        -----------------------------------------------------------------------
```

```
        tcpsock->03   UP     IPv4         none   192.168.100.75->192.168.100.202
```

Example A-10 shows node a2 still pointing to the old repository disk.

Example A-10 Node a2 still points to the old repository disk

```
root@a2:/> lspv
hdisk0          00f70c99540419ff                    rootvg          active
hdisk1          00f70c9975f30ff1                    None
hdisk2          00f70c9976cc355b                    caavg_private   active
hdisk3          00f70c9976cc35af                    sasapp_vg
hdisk4          00f70c9976cc35e2                    sasapp_vg
hdisk5          00f70c9976cc3613                    None
hdisk6          00f70c9976cc3646                    None
```

Example A-11 shows how to remove the old cluster configuration.

Example A-11 Removing the old cluster configuration

```
root@a2:/> export CAA _FORCE _ENABLED=true

root@a2:/> clusterconf -fu

root@a2:/> echo $?
3
root@a2:/> lscluster -c
lscluster: Cluster services are not active.
```

If you know which of the disks is the new repository disk, you can issue the command **clusterconf -r <hdiskx>** as shown in Example A-12. If no command is issued, the node waits up to 600 seconds to automatically join cluster.

Example A-12 Issuing the command to use the repository disk if known

```
root@a2:/> clusterconf -r hdisk5
root@a2:/>
```

Example A-13 shows the cluster configuration after the node joints the cluster again.

Example A-13 Cluster configuration after the node joint the cluster again

```
root@a2:/> lscluster -m
Calling node query for all nodes...
Node query number of nodes examined: 2

        Node name: a2
        Cluster shorthand id for node: 2
        UUID for node: 573ae80e-52ef-11e3-abd9-7a40c9ce2704
        State of node: UP  NODE_LOCAL
        Smoothed rtt to node: 0
        Mean Deviation in network rtt to node: 0
        Number of clusters node is a member in: 1
        CLUSTER NAME       SHID         UUID
        sas_itso_cl        0            5741f7c0-52ef-11e3-abd9-7a40c9ce2704
        SITE NAME          SHID         UUID
        LOCAL              1            51735173-5173-5173-5173-517351735173
```

```
        Points of contact for node: 0

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

        Node name: b2
        Cluster shorthand id for node: 3
        UUID for node: 573ae868-52ef-11e3-abd9-7a40c9ce2704
        State of node: UP
        Smoothed rtt to node: 11
        Mean Deviation in network rtt to node: 8
        Number of clusters node is a member in: 1
        CLUSTER NAME       SHID         UUID
        sas_itso_cl        0            5741f7c0-52ef-11e3-abd9-7a40c9ce2704
        SITE NAME          SHID         UUID
        LOCAL              1            51735173-5173-5173-5173-517351735173

        Points of contact for node: 1
        ------------------------------------------------------------------------
        Interface     State  Protocol    Status     SRC_IP->DST_IP
        ------------------------------------------------------------------------
        tcpsock->03   UP     IPv4         none   192.168.100.75->192.168.100.202
```

The ODM still has the old entries that need to be corrected as shown in Example A-14.

Example A-14 Incorrect ODM data

```
root@a2:/> odmget HACMPsircol

HACMPsircol:
        name = "sas_itso_cl_sircol"
        id = 1
        uuid = "0"
        ip_address = ""
        repository = "00f70c9976cc355b"
        backup_repository = "00f70c9976cc3613"

root@a2:/> lspv
hdisk0          00f70c99540419ff                    rootvg          active
hdisk1          00f70c9975f30ff1                    None
hdisk2          00f70c9976cc355b                    None
hdisk3          00f70c9976cc35af                    sasapp_vg
hdisk4          00f70c9976cc35e2                    sasapp_vg
hdisk5          00f70c9976cc3613                    caavg_private   active
hdisk6          00f70c9976cc3646                    None
```

Before restarting PowerHA on the affected node, the PowerHA configuration needs to be synchronized. The synchronization needs to be started at the node that was up while the repository disk was replaced. Switch to node b2, and sync the cluster as shown in Example A-15.

Example A-15 sync the cluster

```
root@b2:/> /usr/es/sbin/cluster/utilities/cldare -tr
Timer object autoclverify already exists

Verification to be performed on the following:
```

```
        Cluster Topology
        Cluster Resources
.........
NO Errors
```

Example A-16 shows how node b2 discovers the old repository disk.

Example A-16 Run configuration manager (cfgmgr) on node b2 to discover old repository disk

```
root@b2:/> lspv
hdisk0          00f6f5d056bafee2                    rootvg          active
hdisk1          00f6f5d076cce945                    None
hdisk3          00f70c9976cc35af                    sasapp_vg       concurrent
hdisk4          00f70c9976cc35e2                    sasapp_vg       concurrent
hdisk5          00f70c9976cc3613                    caavg_private   active
hdisk6          00f70c9976cc3646                    None

root@b2:/> cfgmgr

root@b2:/> lspv
hdisk0          00f6f5d056bafee2                    rootvg          active
hdisk1          00f6f5d076cce945                    None
hdisk2          00f70c9976cc355b                    None
hdisk3          00f70c9976cc35af                    sasapp_vg       concurrent
hdisk4          00f70c9976cc35e2                    sasapp_vg       concurrent
hdisk5          00f70c9976cc3613                    caavg_private   active
hdisk6          00f70c9976cc3646                    None
```

B

Custom monitoring scripts

This appendix provides custom monitoring scripts and includes the following sections:

- Custom monitoring query script example 1: # qha
- Custom monitoring query script example 2: # qha_remote
- Custom monitoring query script example 3: # qha_rmc
- Custom monitoring query script example 4: # liveHA
- PowerHA MIB file
- Tivoli Monitoring Universal Agent metafile for PowerHA
- Tivoli Monitoring Universal Agent TRAPCNFG for PowerHA SNMP monitoring

Custom monitoring query script example 1: `# qha`

Example B-1 shows the query HA custom monitoring script.

Example B-1 Query HA custom monitoring script

```
#!/bin/ksh

# Purpose: Provides an alternative to SNMP monitoring for PowerHA/HACMP (clinfo
and clstat).
#          Designed to be run within the cluster, not remotely. See next point!
#          Can be customised to run remotely and monitor multiple clusters!
# Version: 9.06
#          Updates for PowerHA version 7.1
# Authors: 1. Alex Abderrazag IBM UK
#  #       2. Bill Miller IBM US
#            Additions since 8.14.

# qha can be freely distributed.  If you have any questions or would like to see
any enhancements/updates, please email abderra@uk.ibm.com

# VARS

export PATH=$PATH:/usr/es/sbin/cluster/utilities
VERSION=`lslpp -L |grep -i cluster.es.server.rte |awk '{print $2}'| sed 's/\.//g'`
CLUSTER=`odmget HACMPcluster | grep -v node |grep name | awk '{print $3}' |sed
"s:\"::g"`
UTILDIR=/usr/es/sbin/cluster/utilities
# clrsh dir in v7 must be /usr/sbin in previous version's it's
/usr/es/sbin/cluster/utilities.
# Don't forget also that the rhost file for >v7 is /etc/cluster/rhosts
if [[ `lslpp -L |grep -i cluster.es.server.rte |awk '{print $2}' | cut -d'.' -f1`
-ge 7 ]]; then
  CDIR=/usr/sbin
else
  CDIR=$UTILDIR
fi
OUTFILE=/tmp/.qha.$$
LOGGING=/tmp/qha.out.$$
ADFILE=/tmp/.ad.$$
HACMPOUT=`/usr/bin/odmget -q name="hacmp.out" HACMPlogs | fgrep value | sed
's/.*=\ "\(.*\)"$/\1\/hacmp.out/'`
COMMcmd="$CDIR/clrsh"
REFRESH=0

usage()
{
       echo "qha version 9.06"
         echo "Usage: qha [-n] [-N] [-v] [-l] [-e] [-m] [-1] [-c]"
       echo "\t\t-n displays network interfaces\n\t\t-N displays network
interfaces + nonIP heartbeat disk\n\t\t-v shows online VGs\n\t\t-l logs entries to
/tmp/qha.out\n\t\t-e shows running event\n\t\t-m shows appmon status\n\t\t-1
single interation\n\t\t-c shows CAA SAN/Disk Status (AIX7.1 TL3 min.)"
}
```

```
function adapters
{
i=1
j=1
cat $ADFILE | while read line
do
  en[i]=`echo $line | awk '{print $1}'`
  name[i]=`echo $line | awk '{print $2}'`
  if [ i -eq 1 ];then printf "   ${en[1]} "; fi
  if [[ ${en[i]} = ${en[j]} ]]
  then
       printf "${name[i]} "
  else
       printf "\n${en[i]} ${name[i]} "
  fi
  let i=i+1
  let j=i-1
done
rm $ADFILE

if [ $HBOD = "TRUE" ]; then # Code for v6 and below only. To be deleted soon.
 # Process Heartbeat on Disk networks (Bill Millers code)
 VER=`echo $VERSION | cut -c 1`
 if [[ $VER = "7" ]]; then
       print "[HBOD option not supported]" >> $OUTFILE
 fi
 HBODs=$($COMMcmd $HANODE "$UTILDIR/cllsif" | grep diskhb | grep -w $HANODE | awk 
'{print $8}')
 for i in $(print $HBODs)
 do
  APVID=$($COMMcmd $HANODE "lspv" | grep -w $i | awk '{print $2}' | cut -c 13-)
  AHBOD=$($COMMcmd $HANODE lssrc -ls topsvcs | grep -w r$i | awk '{print $4}')
  if [ $AHBOD ]
  then
    printf "\n\t%-13s %-10s" $i"("$APVID")" [activeHBOD]
  else
    printf "\n\t%-13s %-10s" $i [inactiveHBOD]
  fi
 done
fi
}

function work
{
HANODE=$1; CNT=$2 NET=$3 VGP=$4
#clrsh $HANODE date > /dev/null 2>&1 || ping -w 1 -c1 $HANODE > /dev/null 2>&1
$COMMcmd $HANODE date > /dev/null 2>&1
if [ $? -eq 0 ]; then
 EVENT="";
 CLSTRMGR=`$COMMcmd $HANODE lssrc -ls clstrmgrES | grep -i state | sed 's/Current 
state: //g'`
 if [[ $CLSTRMGR != ST_STABLE && $CLSTRMGR != ST_INIT && $SHOWEVENT = TRUE ]]; 
then
   EVENT=$($COMMcmd $HANODE cat $HACMPOUT | grep "EVENT START" |tail -1 | awk 
'{print $6}')
```

```
   printf "\n%-8s %-7s %-15s\n" $HANODE iState: "$CLSTRMGR [$EVENT]"
 else
  printf "\n%-8s %-7s %-15s\n" $HANODE iState: "$CLSTRMGR"
 fi
 $UTILDIR/clfindres -s 2>/dev/null |grep -v OFFLINE | while read A
        do
        if [[ "`echo $A | awk -F: '{print $3}'`" == "$HANODE" ]];
       then
         echo $A | awk -F: '{printf "   %-18.16s %-10.12s %-1.20s", $1, $2, $9}'
         if [ $APPMONSTAT = "TRUE" ]; then
            RG=`echo $A | awk -F':' '{print $1}'`
            APPMON=`$UTILDIR/clRGinfo -m | grep -p $RG | grep "ONLINE" | awk
'NR>1' | awk '{print $1" "$2}`
            print "($APPMON)"
         else
            print ""
         fi
       fi
        done
 if [ $CAA = "TRUE" ]; then
  IP_Comm_method=`odmget HACMPcluster | grep heartbeattype | awk -F'"' '{print
$2}'`
  case $IP_Comm_method in
       C) # we're multicasting
       printf "   CAA Multicasting:"
       $COMMcmd $HANODE lscluster -m | grep en[0-9] | awk '{printf " ("$1"
"$2")"}'
       echo ""
       ;;
       U) # we're unicasting
       printf "   CAA Unicasting:"
       $COMMcmd $HANODE lscluster -m | grep tcpsock | awk '{printf " ("$2" "$3"
"$5")"}'
       echo ""
       ;;
  esac

  SAN_COMMS_STATUS=$(/usr/lib/cluster/clras sancomm_status | egrep -v "(--|UUID)"
| awk -F'|' '{print $4}' | sed 's/ //g')
  DP_COMM_STATUS=$(/usr/lib/cluster/clras dpcomm_status | grep $HANODE | awk -F'|'
'{print $4}' | sed 's/ //g')
  print "   CAA SAN Comms: $SAN_COMMS_STATUS | DISK Comms: $DP_COMM_STATUS"
 fi


 if [ $NET = "TRUE" ]; then
  $COMMcmd $HANODE netstat -i | egrep -v "(Name|link|lo)" | awk '{print $1" "$4"
"}' > $ADFILE
  adapters; printf "\n- "
 fi
 if [ $VGP = "TRUE" ]; then
   VGO=`$COMMcmd $HANODE "lsvg -o |fgrep -v caavg_private |fgrep -v rootvg |lsvg
-pi 2> /dev/null" |awk '{printf $1")"}' |sed 's:)PV_NAME)hdisk::g' | sed 's/:/(/g'
|sed 's:):) :g' |sed 's: hdisk:(:g' 2> /dev/null`
   if [ $NET = "TRUE" ]; then
```

```
    echo "$VGO-"
  else
    echo "- $VGO-"
  fi
 fi
else
 ping -w 1 -c1 $HANODE > /dev/null 2>&1
 if [ $? -eq 0 ]; then
      echo "\nPing to $HANODE good, but can't get the status. Check clcomdES."
 else
      echo "\n$HANODE not responding, check network availability."
 fi
fi
}

# Main
NETWORK="FALSE"; VG="FALSE"; HBOD="FALSE"; LOG=false; APPMONSTAT="FALSE"; STOP=0;
CAA=FALSE; REMOTE="FALSE";
# Get Vars
while getopts :nNvlem1c ARGs
do
               case $ARGs in
               n)     # -n show interface info
                  NETWORK="TRUE";;
              N)     # -N show interface info and activeHBOD
                        NETWORK="TRUE"; HBOD="TRUE";;
               v)     # -v show ONLINE VG info
                  VG="TRUE";;
               l)     # -l log to /tmp/qha.out
                  LOG="TRUE";;
              e)     # -e show running events if cluster is unstable
                        SHOWEVENT="TRUE";;
              m)     # -m show status of monitor app servers if present
                        APPMONSTAT="TRUE";;
              1)     # -1 exit after first iteration
                   STOP=1;;
              c)     # CAA SAN / DISK Comms
                  CAA=TRUE;;
              \?) printf "\nNot a valid option\n\n" ; usage ; exit ;;
               esac
done

OO=""
trap "rm $OUTFILE; exit 0" 1 2 12 9 15
while true
do
COUNT=0
 print "\\033[H\\033[2J\t\tCluster: $CLUSTER ($VERSION)" > $OUTFILE
 echo "\t\t$(date +%T" "%d%b%y)" >> $OUTFILE
 if [[ $REMOTE = "TRUE" ]]; then
   Fstr=`cat $CLHOSTS |grep -v "^#"`
 else
   Fstr=`odmget HACMPnode |grep name |sort -u | awk '{print $3}' |sed "s:\"::g"`
 fi
 for MAC in `echo $Fstr`
```

```
 do
 let COUNT=COUNT+1
       work $MAC $COUNT $NETWORK $VG $HBOD
 done >> $OUTFILE

 cat $OUTFILE
 if [ $LOG = "TRUE" ]; then
   wLINE=$(cat $OUTFILE |sed s'/^.*Cluster://g' | awk '{print " "$0}' |tr -s
'[:space:]' '[ *]' | awk '{print $0}')
   wLINE_three=$(echo $wLINE | awk '{for(i=4;i<=NF;++i) printf("%s ",  $i) }')
   if [[ ! "$00" = "$wLINE_three" ]]; then
       # Note, there's been a state change, so write to the log
       # Alternatively, do something addtional, for example: send an snmp trap
alert, using the snmptrap command. For example:
       # snmptrap -c <community> -h <anmp agent> -m "appropriate message"
       echo "$wLINE" >> $LOGGING
   fi
   OO="$wLINE_three"
 fi
 if [[ STOP -eq 1 ]]; then
       exit
 fi
 sleep $REFRESH
done
```

Custom monitoring query script example 2: # qha_remote

Example B-2 shows the query HA custom monitoring script modified to run outside the cluster.

Example B-2 Query HA modified to run outside of the cluster

```
#!/bin/ksh

# Purpose: Query HA (qha) modified to run remotely outside of the cluster
# Based on qha version: 9.06
#          Updates for PowerHA version 7.1
# Authors: 1. Alex Abderrazag IBM UK
#           2. Bill Miller IBM US
#             Additions since 8.14.

# qha_remote can be freely distributed.  If you have any questions or would like
to see any enhancements/updates, please email abderra@uk.ibm.com

# VARS

export PATH=$PATH:/usr/es/sbin/cluster/utilities
CLHOSTS="/alex/clhosts"
UTILDIR=/usr/es/sbin/cluster/utilities
# clrsh dir in v7 must be /usr/sbin in previous version's it's
/usr/es/sbin/cluster/utilities.
# Don't forget also that the rhost file for >v7 is /etc/cluster/rhosts
OUTFILE=/tmp/.qha.$$
```

```
LOGGING=/tmp/qha.out.$$
ADFILE=/tmp/.ad.$$
HACMPOUT=/var/hacmp/log/hacmp.out
COMMcmd="ssh -o ConnectTimeout=3 -o ServerAliveInterval=3"
REFRESH=0

usage()
{
        echo "Usage: qha [-n] [-N] [-v] [-l] [-e] [-m] [-1] [-c]"
       echo "\t\t-n displays network interfaces\n\t\t-N displays network
interfaces + nonIP heartbeat disk\n\t\t-v shows online VGs\n\t\t-l logs entries to
/tmp/qha.out\n\t\t-e shows running event\n\t\t-m shows appmon status\n\t\t-1
single interation\n\t\t-c shows CAA SAN/Disk Status (AIX7.1 TL3 min.)"
}

function adapters
{
i=1
j=1
cat $ADFILE | while read line
do
  en[i]=`echo $line | awk '{print $1}'`
  name[i]=`echo $line | awk '{print $2}'`
  if [ i -eq 1 ];then printf "   ${en[1]} "; fi
  if [[ ${en[i]} = ${en[j]} ]]
  then
       printf "${name[i]} "
  else
       printf "\n${en[i]} ${name[i]} "
  fi
  let i=i+1
  let j=i-1
done
rm $ADFILE

if [ $HBOD = "TRUE" ]; then # Code for v6 and below only. To be deleted soon.
 # Process Heartbeat on Disk networks (Bill Millers code)
 VER=`echo $VERSION | cut -c 1`
 if [[ $VER = "7" ]]; then
       print "[HBOD option not supported]" >> $OUTFILE
 fi
 HBODs=$($COMMcmd $HANODE "$UTILDIR/cllsif" | grep diskhb | grep -w $HANODE | awk
'{print $8}')
 for i in $(print $HBODs)
 do
  APVID=$($COMMcmd $HANODE "lspv" | grep -w $i | awk '{print $2}' | cut -c 13-)
  AHBOD=$($COMMcmd $HANODE lssrc -ls topsvcs | grep -w r$i | awk '{print $4}')
  if [ $AHBOD ]
  then
    printf "\n\t%-13s %-10s" $i"("$APVID")" [activeHBOD]
  else
    printf "\n\t%-13s %-10s" $i [inactiveHBOD]
  fi
 done
fi
```

```
}

function initialise
{
if [[ -n $CLUSTER ]]; then return; fi
echo "Initialising..."
HANODE=$1;
$COMMcmd $HANODE date > /dev/null 2>&1
if [ $? -eq 0 ]; then
 CLUSTER=`$COMMcmd $HANODE odmget HACMPcluster | grep -v node |grep name | awk
'{print $3}' |sed "s:\"::g"`
 VERSION=`$COMMcmd $HANODE lslpp -L |grep -i cluster.es.server.rte |awk '{print
$2}'| sed 's/\.//g'`
fi
}

function work
{
HANODE=$1; CNT=$2 NET=$3 VGP=$4
#clrsh $HANODE date > /dev/null 2>&1 || ping -w 1 -c1 $HANODE > /dev/null 2>&1
$COMMcmd $HANODE date > /dev/null 2>&1
if [ $? -eq 0 ]; then
 EVENT="";
 CLSTRMGR=`$COMMcmd $HANODE lssrc -ls clstrmgrES | grep -i state | sed 's/Current
state: //g'`
 if [[ $CLSTRMGR != ST_STABLE && $CLSTRMGR != ST_INIT && $SHOWEVENT = TRUE ]];
then
   EVENT=$($COMMcmd $HANODE cat $HACMPOUT | grep "EVENT START" |tail -1 | awk
'{print $6}')
   printf "\n%-8s %-7s %-15s\n" $HANODE iState: "$CLSTRMGR [$EVENT]"
 else
  printf "\n%-8s %-7s %-15s\n" $HANODE iState: "$CLSTRMGR"
 fi

 # RG status
 if [[ $APPMONSTAT = "TRUE" ]]; then
  $COMMcmd $HANODE "
  $UTILDIR/clfindres -s 2>/dev/null | grep ONLINE | grep $HANODE | awk -F':'
'{print \$1}' | while read RG
  do
   $UTILDIR/clfindres -m "\$RG" 2>/dev/null | egrep -v '(---|Group Name)' | sed
's/  */ /g' | sed '/^$/d' | awk '{printf}'
   echo ""
  done
  " | awk '{printf "  "$1"\t"$2" ("; for (i=4; i <= NF; i++) printf FS$i; print
")" }' | sed 's/( /(/g'
 else
  $COMMcmd $HANODE $UTILDIR/clfindres -s 2>/dev/null |grep -v OFFLINE | while read
A
        do
        if [[ "`echo $A | awk -F: '{print $3}'`" == "$HANODE" ]];
        then
          echo $A | awk -F: '{printf "   %-18.16s %-10.12s %-1.20s\n", $1, $2,
$9}'
        fi
```

```
        done
 fi
 # End RG status

 if [ $CAA = "TRUE" ]; then
  IP_Comm_method=`$COMMcmd $HANODE odmget HACMPcluster | grep heartbeattype | awk
-F'"' '{print $2}'`
  case $IP_Comm_method in
       C) # we're multicasting
       printf "   CAA Multicasting:"
       $COMMcmd $HANODE lscluster -m | grep en[0-9] | awk '{printf " ("$1"
"$2")"}'
       echo ""
       ;;
       U) # we're unicasting
       printf "   CAA Unicasting:"
       $COMMcmd $HANODE lscluster -m | grep tcpsock | awk '{printf " ("$2" "$3"
"$5")"}'
       echo ""
       ;;
  esac

  SAN_COMMS_STATUS=$($COMMcmd $HANODE /usr/lib/cluster/clras sancomm_status |
egrep -v "(--|UUID)" | awk -F'|' '{print $4}' | sed 's/ //g')
  DP_COMM_STATUS=$($COMMcmd $HANODE /usr/lib/cluster/clras dpcomm_status | grep
$HANODE | awk -F'|' '{print $4}' | sed 's/ //g')
  print "   CAA SAN Comms: $SAN_COMMS_STATUS | DISK Comms: $DP_COMM_STATUS"
 fi


 if [ $NET = "TRUE" ]; then
  $COMMcmd $HANODE netstat -i | egrep -v "(Name|link|lo)" | awk '{print $1" "$4"
"}' > $ADFILE
  adapters; printf "\n- "
 fi
 if [ $VGP = "TRUE" ]; then
   VGO=`$COMMcmd $HANODE "lsvg -o |fgrep -v caavg_private |fgrep -v rootvg |lsvg
-pi 2> /dev/null" |awk '{printf $1")"}' |sed 's:)PV_NAME)hdisk::g' | sed 's/:/(/g'
|sed 's:):) :g' |sed 's: hdisk:(:g' 2> /dev/null`
   if [ $NET = "TRUE" ]; then
     echo "$VGO-"
   else
     echo "- $VGO-"
   fi
 fi
else
 ping -w 1 -c1 $HANODE > /dev/null 2>&1
 if [ $? -eq 0 ]; then
       echo "\nPing to $HANODE good, but can't get the status. Check clcomdES."
 else
       echo "\n$HANODE not responding, check network availability."
 fi
fi
}
```

```
# Main
NETWORK="FALSE"; VG="FALSE"; HBOD="FALSE"; LOG=false; APPMONSTAT="FALSE"; STOP=0;
CAA=FALSE; REMOTE="FALSE";
# Get Vars
while getopts :nNvlem1c ARGs
do
               case $ARGs in
               n)     # -n show interface info
                  NETWORK="TRUE";;
              N)     # -N show interface info and activeHBOD
                        NETWORK="TRUE"; HBOD="TRUE";;
               v)     # -v show ONLINE VG info
                  VG="TRUE";;
               l)     # -l log to /tmp/qha.out
                  LOG="TRUE";;
              e)     # -e show running events if cluster is unstable
                        SHOWEVENT="TRUE";;
              m)     # -m show status of monitor app servers if present
                        APPMONSTAT="TRUE";;
              1)     # -1 exit after first iteration
                   STOP=1;;
              c)     # CAA SAN / DISK Comms
                  CAA=TRUE;;
              \?) printf "\nNot a valid option\n\n" ; usage ; exit ;;
               esac
done

COUNT=0; OO=""
trap "rm $OUTFILE; exit 0" 1 2 12 9 15
while true
do
 Fstr=`cat $CLHOSTS |grep -v "^#"`
 if [[ COUNT -eq 0 ]]; then
  for MAC in `echo $Fstr`; do
    initialise $MAC
  done
 fi
 print "\\033[H\\033[2J\t\tCluster: $CLUSTER ($VERSION)" > $OUTFILE
 echo "\t\t$(date +%T" "%d%b%y)" >> $OUTFILE
 for MAC in `echo $Fstr`
 do
 let COUNT=COUNT+1
       work $MAC $COUNT $NETWORK $VG $HBOD
 done >> $OUTFILE

 cat $OUTFILE
 if [ $LOG = "TRUE" ]; then
   wLINE=$(cat $OUTFILE |sed s'/^.*Cluster://g' | awk '{print " "$0}' |tr -s
'[:space:]' '[ *]' | awk '{print $0}')
   wLINE_three=$(echo $wLINE | awk '{for(i=4;i<=NF;++i) printf("%s ",  $i) }')
   if [[ ! "$OO" = "$wLINE_three" ]]; then
       # Note, there's been a state change, so write to the log
       # Alternatively, do something addtional, for example: send an snmp trap
alert, using the snmptrap command. For example:
       # snmptrap -c <community> -h <anmp agent> -m "appropriate message"
```

```
        echo "$wLINE" >> $LOGGING
    fi
    OO="$wLINE_three"
  fi
  if [[ STOP -eq 1 ]]; then
        exit
  fi
  sleep $REFRESH
done
```

Custom monitoring query script example 3: # qha_rmc

Example B-3 shows the query HA remote multi cluster custom monitoring script.

Example B-3   Query HA remote multi cluster

```
#!/bin/ksh
###############################################################################
# Purpose:  To provide an instant cluster status overview for multiple
# clusters (no limit). The output is really intended to be viewed from a web
# browser and provides an ideal view for first line support personnel.
#
# Description: This tool is designed to remotely monitor multiple
# clusters.  It will display the internal state of each cluster mgr,
# plus the state of each RG as reported by clRGinfo
# Note: Unprompted ssh access must be configured to each cluster node
#
# You must first create a CLUSTERfile, see vars. Format of the file is:
# cluster:<clustername>:node1 node2 node3 etc etc
# eg. cluster:matrix:neo trinity
#
# Optional but highly recommended, an apache or equivalent web server
#
# Version:  1.02
#
# Author:   Alex Abderrazag, IBM UK Ltd.
###############################################################################

##################VARS - can be changed #######################################
CLUSTERfile=/alex/QHAhosts
CGIPATH=/opt/freeware/apache/share/cgi-bin #Path to Web server cgi-bin
CGIFILE="$CGIPATH/qhar.cgi" #CGI file to be displayed in the web browser
OUTFILE=/tmp/.aastatus
CDIR=/usr/es/sbin/cluster/utilities
#VERSION=`lslpp -L |grep -i cluster.es.server.rte |awk '{print $2}'`
#CLUSTER=`ssh root@neo odmget HACMPcluster | grep -v node |grep name | awk '{print
$3}' |sed "s:\"::g"`
SLEEPTIME=2

usage()
{
        echo "\nUsage: qhar\n"
}
```

```
################################################################################
#
#  Name: format_cgi Create the cgi (on the fly!)
#
format_cgi()
{
if [ -f $CGIFILE ]; then rm $CGIFILE; fi
touch $CGIFILE
ex -s $CGIFILE <<EOF
a
#!/usr/bin/ksh
print "Content-type: text/html\n";

echo "<!DOCTYPE HTML>"
echo "<head>"
echo "<TITLE>Status of HACMP clusters</TITLE>"
print "<META HTTP-EQUIV="REFRESH" CONTENT="5">"
echo "</head>"
echo "<body>"

############################################################ Start table ...
echo "<table border="0" width="100%" cellpadding="10">"
echo "<tr>"

### Start section 1 ###
echo "<td width="50%" valign="top">Cluster Status Report on `date`"
cat << EOM1
<PRE style="line-height:16px">
<HR SIZE=2><font style="font-size:120%;color:black"><B>
EOM1
echo "</td>" ### End section

### Start section 2 ###
echo "<td width="50%" valign="top">By Alex Abderrazag (IBM UK)"
cat << EOM2
<PRE style="line-height:16px">
<HR SIZE=2><font style="font-size:120%;color:black"><B>
EOM2
echo "</td>" ### End section

############################################################ End table stuff
echo "</tr>"
echo "</table>"

echo "</body>"
echo "</html>"
.
wq
EOF
chmod 755 $CGIFILE
}

function work
{
```

```
HANODE=$1
ping -w 1 -c1 $HANODE > /dev/null 2>&1
if [ $? -eq 0 ]; then
i=1
ssh -o ConnectTimeout=3 -o ServerAliveInterval=3 root@$HANODE '

lssrc -ls clstrmgrES |grep -i state |sed 's:Current:$HANODE:g' |sed "s:state: :g"
/usr/es/sbin/cluster/utilities/clfindres -s 2>/dev/null |grep -v OFFLINE

' | while read A
        do
       if [ $i -eq 1 ]; then
       echo $A | awk -F: '{printf "Node: %-10s %-15s\n", $1, $2 $3}'
       let i=i+1
       else
        echo $A |egrep -i "(state|$HANODE)" | awk -F: '{printf "  %-15s %-20s
%-10s %-10s\n", $1, $2, $3, $9}'
       fi
        done
else
       echo "\nCan't get Status for $HANODE, check the network availability"
fi
}

# Main

format_cgi
rm $OUTFILE*
for clusternumber in `cat $CLUSTERfile | grep "^cluster:"  | cut -f2 -d:`
do
   NODES=`grep "^cluster:$clusternumber:" $CLUSTERfile | cut -f3 -d:`
       echo "\t\tCluster: $clusternumber " >> $OUTFILE.$clusternumber
       for MAC in $NODES
       do
             work $MAC
       done  >> $OUTFILE.$clusternumber  &
done
sleep $SLEEPTIME
# got to wait for jobs to be completed, this time may have to be tuned depending
on the no. of clusters
cat $OUTFILE*
# add the outfiles to the cgi
ONE=TRUE
for f in $OUTFILE*
do
# sed the file # hack as aix/sed does not have -i
cat $f | sed 's:ONLINE:<font color="#00FF00">ONLINE<font color="#000000">:g' \
| sed 's:ST_STABLE:<font color="#00FF00">UP \& STABLE<font color="#000000">:g' \
| sed 's:ERROR:<font color="#FF0000">ERROR!<font color="#000000">:g' \
| sed 's:ST_RP_FAILED:<font color="#FF0000">SCRIPT FAILURE!<font
color="#000000">:g' \
| sed 's:ST_INIT:<font color="#2B65EC">NODE DOWN<font color="#000000">:g' \
| sed 's:SECONDARY:<font color="#2B65EC">SECONDARY<font color="#000000">:g' \
| sed 's:ST_JOINING:<font color="#2B65EC">NODE JOINING<font color="#000000">:g' \
| sed 's:ST_VOTING:<font color="#2B65EC">CLUSTER VOTING<font color="#000000">:g' \
```

```
| sed 's:ST_RP_RUNNING:<font color="#2B65EC">SCRIPT PROCESSING<font
color="#000000">:g' \
| sed 's:ST_BARRIER:<font color="#2B65EC">BARRIER<font color="#000000">:g' \
| sed 's:ST_CBARRIER:<font color="#2B65EC">CBARRIER<font color="#000000">:g' \
| sed 's:ST_UNSTABLE:<font color="#FF0000">UNSTABLE<font color="#000000">:g' \
|tee $f
## end sed
   if [ $ONE = "TRUE" ]; then
ex -s $CGIFILE <<END
/^EOM1
a
cat $f
echo "------------------------------------------------------------"
.
wq
END
   ONE=FALSE
   else
ex -s $CGIFILE <<END
/^EOM2
a
cat $f
echo "------------------------------------------------------------"
.
wq
END
   ONE=TRUE
   fi
done > /dev/null 2>&1
```

Custom monitoring query script example 4: # liveHA

Example B-4 shows the live HA custom monitoring script.

Example B-4 Live HA

```
#!/bin/ksh
################################################################################
# Purpose:  To have a common secure local/remote cluster status monitor which
# does not use clinfo and provides a different degree of flexibility
# than clstat - in one tool
#
# Description: An 'clstat' alternative monitoring script. See Usage.
# Differences to clstat :
#      1/. Uses ssh rather clinfo. Unprompted ssh access must be configured
#         - prior to running this script
#      2/. Designed to be configurable by the end user
#      3/. Displays the internal cluster mgr state [-i]
#      4/. Cmd line script, produces both text std out and cgi
#         - for color display via web brower (refresh 5 secs)
#      5/. Output can be changed by to remove network/address information [-n]
#      6/. Can be run as a one off report [-1], will loop by default
```

```
#      7/. Displays the status of SAN communication [-c], requires HA v7 and AIX
7100-03-01 min
#      8/. Monitor's a single cluster
#         - future enhancements to follow..
#
# Version:  1.007
#
# Author:   Alex Abderrazag, IBM UK Ltd.
###############################################################################

usage()
{
    printf "Usage: $PROGNAME [-n] [-1] [-i]\n"
    printf "\t-n Omit Network info\n"
    printf "\t-1 Display 1 report rather than loop\n"
    printf "\t-i Displays the internal state of cluster manager\n"
    printf "\t-c Displays the state of SAN Communications\n"
    printf "Note: By default unprompted ssh must be configured from\n"
    printf "      the client monitor to each cluster node\n"
    exit 1
}

###############################################################################
#
#  Global VARs
#
###############################################################################

#*******************Please Alter the VARs below as appropriate*****

LOGFILE="/tmp/.qhaslog.$$" #General log file
HTMLFILE="/tmp/.qhashtml.$$" #HTML output file
CGIPATH=/opt/freeware/apache/share/cgi-bin #Path to Web server cgi-bin
CGIFILE="$CGIPATH/qhasA.cgi" #CGI file to be displayed in the web browser
CLHOSTS="/alex/clhosts" #Populate this file with the resolvable names of each
cluster node
USER=root # to be used for ssh access
SNMPCOMM=public #SNMP community name
SSHparams="-o ConnectTimeout=3 -o ServerAliveInterval=3"

#******ONLY alter the code below this line, if you want to change******
#********************this behaviour of this script********************

INTERNAL=0
PROGNAME=$(basename ${0})

#export PATH=$(/usr/es/sbin/cluster/utilities/cl_get_path all)

#HA_DIR="$(cl_get_path)"

# set up some global variables with SNMP branch info
# cluster
  CLUSTER_BRANCH="1.3.6.1.4.1.2.3.1.2.1.5.1"
  CLUSTER_NAME="$CLUSTER_BRANCH.2"
  CLUSTER_STATE="$CLUSTER_BRANCH.4"
```

```
  CLUSTER_SUBSTATE="$CLUSTER_BRANCH.8"
  CLUSTER_NUM_NODES="$CLUSTER_BRANCH.11"
# node
  NODE_BRANCH="1.3.6.1.4.1.2.3.1.2.1.5.2.1.1"
  NODE_ID="$NODE_BRANCH.1"
  NODE_STATE="$NODE_BRANCH.2"
  NODE_NUM_IF="$NODE_BRANCH.3"
  NODE_NAME="$NODE_BRANCH.4"
# network
  NETWORK_BRANCH="1.3.6.1.4.1.2.3.1.2.1.5.4.1.1"
  NETWORK_ID="$NETWORK_BRANCH.2"
  NETWORK_NAME="$NETWORK_BRANCH.3"
  NETWORK_ATTRIBUTE="$NETWORK_BRANCH.4"
  NETWORK_STATE="$NETWORK_BRANCH.5"
# address
  ADDRESS_BRANCH="1.3.6.1.4.1.2.3.1.2.1.5.3.1.1"
  ADDRESS_IP="$ADDRESS_BRANCH.2"
  ADDRESS_LABEL="$ADDRESS_BRANCH.3"
  ADDRESS_NET="$ADDRESS_BRANCH.5"
  ADDRESS_STATE="$ADDRESS_BRANCH.6"
  ADDRESS_ACTIVE_NODE="$ADDRESS_BRANCH.7"

################################################################################
#
#  Name: format_cgi
#
#  Create the cgi (on the fly!)
#
################################################################################
format_cgi()
{
if [ -f $CGIFILE ]; then rm $CGIFILE; fi
touch $CGIFILE
ex -s $CGIFILE <<EOF
a
#!/usr/bin/ksh
print "Content-type: text/html\n";

cat $HTMLFILE | sed 's:UNSTABLE:<font color="#FDD017">UNSTABLE<font
color="#ffffff">:g' | sed 's: STABLE:<font color="#00FF00"> STABLE<font
color="#ffffff">:g' | sed 's/qn:/<font color="#2B65EC">qn:<font
color="#ffffff">/g' | sed 's:UP:<font color="#00FF00">UP<font color="#ffffff">:g'
| sed 's:DOWN:<font color="#FF0000">DOWN<font color="#ffffff">:g'| sed
's:ONLINE:<font color="#00FF00">ONLINE<font color="#ffffff">:g' | sed
's:OFFLINE:<font color="#0000FF">OFFLINE<font color="#ffffff">:g' |sed '1,1d' >
/tmp/.aastat

cat << EOM
<HTML>
<META HTTP-EQUIV="REFRESH" CONTENT="5">
<HEAD><TITLE>HACMP Cluster Status - a Mancunian production </TITLE>
<BODY COLOR="white" LINK="red" VLINK="blue" BGCOLOR="white">
<div
style="position:fixed;width:700px;height:700px;top:0;bottom:0;left:0;right:0;margi
n:10px auto;padding:20px;background:black">
```

```
<PRE style="font-family:verdana,arial,sans-serif;font-size:16px;color:white">
Remote Custom Cluster Monitoring via SSH/SNMP
<HR SIZE=3>
EOM
cat /tmp/.aastat
.
wq
EOF
chmod 755 $CGIFILE

}

################################################################################
#
#  Name: print_address_info
#
#  Prints the address information for the node and network given in the
#  environment
#
################################################################################
print_address_info()
{
  [[ "$VERBOSE_LOGGING" = "high" ]] && set -x

  # Get key (IP addresses) from MIB
  addresses=$(echo "$ADDRESS_MIB_FUNC" | grep -w "$ADDRESS_IP.$node_id"| uniq |
sort | cut -f3 -d" ")

  # Get the active Node for each IP address
  for address in $addresses
  do
    address_net_id=$(echo "$ADDRESS_MIB_FUNC" | grep -w
"$ADDRESS_NET.$node_id.$address" | cut -f3 -d" ")

    if [[ "$address_net_id" = "$net_id" ]]
    then
       active_node=$(echo "$ADDRESS_MIB_FUNC" | grep -w
"$ADDRESS_ACTIVE_NODE.$node_id.$address" | cut -f3 -d" ")

        if [[ "$active_node" = $node_id ]]
        then
           address_label=$(echo "$ADDRESS_MIB_FUNC" | grep -w
"$ADDRESS_LABEL.$node_id.$address" | cut -f2 -d\")
           address_state=$(echo "$ADDRESS_MIB_FUNC" | grep -w
"$ADDRESS_STATE.$node_id.$address" | cut -f3 -d" ")
           printf "\t%-15s %-20s " $address $address_label

            case $address_state in
                2)
              printf "UP\n"
                  ;;
                4)
              printf "DOWN\n"
                  ;;
                *)
```

```
                printf "UNKNOWN\n"
                    ;;
             esac
         fi
     fi

  done
}

################################################################################
#
#  Name: print_rg_info
#
#  Prints the online RG status info.
#
################################################################################
print_rg_info()
{
i=1;
RGONSTAT=`echo "$CLUSTER_MIB" | grep -w "$node_name" |egrep -w
"(ONLINE|ERROR|ACQUIRING|RELEASING)" | while read A
do
        if [ i -eq 1 ];then printf "\n\tResource Group(s) active on
$node_name:\n"; fi
        echo "$A" | awk -F: '{printf "\t  %-15s %-10s %-10s\n", $1, $2, $9}'
        let i=i+1
        done`
#if [ $i -gt 1 ]; then printf "$RGONSTAT\n"; fi
echo $RGONSTAT  > /dev/null 2>&1
#echo $RGONSTAT | grep ONLINE > /dev/null 2>&1
#printf "$RGONSTAT\n"
if [ $? -eq 0 ]
then
        printf "$RGONSTAT\n"
fi
}

################################################################################
#
#  Name: print_network_info
#
#  Prints the network information for the node given in the environment
#
################################################################################
print_network_info()
{
  [[ "$VERBOSE_LOGGING" = "high" ]] && set -x

  # Get network IDs
   network_ids=$(echo "$NETWORK_MIB_FUNC" | grep -w "$NETWORK_ID.$node_id" | cut
-f3 -d" " | uniq | sort -n )

  # Get states for these networks on this node
  for net_id in $network_ids
  do
```

```
    printf "\n"
    network_name=$(echo "$NETWORK_MIB_FUNC" | grep -w
"$NETWORK_NAME.$node_id.$net_id" | cut -f2 -d\")
    network_attribute=$(echo "$NETWORK_MIB_FUNC" | grep -w
"$NETWORK_ATTRIBUTE.$node_id.$net_id" | cut -f3 -d" ")
    network_state=$(echo "$NETWORK_MIB_FUNC" | grep -w
"$NETWORK_STATE.$node_id.$net_id" | cut -f3 -d" ")
    formatted_network_name=$(echo "$network_name" | awk '{printf  "%-18s", $1}')

    printf "  Network : $formatted_network_name State: " "$formatted_network_name"
    case $network_state in
        2)
         printf "UP\n"
          ;;
        4)
         printf "DOWN\n"
          ;;
        32)
         printf "JOINING\n"
          ;;
        64)
         printf "LEAVING\n"
          ;;
        *)
         printf "N/A\n"
          ;;
    esac

    PRINT_IP_ADDRESS="true"

    # If serial type network, then don't attempt to print IP Address
    [[ $network_attribute -eq 4 ]] && PRINT_IP_ADDRESS="false"

    print_address_info

    #CAA SAN Comms
    # Note: Must be HA 7 and AIX 7.1 TL3 !!
    if [[ $CAA -eq 1 ]]; then
     caa_san_comms=`ssh $SSHparams $USER@$node_name /usr/lib/cluster/clras
sancomm_status | egrep -v '(--|UUID)' | awk -F'|' '{print $4}' | sed 's/ //g'`
     print "  CAA SAN Comms\t\tState: $caa_san_comms"
    fi

  done
}

###############################################################################
#
#  Name: print_node_info
#
#  Prints the node information for each node found in the MIB
#
###############################################################################
print_node_info()
{
```

```
  [[ "$VERBOSE_LOGGING" = "high" ]] && set -x

  NODE_ID_COUNTER=0

  while [[ $cluster_num_nodes -ne 0 ]]
  do
    # Get node information for each node
    node_id=$(echo "$NODE_MIB" | grep -w "$NODE_ID.$NODE_ID_COUNTER" | cut -f3 -d
" ")

    let NODE_ID_COUNTER=NODE_ID_COUNTER+1

    # Node ids may not be contiguous
    if [[ -z "$node_id" ]]
    then
       continue
    fi

    node_state=$(echo "$NODE_MIB" | grep -w "$NODE_STATE.$node_id" | cut -f3 -d"
")
    node_num_if=$(echo "$NODE_MIB" | grep -w "$NODE_NUM_IF.$node_id" | cut -f3 -d"
")
    node_name=$(echo "$NODE_MIB" | grep -w "$NODE_NAME.$node_id" | cut -f2 -d\")
    formatted_node_name=$(echo "$node_name" | awk '{printf  "%-15s", $1}')

    echo ""
    printf "Node : $formatted_node_name State: " "$formatted_node_name"
    if [ INTERNAL -eq 1 ]; then
       internal_state=`ssh $SSHparams $USER@$node_name lssrc -ls clstrmgrES
2>/dev/null |grep -i state |awk '{print $3}'`
       finternal_state=`echo "($internal_state)"`
    fi
    case $node_state in
        2)
         printf "UP $finternal_state\n"
          ;;
        4)
         printf "DOWN $finternal_state\n"
          ;;
        32)
         printf "JOINING $finternal_state\n"
          ;;
        64)
         printf "LEAVING $finternal_state\n"
          ;;
    esac

    NETWORK_MIB_FUNC=$(echo "$NETWORK_MIB" | grep -w
"$NETWORK_BRANCH\..\.$node_id")
    ADDRESS_MIB_FUNC=$(echo "$ADDRESS_MIB" | grep -w
"$ADDRESS_BRANCH\..\.$node_id")

    if [ $NETWORK = "TRUE" ]; then
     print_network_info
```

```
    fi
    print_rg_info

    let cluster_num_nodes=cluster_num_nodes-1

  done

}

###############################################################################
#
#  Name: print_cluster_info
#
#  Prints the cluster information for the cluster found in the MIB of which
#  this node is a member.
#
###############################################################################
print_cluster_info ()
{
  HANODE=$1

  cluster_name=$(echo "$CLUSTER_MIB" | grep -w "$CLUSTER_NAME\.0" | cut -f2 -d\")

  cluster_state=$(echo "$CLUSTER_MIB" | grep -w "$CLUSTER_STATE\.0" | cut -f3 -d"
")
  cluster_substate=$(echo "$CLUSTER_MIB" | grep -w "$CLUSTER_SUBSTATE\.0" | cut
-f3 -d" ")

  case $cluster_state in
       2)
         cs="UP"
         ;;
       4)
         cs="DOWN"
         ;;
  esac

  case $cluster_substate in
       4)
         css="DOWN"
         ;;
       8)
         css="UNKNOWN"
         ;;
       16)
         css="UNSTABLE"
         ;;
       2 | 32)
         css="STABLE"
         ;;
       64)
         css="ERROR"
         ;;
       128)
         css="RECONFIG"
```

```
        ;;
  esac

echo "\\033[H\\033[2J\n\t\tStatus for $cluster_name on $(date +%d" "%b" "%y" "%T)"
echo "\t\t\t  Cluster is ($cs & $css)    qn: $HANODE\n"

cluster_num_nodes=$(echo "$CLUSTER_MIB" | grep -w "$CLUSTER_NUM_NODES\.0" | cut
-f3 -d" ")

print_node_info
echo "\n"

}

################################################################################
# Main
################################################################################

# sort the flags

trap "rm $LOGFILE $HTMLFILE; exit 0" 1 2 12 9 15
NETWORK="TRUE"; STOP=0
while getopts :n1ic ARGs
do
 case $ARGs in
       n) NETWORK="FALSE" ;;
       1) STOP=1 ;;
       i) INTERNAL=1 ;;
       c) CAA=1 ;;
       \?) printf "\nNot a valid option\n\n" ; usage ; exit ;;
esac
done

################ get the nodes and start

format_cgi
while true
do
 for NODE in `cat $CLHOSTS |grep -v "^#"`
 do
   SUCCESS=1
   while [ SUCCESS -eq 1 ]
   do
        #ping -w 1 -c1 $NODE > /dev/null 2>&1
       ssh $SSHparams ${USER}@${NODE} date > /dev/null 2>&1
        if [ $? -eq 0 ]; then
        # get the snmp info
         CLUSTER_MIB=`ssh $SSHparams $USER@$NODE "snmpinfo -c $SNMPCOMM -m dump -o
/usr/es/sbin/cluster/hacmp.defs cluster
         snmpinfo -c $SNMPCOMM -m dump -o /usr/es/sbin/cluster/hacmp.defs network
         snmpinfo -c $SNMPCOMM -m dump -o /usr/es/sbin/cluster/hacmp.defs node
         snmpinfo -c $SNMPCOMM -m dump -o /usr/es/sbin/cluster/hacmp.defs address
         /usr/es/sbin/cluster/utilities/clfindres -s 2> /dev/null"`
         # is there any snmp info?
         snmpinfocheck=`echo $CLUSTER_MIB |grep $CLUSTER_BRANCH`
```

```
        if [[ $RC -eq 0 && $snmpinfocheck != "" ]]; then
            NODE_MIB=$CLUSTER_MIB
            NETWORK_MIB=$CLUSTER_MIB
            ADDRESS_MIB=$CLUSTER_MIB
            # Print Topology Information
                SUCCESS=1 && print_cluster_info $NODE > $LOGFILE
                cat $LOGFILE
            cp $LOGFILE $HTMLFILE
            if [ $STOP -eq 1 ]; then exit; fi
          else
                SUCCESS=0 && echo "\n Data unavailable on NODE: $NODE \n
            Check clhost file and/or local cluster node state"
          fi
        else
                SUCCESS=0 && echo "\n NODE: $NODE not responding"
        fi
   done
 done
done

exit 0
```

PowerHA MIB file

You can use the MIB file shown in Example B-5 for the integration of PowerHA V5/V6/V7 with external tools as required.

Example B-5 PowerHA MIB (hacmp.my)

```
-- @(#)51       1.31  src/43haes/usr/sbin/cluster/clsmuxpd/hacmp.my,
hacmp.clsmuxpd, 61haes_r712 5/7/08 06:11:44
-- IBM_PROLOG_BEGIN_TAG
-- This is an automatically generated prolog.
--
-- 61haes_r712 src/43haes/usr/sbin/cluster/clsmuxpd/hacmp.my 1.31
--
-- Licensed Materials - Property of IBM
--
-- COPYRIGHT International Business Machines Corp. 1990,2008
-- 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
--
--   COMPONENT_NAME: CLSMUXPD
--
--   FUNCTIONS: none
--

---------------------------------------------------------------------------
--
```

```
-- I. HEADER
--
-- A) High Availability Cluster Multi-Processing for AIX Cluster
--        SNMP Peer MIB Definition.
--
   RISC6000CLSMUXPD-MIB
--
   DEFINITIONS ::= BEGIN
--
-- B) Imported syntaxes.
--
   IMPORTS
      enterprises, IpAddress, Counter
         FROM RFC1065-SMI
      DisplayString
         FROM RFC1213-MIB
                OBJECT-TYPE
                          FROM RFC-1212;

--
-- C) The root of the RISC6000CLSMUXPD-MIB is as follows:
--
        ibm             OBJECT IDENTIFIER    ::= { enterprises 2 }
        ibmAgents       OBJECT IDENTIFIER    ::= { ibm 3 }
        aix             OBJECT IDENTIFIER    ::= { ibmAgents 1  }
        aixRISC6000     OBJECT IDENTIFIER    ::= { aix 2 }
        risc6000agents  OBJECT IDENTIFIER    ::= { aixRISC6000 1 }
        risc6000clsmuxpdOBJECT IDENTIFIER    ::= { risc6000agents 5 }
--
        clusterOBJECT IDENTIFIER    ::= { risc6000clsmuxpd 1 }
        nodeOBJECT IDENTIFIER    ::= { risc6000clsmuxpd 2 }
        addressOBJECT IDENTIFIER    ::= { risc6000clsmuxpd 3 }
        networkOBJECT IDENTIFIER    ::= { risc6000clsmuxpd 4 }
--
        clstrmgrOBJECT IDENTIFIER    ::= { risc6000clsmuxpd 5 }
        cllockdOBJECT IDENTIFIER    ::= { risc6000clsmuxpd 6 }
        clinfoOBJECT IDENTIFIER    ::= { risc6000clsmuxpd 7 }
--
        applicationOBJECT IDENTIFIER    ::= { risc6000clsmuxpd 8 }
--
        clsmuxpdOBJECT IDENTIFIER    ::= { risc6000clsmuxpd 9 }
        eventOBJECT IDENTIFIER    ::= { risc6000clsmuxpd 10 }
--
        resmanager      OBJECT IDENTIFIER    ::= { risc6000clsmuxpd 11 }
        site            OBJECT IDENTIFIER    ::= { risc6000clsmuxpd 12 }
--
        address6        OBJECT IDENTIFIER    ::= { risc6000clsmuxpd 13 }
--
-- II. The Cluster Group
--
-- A) clusterId
--    This field is read from the HACMP for AIX object repository.
--
   clusterIdOBJECT-TYPE
      SYNTAXINTEGER
```

```
        ACCESSread-only
        STATUSmandatory
        DESCRIPTION
           "The ID of the cluster"
        ::= { cluster 1 }
--
-- B) clusterName
--      This field is read from the HACMP for AIX object repository.
--
    clusterNameOBJECT-TYPE
        SYNTAXDisplayString
        ACCESSread-only
        STATUSmandatory
        DESCRIPTION
           "User configurable cluster Name"
        ::= { cluster 2 }
--
-- C) clusterConfiguration
--      This field is read from the HACMP for AIX object repository.
--
    clusterConfigurationOBJECT-TYPE
        SYNTAXDisplayString
        ACCESSread-only
        STATUSdeprecated
        DESCRIPTION
           "The cluster configuration"
        ::= { cluster 3 }
--
-- D) clusterState
--      This field is returned by the clstrmgr.
--
    clusterStateOBJECT-TYPE
        SYNTAXINTEGER { up(2), down(4),
                unknown(8), notconfigured(256) }
        ACCESSread-only
        STATUSmandatory
        DESCRIPTION
           "The cluster status"
        ::= { cluster 4 }

    trapClusterStateTRAP-TYPE
        ENTERPRISErisc6000clsmuxpd
        VARIABLES{ clusterState, clusterId, clusterNodeId }
        DESCRIPTION
           "Fires whenever the cluster changes state."
        ::= 10

--
-- E) clusterPrimary
--      This field is returned by the clstrmgr.
--      Status is deprecated as lock manager is no longer supported.
--
--
    clusterPrimaryOBJECT-TYPE
        SYNTAXINTEGER
```

```
    ACCESSread-only
    STATUSdeprecated
    DESCRIPTION
       "The Node ID of the Primary Lock Manager"
    ::= { cluster 5 }

--
-- F) clusterLastChange
--    This field is a integer string returned by the gettimeofday()
--    library call and is updated if any cluster, node,
--    or address information changes.
--
   clusterLastChangeOBJECT-TYPE
    SYNTAXINTEGER
    ACCESSread-only
    STATUSmandatory
    DESCRIPTION
       "Time in seconds of last change in this cluster."
    ::= { cluster 6 }
--
-- G) clusterGmtOffset
--    This field is a integer string returned by the gettimeofday()
--    library call and is updated if any cluster, node,
--    or address information changes.
--
   clusterGmtOffsetOBJECT-TYPE
    SYNTAXINTEGER
    ACCESSread-only
    STATUSmandatory
    DESCRIPTION
       "Seconds west of GMT for the time of last change in this cluster."
    ::= { cluster 7 }
--
-- H) clusterSubState
--    This field is returned by the clstrmgr.
--
--
   clusterSubStateOBJECT-TYPE
    SYNTAXINTEGER { unstable(16), error(64),
            stable(32), unknown(8), reconfig(128),
            notconfigured(256), notsynced(512) }
    ACCESSread-only
    STATUSmandatory
    DESCRIPTION
       "The cluster substate"
    ::= { cluster 8 }

   trapClusterSubStateTRAP-TYPE
    ENTERPRISErisc6000clsmuxpd
    VARIABLES{ clusterSubState, clusterId, clusterNodeId }
    DESCRIPTION
       "Fires whenever the cluster changes substate."
    ::= 11

--
```

```
-- I) clusterNodeName
--    This field is read from the HACMP for AIX object repository.
--
   clusterNodeNameOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "User configurable cluster local node name"
      ::= { cluster 9 }
--
--      J) clusterPrimaryNodeName
--              This field is returned by the clstrmgr.
--
--
   clusterPrimaryNodeName  OBJECT-TYPE
      SYNTAX  DisplayString
      ACCESS  read-only
      STATUS  mandatory
      DESCRIPTION
         "The Node Name of the primary cluster node"
      ::= { cluster 10 }

   trapNewPrimaryTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ clusterPrimary, clusterId, clusterNodeId }
      DESCRIPTION
         "Fires whenever the primary node changes."
      ::= 15
--
--      K) clusterNumNodes
--              This field is returned by the clstrmgr.
--
--
   clusterNumNodes  OBJECT-TYPE
      SYNTAX  INTEGER
      ACCESS  read-only
      STATUS  mandatory
      DESCRIPTION
         "The number of nodes in the cluster"
      ::= { cluster 11 }
--
-- L) clusterNodeId
--    This field is read from the HACMP for AIX object repository.
--
   clusterNodeIdOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The ID of the local node"
      ::= { cluster 12 }
--
--      M) clusterNumSites
--              This field is returned by the clstrmgr.
```

```
--
--
   clusterNumSites  OBJECT-TYPE
      SYNTAX  INTEGER
      ACCESS  read-only
      STATUS  mandatory
      DESCRIPTION
         "The number of sites in the cluster"
      ::= { cluster 13 }

--
--
-- III. The node group
--
-- A) The node table
--    This is a variable length table which is indexed by
--    the node Id.
--
   nodeTableOBJECT-TYPE
      SYNTAXSEQUENCE OF NodeEntry
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "A series of Node descriptions"
      ::= { node 1 }
--
   nodeEntryOBJECT-TYPE
      SYNTAXNodeEntry
      ACCESSnot-accessible
      STATUSmandatory
      INDEX{ nodeId }
      ::= { nodeTable 1 }
--
   NodeEntry::= SEQUENCE {
      nodeIdINTEGER,
      nodeStateINTEGER,
      nodeNumIfINTEGER,
      nodeNameDisplayString,
      nodeSiteDisplayString
   }
--
-- B) nodeId
--    This field is read from the HACMP for AIX object repository.
--
   nodeIdOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The ID of the Node"
      ::= { nodeEntry 1 }
--
-- C) nodeState
--    This row is returned by the clstrmgr.
--
```

```
--
   nodeStateOBJECT-TYPE
      SYNTAXINTEGER { up(2), down(4),
              joining(32), leaving(64) }
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The State of the Node"
      ::= { nodeEntry 2 }

   trapNodeStateTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ nodeState, clusterId, clusterNodeId }
      DESCRIPTION
         "Fires whenever a node changes state."
      ::= 12

--
-- D) nodeNumIf
--    This row is returned by the clstrmgr.
--
--
   nodeNumIfOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The number of network interfaces in this node"
      ::= { nodeEntry 3 }
--
-- E) nodeName
--    This row is returned by the clstrmgr.
--
--
   nodeNameOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The name of this node"
      ::= { nodeEntry 4 }

   nodeSiteOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The site associated with this node"
      ::= { nodeEntry 5 }

--
--
--  The site group
--
-- A) The site table
```

```
--    This is a variable length table which is indexed by
--    the site Id.
--
   siteTableOBJECT-TYPE
      SYNTAXSEQUENCE OF SiteEntry
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "A series of Site descriptions"
      ::= { site 1 }
--
   siteEntryOBJECT-TYPE
      SYNTAXSiteEntry
      ACCESSnot-accessible
      STATUSmandatory
      INDEX{ siteId }
      ::= { siteTable 1 }
--
   SiteEntry::= SEQUENCE {
      siteIdINTEGER,
      siteNameDisplayString,
      sitePriorityINTEGER,
      siteBackupINTEGER,
      siteNumNodesINTEGER,
      siteStateINTEGER
   }
--
-- B) siteId
--    This field is read from the HACMP for AIX object repository.
--
   siteIdOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The ID of the site"
      ::= { siteEntry 1 }
--
-- C) siteName
--    This row is returned by the clstrmgr.
--
--
   siteNameOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The name of this site"
      ::= { siteEntry 2 }
--
-- E) sitePriority
--    Priority or dominance of the site
--
--
   sitePriorityOBJECT-TYPE
```

```
      SYNTAXINTEGER { primary(1), secondary(2),
              tertiary(4) }
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The Priority of the Site"
      ::= { siteEntry 3 }
--
-- F) siteBackup
--    Backup communications method for the site
--
--
   siteBackupOBJECT-TYPE
      SYNTAXINTEGER { none(1), dbfs(2),
              sgn(4) }
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "Backup communications method for the site"
      ::= { siteEntry 4 }
--
-- G) siteNumNodes
--    Number of nodes in this site
--
--
   siteNumNodesOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The number of nodes in this site"
      ::= { siteEntry 5 }
--
-- D) siteState
--    This row is returned by the clstrmgr.
--
--
   siteStateOBJECT-TYPE
      SYNTAXINTEGER { up(2), down(4),
              joining(16), leaving(32),
              isolated(257)
           }
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The State of the site"
      ::= { siteEntry 6 }

   trapSiteStateTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ siteState, clusterId, siteId }
      DESCRIPTION
         "Fires whenever a site changes state."
      ::= 18
```

```
--
--
-- Site node table
--
-- A) The site to node mapping Table
--
--
   siteNodeTableOBJECT-TYPE
      SYNTAXSEQUENCE OF SiteNodeEntry
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "A series of site node descriptions"
      ::= { site 2 }
--
   siteNodeEntryOBJECT-TYPE
      SYNTAXSiteNodeEntry
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "Node ids for all nodes in this site"
      INDEX { siteNodeSiteId, siteNodeNodeId }
      ::= { siteNodeTable 1 }
--
   SiteNodeEntry::= SEQUENCE {
      siteNodeSiteIdINTEGER,
      siteNodeNodeIdINTEGER
   }
--
-- B) The site Id
--    HACMP defined site id.
--
        siteNodeSiteId     OBJECT-TYPE
                SYNTAXINTEGER
                ACCESSread-only
                STATUSmandatory
                DESCRIPTION
                        "The ID of the cluster site"
      ::= { siteNodeEntry 1 }
--
-- C) The site node Id
--    The node id for each node in this site
--
        siteNodeNodeId     OBJECT-TYPE
                SYNTAXINTEGER
                ACCESSread-only
                STATUSmandatory
                DESCRIPTION
                        "The node ID of the node in this site"
      ::= { siteNodeEntry 2 }

--
--
--
```

```
-- IV. The address group
--
--
-- IV. The address group
--
-- A) The address table
--    This is a variable length table which is indexed by
--    the node Id and the dotted decimal IP address.
--
   addrTableOBJECT-TYPE
      SYNTAXSEQUENCE OF AddrEntry
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "A series of IP address descriptions"
      ::= { address 1 }
--
   addrEntryOBJECT-TYPE
      SYNTAXAddrEntry
      ACCESSnot-accessible
      STATUSmandatory
      INDEX{ addrNodeId, addrAddress }
      ::= { addrTable 1 }
--
   AddrEntry::= SEQUENCE {
      addrNodeId  INTEGER,
      addrAddress  IpAddress,
      addrLabel  DisplayString,
      addrRole  INTEGER,
      addrNetId  INTEGER,
      addrState  INTEGER,
      addrActiveNode  INTEGER,
      oldAddrActiveNode INTEGER
   }
--
-- B) addrNodeId
--    This field is read from the HACMP for AIX object repository.
--
   addrNodeIdOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The ID of the Node this IP address is configured"
      ::= { addrEntry 1 }
--
-- C) addrAddress
--    This field is read from the HACMP for AIX object repository.
--
   addrAddressOBJECT-TYPE
      SYNTAXIpAddress
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The IP address"
```

```
      ::= { addrEntry 2 }
--
-- D) addrLabel
--    This field is read from the HACMP for AIX object repository.
--
   addrLabelOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The IP label associated with the IP address"
      ::= { addrEntry 3 }
--
-- E) addrRole
--    This field is read from the HACMP for AIX object repository.
--    Note that use of sharedService and standby is deprecated.
--
   addrRoleOBJECT-TYPE
      SYNTAXINTEGER { boot(64), service(16),
              persistent(8),
              sharedService(128), standby(32) }
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The role of the IP address"
      ::= { addrEntry 4 }
--
-- F) addrNetId
--    This field is read from the HACMP for AIX object repository.
--    It is provide so that clients can determine the corresponding
--    index into the network table.
--
   addrNetIdOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The network ID of the IP address"
      ::= { addrEntry 5 }
--
-- G) addrState
--    This field is returned from the Cluster Manager.
--
   addrStateOBJECT-TYPE
      SYNTAXINTEGER { up(2), down(4), unknown(8) }
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The state of the IP address"
      ::= { addrEntry 6 }

--
-- H) addrActiveNode
--    This field is returned from the Cluster Manager.
--
```

```
   addrActiveNodeOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The ID of the Node on which this IP address is active"
      ::= { addrEntry 7 }
--
--  I) oldAddrActiveNode
--      This field is returned from the Cluster Manager.
--
    oldAddrActiveNode OBJECT-TYPE
            SYNTAX  INTEGER
            ACCESS  not-accessible
            STATUS  mandatory
            DESCRIPTION
                    "The ID of the Node on which this IP address was previously
active"
            ::= { addrEntry 8 }



--
--
-- V. The network group
--
-- A) The network table
--    This is a variable length table index by node Id
--    and network Id.
--
   netTableOBJECT-TYPE
      SYNTAXSEQUENCE OF NetEntry
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "A series of Network descriptions"
      ::= { network 1 }
--
   netEntryOBJECT-TYPE
      SYNTAXNetEntry
      ACCESSnot-accessible
      STATUSmandatory
      INDEX{ netNodeId, netId }
      ::= { netTable 1 }
--
   NetEntry::= SEQUENCE {
      netNodeIdINTEGER,
      netId INTEGER,
      netNameDisplayString,
      netAttributeINTEGER,
      netStateINTEGER,
      netODMidINTEGER,
      netTypeDisplayString,
      netFamily   INTEGER
   }
--
```

```
-- B) netNodeId
--    This field is read from the HACMP for AIX object repository.
--
   netNodeIdOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The ID of the Node this network is configured"
      ::= { netEntry 1 }
--
-- C) netId
--    This field is read from the HACMP for AIX object repository.
--
   netIdOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The ID of the network"
      ::= { netEntry 2 }
--
-- D) netName
--    This field is read from the HACMP for AIX object repository.
--
   netNameOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The name of network"
      ::= { netEntry 3 }
--
-- E) netAttribute
--    This field is read from the HACMP for AIX object repository.
--    If the attribute is public or private, it is an IP based
--    network, otherwise it is non-IP or serial. Note that the
--    public / private setting is only used by Oracle for selecting
--    a network for intra-node communications - it has no effect
--    on HACMP's handling of the network.
--
   netAttributeOBJECT-TYPE
      SYNTAXINTEGER { public(2), private(1), serial(4) }
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The attribute of the network."
      ::= { netEntry 4 }
--
-- F) netState
--    This row is returned by the clstrmgr.
--
--
   netStateOBJECT-TYPE
      SYNTAXINTEGER { up(2), down(4), joining(32), leaving(64) }
```

```
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The state of the network"
      ::= { netEntry 5 }

   trapNetworkStateTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ netState, clusterId, clusterNodeId }
      DESCRIPTION
         "Fires whenever a network changes state."
      ::= 13
--
-- G) netODMid
--    This field is read from the HACMP for AIX object repository.
--
   netODMidOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The ODM id of the network"
      ::= { netEntry 6 }
--
-- H) netType
--    This field is read from the HACMP for AIX object repository.
--    It indicates the physical type of the network: ethernet, token
--    ring, ATM, etc.
--
   netTypeOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The physical network type, e.g. ethernet"
      ::= { netEntry 7 }
--
--       I) netFamily
--               This field is read from the HACMP for AIX object repository.
--               It indicates if the HACMP-network is a INET/INET6/Hybrid network.
--
         netFamily OBJECT-TYPE
               SYNTAX  INTEGER { unknown(0), clinet(1), clinet6(2), clhybrid(3) }
               ACCESS  read-only
               STATUS  mandatory
               DESCRIPTION
                       "Family of the network."
               ::= { netEntry 8 }

--
--
--
--
-- VI. The Cluster Manager (clstrmgr) group
--
```

```
-- A) The clstrmgr table
--    This is a variable length table which is indexed by
--    the node Id.
--
   clstrmgrTableOBJECT-TYPE
      SYNTAXSEQUENCE OF ClstrmgrEntry
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "A series of clstrmgr entries"
      ::= { clstrmgr 1 }
--
   clstrmgrEntryOBJECT-TYPE
      SYNTAXClstrmgrEntry
      ACCESSnot-accessible
      STATUSmandatory
      INDEX{ clstrmgrNodeId }
      ::= { clstrmgrTable 1 }
--
   ClstrmgrEntry::= SEQUENCE {
      clstrmgrNodeIdINTEGER,
      clstrmgrVersionDisplayString,
      clstrmgrStatusINTEGER
   }
--
-- B) clstrmgrNodeId
--    This field is read from the cluster configuration.
--
   clstrmgrNodeIdOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The node ID of the Cluster Manager"
      ::= { clstrmgrEntry 1 }
--
-- C) clstrmgrVersion
--    This field is hard coded into the daemon.
--
   clstrmgrVersionOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The version of the Cluster Manager"
      ::= { clstrmgrEntry 2 }
--
-- D) clstrmgrStatus
--    This field indicates the state of the cluster manager on the
--    node.
--    Note that "suspended" and "unknown" are no longer used.
--    graceful, forced and takeover reflect the mode
--    which was used to stop cluster services.
--
   clstrmgrStatusOBJECT-TYPE
```

```
    SYNTAXINTEGER{ up(2), down(4), suspended(16), unknown(8),
            graceful(32), forced(64), takeover(128) }
    ACCESSread-only
    STATUSmandatory
    DESCRIPTION
       "The state of the Cluster Manager"
    ::= { clstrmgrEntry 3 }
--
--
-- VII. The Cluster Lock Daemon (cllockd) group
--
-- The cluster lock daemon is no longer supported, the information
-- in this section is for reference only.
--
-- A) The cllockd table
--    This is a variable length table which is indexed by
--    the node Id.
--
      cllockdTable    OBJECT-TYPE
              SYNTAX  SEQUENCE OF CllockdEntry
              ACCESS  not-accessible
              STATUS  mandatory
              DESCRIPTION
                      "A series of cllockd process entries"
              ::= { cllockd 1 }
--
      cllockdEntry    OBJECT-TYPE
              SYNTAX  CllockdEntry
              ACCESS  not-accessible
              STATUS  mandatory
              INDEX   { cllockdNodeId }
              ::= { cllockdTable 1 }
--
      CllockdEntry    ::= SEQUENCE {
              cllockdNodeId   INTEGER,
              cllockdVersion  DisplayString,
              cllockdStatus   INTEGER
      }
--
--    B) cllockdNodeId
--            This field is determined by the IP address used to connect
--            to the cllockd.
--
      cllockdNodeId   OBJECT-TYPE
              SYNTAX  INTEGER
              ACCESS  read-only
              STATUS  mandatory
              DESCRIPTION
                      "The node ID of the Lock Manager"
              ::= { cllockdEntry 1 }
--
--    C) cllockdVersion
--            This field is returned by the srcstat() library call.
--
      cllockdVersion  OBJECT-TYPE
```

```
                SYNTAX  DisplayString
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "The version of the Lock Manager"
                ::= { cllockdEntry 2 }
--
--      D) cllockdStatus
--              This field is is always 4 (down) - cllockd is no longer
--              supported
--
        cllockdStatus   OBJECT-TYPE
                SYNTAX  INTEGER { up(2), down(4), unknown(8),
                                        suspended(16), stalled(256) }
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "The state of the Lock Manager"
                ::= { cllockdEntry 3 }

--
--
-- VIII. The Client Information Daemon (clinfo) group
--
-- A) The clinfo table
--    This is a variable length table which is indexed by
--    the node Id.
--
   clinfoTableOBJECT-TYPE
      SYNTAXSEQUENCE OF ClinfoEntry
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "A series of clinfo process entries"
      ::= { clinfo 1 }
--
   clinfoEntryOBJECT-TYPE
      SYNTAXClinfoEntry
      ACCESSnot-accessible
      STATUSmandatory
      INDEX{ clinfoNodeId }
      ::= { clinfoTable 1 }
--
   ClinfoEntry::= SEQUENCE {
      clinfoNodeIdINTEGER,
      clinfoVersionDisplayString,
      clinfoStatusINTEGER
   }
--
-- B) clinfoNodeId
--    This field is the cluster node id running the clinfo daemon.
--
   clinfoNodeIdOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
```

```
      STATUSmandatory
      DESCRIPTION
         "The node ID running the Client Information Daemon"
      ::= { clinfoEntry 1 }
--
-- C) clinfoVersion
--    This is the hard coded version of the clinfo daemon.
--
   clinfoVersionOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The version of the Client Information Daemon"
      ::= { clinfoEntry 2 }
--
-- D) clinfoStatus
--    This status of the daemon on the node.
--    Note that "suspended" state is no longer supported.
--
   clinfoStatusOBJECT-TYPE
      SYNTAXINTEGER { up(2), down(4), unknown(8), suspended(16) }
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The state of the Client Information Daemon"
      ::= { clinfoEntry 3 }
--
--
-- IX. The Application Group
--
-- A) The application table
--    This is a variable length table which is indexed by
--    the node Id followed by the application process Id.
--
--    There is an api which allows
--    applications to register with the HACMP for AIX-SMUX peer.
--
--    See cl_registerwithclsmuxpd() routine in the
--    Programming Client Applications Guide (SC23-4865)
--
   appTableOBJECT-TYPE
      SYNTAXSEQUENCE OF AppEntry
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "A series of application entries"
      ::= { application 1 }
--
   appEntryOBJECT-TYPE
      SYNTAXAppEntry
      ACCESSnot-accessible
      STATUSmandatory
      INDEX{ appNodeId, appPid }
      ::= { appTable 1 }
```

```
--
   AppEntry::= SEQUENCE {
      appNodeIdINTEGER,
      appPidINTEGER,
      appNameDisplayString,
      appVersionDisplayString,
      appDescrDisplayString
   }
--
-- B) appNodeId
--    This is the (cluster) node id where the client has
--    registered the routine (clsmuxpd provides this field).
--
   appNodeIdOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The node ID of the application"
      ::= { appEntry 1 }
--
-- C) appPid
--    This is the process id where the client has
--    registered the routine (clsmuxpd provides this field).
--
   appPidOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The process ID of the application"
      ::= { appEntry 2 }
--
-- D) appName
--    This field is passed to the cl_registerwithclsmuxpd() routine.
--
   appNameOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The name of the application"
      ::= { appEntry 3 }

--
-- E) appVersion
--    This field is passed to the cl_registerwithclsmuxpd() routine.
--
   appVersionOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The version of the application"
      ::= { appEntry 4 }
```

```
--
-- F) appDescr
--    This field is passed to the cl_registerwithclsmuxpd() routine.
--
   appDescrOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The description of the application"
      ::= { appEntry 5 }

--
-- trapAppState
--    This fires whenever the state of the application changes.
--    Note that this is based on application's socket connection
--    with the clsmuxpd daemon: when the socket is active, the
--    application is considered up, otherwise its down.
--
   trapAppStateTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ appName, clusterId, clusterNodeId }
      DESCRIPTION
         "Fires whenever an application is added or deleted."
      ::= 16

--
--
-- X. The Resource Group
-- Contains information about cluster resources and resource groups.
--
-- A) The Resource Group Table
--
   resGroupTableOBJECT-TYPE
      SYNTAXSEQUENCE OF ResGroupEntry
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "A series of Resource Group descriptions"
      ::= { resmanager 1 }
--
   resGroupEntryOBJECT-TYPE
      SYNTAXResGroupEntry
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "Individual Resource Group description"
      INDEX { resGroupId }
      ::= { resGroupTable 1 }
--
   ResGroupEntry::= SEQUENCE {
      resGroupIdINTEGER,
      resGroupNameDisplayString,
      resGroupPolicyINTEGER,
      resGroupUserPolicyNameDisplayString,
```

```
        resGroupNumResourcesINTEGER,
        resGroupNumNodesINTEGER
    }

--
-- B) Resource Group Id
--
--
    resGroupIdOBJECT-TYPE
        SYNTAXINTEGER
        ACCESSread-only
        STATUSmandatory
        DESCRIPTION
            "The ID of the Resource Group"
        ::= { resGroupEntry 1 }

    trapRGAddTRAP-TYPE
        ENTERPRISErisc6000clsmuxpd
        VARIABLES{ resGroupId }
        DESCRIPTION
            "Fires whenever a resource group is added."
        ::= 20

    trapRGDelTRAP-TYPE
        ENTERPRISErisc6000clsmuxpd
        VARIABLES{ resGroupId }
        DESCRIPTION
            "Fires whenever a resource group is deleted."
        ::= 21

--
-- C) Resource Group Name
--
--
    resGroupNameOBJECT-TYPE
        SYNTAXDisplayString
        ACCESSread-only
        STATUSmandatory
        DESCRIPTION
            "The name of the Resource Group"
        ::= { resGroupEntry 2 }

--
-- D) Resource Group Policy
--
--
    resGroupPolicyOBJECT-TYPE
        SYNTAXINTEGER {
                    cascading(1),
                    rotating(2),
                    concurrent(3),
                    userdefined(4),
                    custom(5)
                }
        ACCESSread-only
```

```
      STATUSmandatory
      DESCRIPTION
         "The State of the Resource Group"
      ::= { resGroupEntry 3 }

--
-- E) Resource Group User-Defined Policy Name
--
--
   resGroupUserPolicyNameOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The name of the user-defined policy"
      ::= { resGroupEntry 4 }

--
-- F) Number of Resources in a Resource Group
--
--
   resGroupNumResourcesOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The number of resources defined in the group"
      ::= { resGroupEntry 5 }

--
-- G) Number of Participating Nodes in a Resource Group
--
--
   resGroupNumNodesOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The number of participating nodes in the group"
      ::= { resGroupEntry 6 }

   trapRGChangeTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ resGroupId, resGroupPolicy,
              resGroupNumResources, resGroupNumNodes }
      DESCRIPTION
         "Fires whenever the policy, number of nodes,
         or the number of resources of a resource
         group is changed."
      ::= 22

--
-- H) Resource Group's Startup Policy
--
--
```

```
   resGroupStartupPolicyOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The Resource Group's Startup Policy:
            This can have the following values:
            Online On HomeNode Only - 1
            Online On First Available Node - 2
            Online Using Distribution Policy - 3
            Online On All Available Nodes - 4"
      ::= { resGroupEntry 7 }

--
-- I) Resource Group's Fallover Policy
--
--
   resGroupFalloverPolicyOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The Resource Group's Fallover Policy
            This can have the following values:
            Fallover To Next Priority Node On the List - 5
            Fallover Using DNP - 6
            Bring Offline - 7"
      ::= { resGroupEntry 8 }

--
-- J) Resource Group's Fallback Policy
--
--
   resGroupFallbackPolicyOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The Resource Group's Fallback Policy
            Fallback to Higher Priority Node in the List - 8
            Never Fallback - 9"
      ::= { resGroupEntry 9 }


--
--
-- XI. The Resources
--
-- A) The Resource Table
--
--
   resTableOBJECT-TYPE
      SYNTAXSEQUENCE OF ResEntry
      ACCESSnot-accessible
      STATUSmandatory
```

```
      DESCRIPTION
         "A series of Resource descriptions"
      ::= { resmanager 2 }
--
   resEntryOBJECT-TYPE
      SYNTAXResEntry
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "Individual Resource descriptions"
      INDEX { resGroupId, resourceId }
      ::= { resTable 1 }
--
   ResEntry::= SEQUENCE {
      resourceGroupIdINTEGER,
      resourceIdINTEGER,
      resourceNameDisplayString,
      resourceTypeINTEGER
   }
--
-- B) The Resource Group Id
--    HACMP defined group id.
--
        resourceGroupId     OBJECT-TYPE
                SYNTAXINTEGER
                ACCESSread-only
                STATUSmandatory
                DESCRIPTION
                        "The ID of the resource group"
                ::= { resEntry 1 }

--
-- C) Resource Id
--    This is stored in the hacmp configuration.
--
   resourceIdOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The ID of the Resource"
      ::= { resEntry 2 }

--
-- D) Resource Name
--    User supplied name, e.g. "Ora_vg1" or "app_serv1"
--
   resourceNameOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The name of this resource"
      ::= { resEntry 3 }
```

```
--
-- E) Resource Type
--    What kind of resource is it.
--
--
    resourceType        OBJECT-TYPE
        SYNTAX      INTEGER {
                    serviceLabel(1000), iPLabel(1000),
                    htyServiceLabel(1001),
                    fileSystem(1002),
                    volumeGroup(1003),
                    disk(1004), rawDiskPVID(1004),
                    aixConnectionServices(1005),
                    application(1006),
                    concurrentVolumeGroup(1007),
                    haCommunicationLinks(1008),
                    haFastConnectServices(1009)
                }

      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The Type of the Resource"
      ::= { resEntry 4 }

--
-- XII. The Resource Group Node State
--
-- A) The Resource Group Node State Table
--             The participating nodes and the current location of a given
resource
--             group are determined and maintained via this table and indexed by
--             resource group ID and node ID.
--
--
   resGroupNodeTableOBJECT-TYPE
      SYNTAXSEQUENCE OF ResGroupNodeEntry
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "A series of resource group and associated node state descriptions"
      ::= { resmanager 3 }
--
   resGroupNodeEntryOBJECT-TYPE
      SYNTAXResGroupNodeEntry
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "Individual resource group/node state descriptions"
      INDEX { resGroupNodeGroupId, resGroupNodeId }
      ::= { resGroupNodeTable 1 }
--
   ResGroupNodeEntry::= SEQUENCE {
      resGroupNodeGroupIdINTEGER,
      resGroupNodeIdINTEGER,
```

```
     resGroupNodeStateINTEGER
   }

--
-- B) The Resource Group Id
--    Cluster wide unique id assigned by hacmp.
--
--
       resGroupNodeGroupId    OBJECT-TYPE
               SYNTAXINTEGER
               ACCESSread-only
               STATUSmandatory
               DESCRIPTION
                       "The ID of the resource group"
               ::= { resGroupNodeEntry 1 }

--
-- C) The Participating Node Id
--    Node id of each node in the group.
--
--
       resGroupNodeIdOBJECT-TYPE
               SYNTAXINTEGER
               ACCESSread-only
               STATUSmandatory
               DESCRIPTION
                       "Node ID of node located within resource group"
               ::= { resGroupNodeEntry 2 }


--
-- D) The Resource Group Node State
--    State of the group on each node participating in the group.
--
--
   resGroupNodeStateOBJECT-TYPE
      SYNTAXINTEGER {
               online(2),
               offline(4),
               unknown(8),
               acquiring(16),
               releasing(32),
               error(64),
                     onlineSec (256),
                     acquiringSec (1024),
                     releasingSec (4096),
                     errorsec (16384),
                     offlineDueToFallover (65536),
                     offlineDueToParentOff (131072),
                     offlineDueToLackOfNode (262144),
                     unmanaged(524288),
                     unmanagedSec(1048576)
                     -- offlineDueToNodeForcedDown(2097152)

         }
```

```
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The State of the Resource Group"
      ::= { resGroupNodeEntry 3 }

   trapRGState       TRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ resGroupNodeGroupId, resGroupNodeId,
              resGroupNodeState, clusterId, clusterNodeId }
      DESCRIPTION
         "Fires whenever a resource group changes
         state on a particular node."
      ::= 23

--
--
-- XIII. The clsmuxpd group
-- Various statistics maintained by the smux peer daemon.
--
-- A) clsmuxpdGets
--    Incremented on each get request.
--
   clsmuxpdGetsOBJECT-TYPE
      SYNTAXCounter
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "Number of get requests received"
      ::= { clsmuxpd 1 }
--
-- B) clsmuxpdGetNexts
--    Incremented on each get-next request.
--
   clsmuxpdGetNextsOBJECT-TYPE
      SYNTAXCounter
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "Number of get-next requests received"
      ::= { clsmuxpd 2 }
--
-- C) clsmuxpdSets
--    Incremented on each set request.
--    Note that the smux does not durrently support set requests.
--
   clsmuxpdSetsOBJECT-TYPE
      SYNTAXCounter
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "Number of set requests received"
      ::= { clsmuxpd 3 }
--
-- D) clsmuxpdTraps
```

```
--    Incremented after a trap is generated.
--
   clsmuxpdTrapsOBJECT-TYPE
      SYNTAXCounter
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "Number of traps sent"
      ::= { clsmuxpd 4 }
--
-- E) clsmuxpdErrors
--    Incremented after an error occurs.
--
   clsmuxpdErrorsOBJECT-TYPE
      SYNTAXCounter
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "Number of errors encountered"
      ::= { clsmuxpd 5 }
--
-- F) clsmuxpdVersion
--    Version number of clsmuxpd program.
--
   clsmuxpdVersionOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "Version of clsmuxpd program"
      ::= { clsmuxpd 6 }
--
--
-- XIV. The event group
--     This is a list of the last one thousand events called
--     by the Cluster Manager. This list is used for tracking
--     cluster event history.
--
-- A) eventPtr
--    Points to the most recent event.
--
   eventPtrOBJECT-TYPE
      SYNTAXCounter
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "Pointer to the most recent event"
      ::= { event 1 }
--
-- B) The event table
--    This is a variable length table which is indexed by
--    a counter.  Useful for keeping history of events.
--
   eventTableOBJECT-TYPE
      SYNTAXSEQUENCE OF EventType
```

```
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "A series of cluster events"
      ::= { event 2 }
--
   eventTypeOBJECT-TYPE
      SYNTAXEventType
      ACCESSnot-accessible
      STATUSmandatory
      INDEX{ nodeId, eventCount }
      ::= { eventTable 1 }
--
   EventType::= SEQUENCE {
      eventIdINTEGER,
      eventNodeIdINTEGER,
      eventNetIdINTEGER,
      eventTimeINTEGER,
      eventCountCounter,
      eventNodeNameDisplayString
   }
--
-- C) eventId
--    This field is returned by the cluster manager.
--    Note that the following events are no longer used:
--       unstableTooLong(18)
--
   eventIdOBJECT-TYPE
      SYNTAXINTEGER { swapAdapter(0), swapAdapterComplete(1),
          joinNetwork(2), failNetwork(3),
          joinNetworkComplete(4), failNetworkComplete(5),
          joinNode(6), failNode(7),
          joinNodeComplete(8), failNodeComplete(9),
          joinStandby(10), failStandby(11),
          newPrimary(12),
          clusterUnstable(13), clusterStable(14),
          configStart(15), configComplete(16),
          configTooLong(17), unstableTooLong(18),
          eventError(19),
          dareConfiguration(20),
          dareTopologyStart(21), dareConfigurationComplete(22),
          dareResource(23), dareResourceRelease(24),
          dareResourceAcquire(25), dareResourceComplete(26),
          resourceGroupChange(27),
          joinInterface(28), failInterface(29),
          wait(30), waitComplete(31),
          migrate(32), migrateComplete(33),
          rgMove(34),
          serverRestart(35), serverDown(36),
          siteUp(37), siteDown(38),
             siteUpComplete(39), siteDownComplete(40),
          siteMerge(41), siteIsolation(42),
          siteMergeComplete(43), siteIsolationComplete(44),
          nullEvent(45), externalEvent(46),
          refresh(47), topologyRefresh(48),
```

```
         clusterNotify(49),
         resourceStateChange(50), resourceStateChangeComplete(51),
         externalResourceStateChange(52), externalResourceStateChangeComplete(53)
      }
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The cluster event"
      ::= { eventType 1 }
--
-- D) eventNodeId
--    This field is returned by the cluster manager.
--
   eventNodeIdOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The ID of the Node on which the event occurs"
      ::= { eventType 2 }
--
-- E) eventNetId
--    This field is returned by the cluster manager.
--
   eventNetIdOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The ID of the Network on which the event occurs"
      ::= { eventType 3 }
--
-- F) eventTime
--    This field is an integer string returned by the gettimeofday()
--              library call and is updated whenever an event is received.
--
   eventTimeOBJECT-TYPE
      SYNTAXCounter
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The time at which the event occurred"
      ::= { eventType 4 }
--
-- G) eventCount
--    This field is incremented whenever an event is received.
--
   eventCountOBJECT-TYPE
      SYNTAXCounter
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "A count of the event used for indexing into the table"
      ::= { eventType 5 }
--
```

```
-- H) eventNodeName
--    This field is returned by the cluster manager.
--
   eventNodeNameOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The name of the Node on which the event occurs"
      ::= { eventType 6 }

--
-- State Event traps
--
   trapSwapAdapterTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ nodeName, clusterName, addrLabel, eventCount }
      DESCRIPTION
      "Specified node generated swap adapter event"
      ::= 64

   trapSwapAdapterCompleteTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ nodeName, clusterName, netName, eventCount }
      DESCRIPTION
      "Specified node generated swap adapter complete event"
      ::= 65

   trapJoinNetworkTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ nodeName, clusterName, netName, eventCount }
      DESCRIPTION
      "Specified node has joined the network"
      ::= 66

   trapFailNetworkTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ nodeName, clusterName, netName, eventCount }
      DESCRIPTION
      "Specified node generated fail network event"
      ::= 67

   trapJoinNetworkCompleteTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ nodeName, clusterName, netName, eventCount }
      DESCRIPTION
      "Specified node generated join network complete event"
      ::= 68

   trapFailNetworkCompleteTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ nodeName, clusterName, netName, eventCount }
      DESCRIPTION
      "Specified node generated fail network complete event"
      ::= 69
```

```
trapJoinNodeTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Specified node generated join node event"
   ::= 70

trapFailNodeTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Specified node generated fail join node event"
   ::= 71

trapJoinNodeCompleteTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Specified node generated join node complete event"
   ::= 72

trapFailNodeCompleteTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName , eventCount}
   DESCRIPTION
   "Specified node generated fail node complete event"
   ::= 73

trapJoinStandbyTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Specified node generated join standby event"
   ::= 74

trapFailStandbyTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Specified node has failed standby adapter"
   ::= 75

trapEventNewPrimaryTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, clusterPrimaryNodeName, eventCount}
   DESCRIPTION
   "Specified node has become the new primary"
   ::= 76

trapClusterUnstableTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Specified node generated cluster unstable event"
```

```
    ::= 77

  trapClusterStableTRAP-TYPE
    ENTERPRISErisc6000clsmuxpd
    VARIABLES{ nodeName, clusterName, netName, eventCount }
    DESCRIPTION
    "Specified node generated cluster stable event"
    ::= 78

  trapConfigStartTRAP-TYPE
    ENTERPRISErisc6000clsmuxpd
    VARIABLES{ nodeName, clusterName, netName, eventCount }
    DESCRIPTION
    "Configuration procedure has started for specified node"
    ::= 79

  trapConfigCompleteTRAP-TYPE
    ENTERPRISErisc6000clsmuxpd
    VARIABLES{ nodeName, clusterName, netName, eventCount }
    DESCRIPTION
    "Configuration procedure has completed for specified node"
    ::= 80

  trapClusterConfigTooLongTRAP-TYPE
    ENTERPRISErisc6000clsmuxpd
    VARIABLES{ nodeName, clusterName, netName, eventCount }
    DESCRIPTION
    "Specified node has been in configuration too long"
    ::= 81

--
-- Note that this event is no longer used and this trap will never occur.
--
  trapClusterUnstableTooLongTRAP-TYPE
    ENTERPRISErisc6000clsmuxpd
    VARIABLES{ nodeName, clusterName, netName, eventCount }
    DESCRIPTION
    "Specified node has been unstable too long"
    ::= 82

  trapEventErrorTRAP-TYPE
    ENTERPRISErisc6000clsmuxpd
    VARIABLES{ nodeName, clusterName, netName, eventCount }
    DESCRIPTION
    "Specified node generated an event error"
    ::= 83

  trapDareTopologyTRAP-TYPE
    ENTERPRISErisc6000clsmuxpd
    VARIABLES{ nodeName, clusterName, netName, eventCount }
    DESCRIPTION
    "Dynamic reconfiguration event for topology has been issued"
    ::= 84

  trapDareTopologyStartTRAP-TYPE
```

```
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Dynamic reconfiguration event for topology has started"
   ::= 85

trapDareTopologyCompleteTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Dynamic reconfiguration event for topology has completed"
   ::= 86

trapDareResourceTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Dynamic reconfiguration event for resource has been issued"
   ::= 87

trapDareResourceReleaseTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Dynamic reconfiguration event for resource has been released"
   ::= 88

trapDareResourceAcquireTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Dynamic reconfiguration event for resource has been acquired"
   ::= 89

trapDareResourceCompleteTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Dynamic reconfiguration event for resource has completed"
   ::= 90

trapFailInterfaceTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Interface has failed on the event node"
   ::= 91

trapJoinInterfaceTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Interface has joined on the event node"
   ::= 92
```

```
trapResourceGroupChangeTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "rg_move event has occurred on the event node"
   ::= 93

trapServerRestart TRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Server has been restarted on the event node"
   ::= 94

trapServerRestartComplete TRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Server restart is complete on the event node"
   ::= 95

trapServerDown TRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Server has failed on the event node"
   ::= 96

trapServerDownComplete TRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, netName, eventCount }
   DESCRIPTION
   "Server has failed on the event node"
   ::= 97

trapSiteDown TRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, siteName, eventCount }
   DESCRIPTION
   "Site failed"
   ::= 98

trapSiteDownComplete TRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, siteName, eventCount }
   DESCRIPTION
   "Site failure complete on the event site"
   ::= 99

trapSiteUp TRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, siteName, eventCount }
   DESCRIPTION
   "Site is now up"
   ::= 100
```

```
trapSiteUpComplete TRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, siteName, eventCount }
   DESCRIPTION
   "Site join is complete on the event site"
   ::= 101

trapSiteMerge TRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, siteName, eventCount }
   DESCRIPTION
   "Site has merged with the active site"
   ::= 102

trapSiteMergeComplete TRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, siteName, eventCount }
   DESCRIPTION
   "Site merge is complete on the event site"
   ::= 103

trapSiteIsolation TRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, siteName, eventCount }
   DESCRIPTION
   "Site is isolated"
   ::= 104

trapSiteIsolationComplete TRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, siteName, eventCount }
   DESCRIPTION
   "Site isoaltion is complete on the event site"
   ::= 105

trapClusterNotify TRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, siteName, eventCount }
   DESCRIPTION
   "Cluster Notify event has occurred on event node"
   ::= 106

trapResourceStateChangeTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, siteName, eventCount }
   DESCRIPTION
   "Resource State Change event has occurred on event node"
   ::= 107

trapResourceStateChangeCompleteTRAP-TYPE
   ENTERPRISErisc6000clsmuxpd
   VARIABLES{ nodeName, clusterName, siteName, eventCount }
   DESCRIPTION
   "Resource State Change Complete event has occurred on event node"
```

```
      ::= 108

   trapExternalResourceStateChangeTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ nodeName, clusterName, siteName, eventCount }
      DESCRIPTION
      "External Resource State Change event has occurred on event node"
      ::= 109

   trapExternalResourceStateChangeCompleteTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ nodeName, clusterName, siteName, eventCount }
      DESCRIPTION
      "External Resource State Change Complete event has occurred on event node"
      ::= 110



--
-- XV. The Resource Group Dependency Configuration
-- Contains information about cluster resources group dependencies.
--
-- A) The Resource Group Dependency Table
--
   resGroupDependencyTableOBJECT-TYPE
      SYNTAXSEQUENCE OF ResGroupDependencyEntry
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "A series of Resource Group Dependency descriptions"
      ::= { resmanager 4 }
--
   resGroupDependencyEntryOBJECT-TYPE
      SYNTAXResGroupDependencyEntry
      ACCESSnot-accessible
      STATUSmandatory
      DESCRIPTION
         "Individual Resource Group Dependency description"
      INDEX { resGroupDependencyId }
      ::= { resGroupDependencyTable 1 }
--
   ResGroupDependencyEntry::= SEQUENCE {
      resGroupDependencyIdINTEGER,
      resGroupNameParentDisplayString,
      resGroupNameChildDisplayString,
      resGroupDependencyTypeDisplayString,
      resGroupDependencyTypeIntINTEGER
   }

--
--        B) Resource Group Dependency Id
   resGroupDependencyIdOBJECT-TYPE
      SYNTAXINTEGER
      ACCESSread-only
      STATUSmandatory
```

```
      DESCRIPTION
         "The ID of the Resource Group Dependency"
      ::= { resGroupDependencyEntry 1 }

   trapRGDepAddTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ resGroupDependencyId }
      DESCRIPTION
         "Fires when a new resource group dependency is added."
      ::= 30

   trapRGDepDelTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ resGroupDependencyId }
      DESCRIPTION
         "Fires when a new resource group dependency is deleted."
      ::= 31

   trapRGDepChangeTRAP-TYPE
      ENTERPRISErisc6000clsmuxpd
      VARIABLES{ resGroupDependencyId, resGroupNameParent,
               resGroupNameChild, resGroupDependencyType,
               resGroupDependencyTypeInt }
      DESCRIPTION
         "Fires when an resource group dependency is changed."
      ::= 32

--
-- C) Resource Group Name Parent
--
--
   resGroupNameParentOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The name of the Parent Resource Group"
      ::= { resGroupDependencyEntry 2 }



--
-- D) Resource Group Name
--
--
   resGroupNameChildOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The name of the Child Resource Group"
      ::= { resGroupDependencyEntry 3 }

--
-- E) Resource Group Dependency Type
--
```

```
--
   resGroupDependencyTypeOBJECT-TYPE
      SYNTAXDisplayString
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The type of the resource group dependency."
      ::= { resGroupDependencyEntry 4 }




--
-- F) Resource Group Dependency Policy
--
--
   resGroupDependencyTypeIntOBJECT-TYPE
      SYNTAXINTEGER {
               globalOnline(0)
            }
      ACCESSread-only
      STATUSmandatory
      DESCRIPTION
         "The type of the Resource Group Dependency"
      ::= { resGroupDependencyEntry 5 }


--
--
-- XVI. The address6 group
--
--  A) The address6 table
--      This is a variable length table which is indexed by
--      the node Id, inet_type, octet_count, ip address (in octet form) and
--      prefix length.
--
    addr6Table   OBJECT-TYPE
        SYNTAX  SEQUENCE OF Addr6Entry
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
            "A series of IPv4/v6 address descriptions"
        ::= { address6 1 }
--
    addr6Entry   OBJECT-TYPE
        SYNTAX  Addr6Entry
        ACCESS  not-accessible
        STATUS  mandatory
        INDEX   { addr6NodeId, addr6InetType, addr6OctetCount, addr6Address,
addr6PrefixLength }
        ::= { addr6Table 1 }
--
    Addr6Entry   ::= SEQUENCE {
        addr6NodeId    INTEGER,
        addr6InetType  INTEGER,
```

```
        addr6OctetCount INTEGER,
        addr6Address   OCTET STRING(SIZE (20)),
        addr6PrefixLength INTEGER,
        addr6Label     DisplayString,
        addr6Role      INTEGER,
        addr6NetId     INTEGER,
        addr6State     INTEGER,
        addr6ActiveNode    INTEGER,
        oldAddr6ActiveNode INTEGER
    }
--
--  B) addr6NodeId
--      This field is read from the HACMP for AIX object repository.
--
    addr6NodeId  OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
            "The ID of the Node this IP address is configured"
        ::= { addr6Entry 1 }
--
--  C) addr6InetType
--          A value that represents a type of Internet address.
--
    addr6InetType OBJECT-TYPE
        SYNTAX  INTEGER { unknown(0), ipv4(1), ipv6(2), ipv4z(3), ipv6z(4),
dns(16) }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The internet address type of addrAddress"
        ::= { addr6Entry 2 }
--
--  D) addr6OctetCount
--          A value that represents number of octets in addrAddress.
--
    addr6OctetCount OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of octets in addrAddress"
        ::= { addr6Entry 3 }
--
--  E) addr6Address
--      This field is read from the HACMP for AIX object repository.
--
    addr6Address OBJECT-TYPE
        SYNTAX  OCTET STRING(SIZE (20))
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
            "The IP address"
        ::= { addr6Entry 4 }
```

```
--
-- F) addr6PrefixLength
--         A value that represents number of octets in addrAddress.
--
   addr6PrefixLength OBJECT-TYPE
       SYNTAX  INTEGER
       ACCESS  read-only
       STATUS  mandatory
       DESCRIPTION
               "The prefix length"
       ::= { addr6Entry 5 }
--
-- G) addr6Label
--     This field is read from the HACMP for AIX object repository.
--
   addr6Label   OBJECT-TYPE
       SYNTAX  DisplayString
       ACCESS  read-only
       STATUS  mandatory
       DESCRIPTION
           "The IP label associated with the IP address"
       ::= { addr6Entry 6 }
--
-- H) addr6Role
--     This field is read from the HACMP for AIX object repository.
--     Note that use of sharedService and standby is deprecated.
--
   addr6Role    OBJECT-TYPE
       SYNTAX  INTEGER { boot(64), service(16),
                 persistent(8),
                 sharedService(128), standby(32) }
       ACCESS  read-only
       STATUS  mandatory
       DESCRIPTION
           "The role of the IP address"
       ::= { addr6Entry 7 }
--
-- I) addr6NetId
--     This field is read from the HACMP for AIX object repository.
--     It is provide so that clients can determine the corresponding
--     index into the network table.
--
   addr6NetId   OBJECT-TYPE
       SYNTAX  INTEGER
       ACCESS  read-only
       STATUS  mandatory
       DESCRIPTION
           "The network ID of the IP address"
       ::= { addr6Entry 8 }
--
-- J) addr6State
--     This field is returned from the Cluster Manager.
--
   addr6State   OBJECT-TYPE
       SYNTAX  INTEGER { up(2), down(4), unknown(8) }
```

```
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
            "The state of the IP address"
        ::= { addr6Entry 9 }

    trapAddressState    TRAP-TYPE
        ENTERPRISE  risc6000clsmuxpd
        VARIABLES   { addr6State, addr6NetId, clusterId, clusterNodeId }
        DESCRIPTION
            "Fires whenever a address changes state."
        ::= 14

    trapAdapterSwap TRAP-TYPE
        ENTERPRISE  risc6000clsmuxpd
        VARIABLES   { addr6State, clusterId, clusterNodeId }
        DESCRIPTION
            "Fires whenever a address swap occurs."
        ::= 17
--
--  K) addr6ActiveNode
--      This field is returned from the Cluster Manager.
--
    addr6ActiveNode  OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
            "The ID of the Node on which this IP address is active"
        ::= { addr6Entry 10 }
--
--  L) oldAddr6ActiveNode
--          This field is returned from the Cluster Manager.
--
    oldAddr6ActiveNode OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  not-accessible
        STATUS  mandatory
        DESCRIPTION
                "The ID of the Node on which this IP address was previously
active"
        ::= { addr6Entry 11 }

    trapAddressTakeover  TRAP-TYPE
        ENTERPRISE   risc6000clsmuxpd
        VARIABLES    { addr6ActiveNode, oldAddr6ActiveNode,
                       clusterId, clusterNodeId }
        DESCRIPTION
                     "Fires whenever IP address takeover occurs."
        ::= 19

   END
```

Tivoli Monitoring Universal Agent metafile for PowerHA

The PowerHA Management Information Base (MIB) V1.31(`/usr/es/sbin/cluster/hacmp.my`) provided in Example B-5 on page 437 must be translated into a data definition metafile for use in the IBM Tivoli Monitoring (ITM) Universal Agent model of SNMP trap monitoring.

You may use externally available tools for the conversion of MIB to the Tivoli Monitoring MDL file. The MibUtility, which is available in OPAL, is a common tool that you can use for the conversion. Alternatively, you can write your own definition file, based on your understanding of the MIB.

Example B-6 shows a sample data definition metafile (`PowerHA.mdl`) that may be loaded into Tivoli Monitoring for PowerHA SNMP monitoring.

Example B-6 PowerHA.mdl

```
* ------------------------------------------------------------------
* mibutil_risc6000clsmuxpd

* Universal Agent Application Definition

* Licensed Materials - Property of IBM

* Copyright IBM Corp. 2006 All Rights Reserved

* US Government Users Restricted Rights - Use, duplication or

* disclosure restricted by GSA ADP Schedule Contract with

* IBM Corp.

* This file was created by the IBM Tivoli Monitoring Agent Builder

* Version 6.1.0

* Build Level agent_fac 200612071421

* ------------------------------------------------------------------
//SNMP TEXT

//APPL RISC6000CLSMUXPD risc6000clsmuxpd 1.3.6.1.4.1.2.3.1.2.1 @SNMP application
for enterprise MIB risc6000clsmuxpd

//NAME CLUSTER K 3600 @Data gathered from SNMP Object cluster

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.

clusterId C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.1.1.0 @The ID of the cluster

clusterName D 64 1.3.6.1.4.1.2.3.1.2.1.5.1.2.0 @User configurable cluster Name
```

```
clusterConfiguration D 64 1.3.6.1.4.1.2.3.1.2.1.5.1.3.0 @The cluster configuration

clusterState G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.1.4.0 ENUM{ up(2) down(4)
unknown(8) notconfigured(256)} @The cluster status

clusterPrimary C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.1.5.0 @The Node ID of the
Primary Lock Manager

clusterLastChange C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.1.6.0 @Time in seconds of
last change in this cluster.

clusterGmtOffset C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.1.7.0 @Seconds west of GMT
for the time of last change in this cluster.

clusterSubState G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.1.8.0 ENUM{ unstable(16)
error(64) stable(32) unknown(8) reconfig(128) notconfigured(256) notsynced(512)}
@The cluster substate

clusterNodeName D 64 1.3.6.1.4.1.2.3.1.2.1.5.1.9.0 @User configurable cluster
local node name

clusterPrimaryNodeName D 64 1.3.6.1.4.1.2.3.1.2.1.5.1.10.0 @The Node Name of the
primary cluster node

clusterNumNodes C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.1.11.0 @The number of nodes
in the cluster

clusterNodeId C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.1.12.0 @The ID of the local
node

clusterNumSites C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.1.13.0 @The number of sites
in the cluster

//NAME NODETABLE K 3600 @Data gathered from SNMP Object nodeTable

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.

nodeId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.2.1.1.1 @The ID of the Node

nodeState G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.2.1.1.2 ENUM{ up(2) down(4)
joining(32) leaving(64)} @The State of the Node

nodeNumIf C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.2.1.1.3 @The number of network
interfaces in this node

nodeName D 64 1.3.6.1.4.1.2.3.1.2.1.5.2.1.1.4 @The name of this node

nodeSite D 64 1.3.6.1.4.1.2.3.1.2.1.5.2.1.1.5 @The site associated with this node
```

```
//NAME ADDRTABLE K 3600 @Data gathered from SNMP Object addrTable

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.

addrNodeId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.3.1.1.1 @The ID of the Node
this IP address is configured

addrAddress D 32 KEY 1.3.6.1.4.1.2.3.1.2.1.5.3.1.1.2 @The IP address

addrLabel D 64 1.3.6.1.4.1.2.3.1.2.1.5.3.1.1.3 @The IP label associated with the
IP address

addrRole G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.3.1.1.4 ENUM{ boot(64) service(16)
persistent(8) sharedService(128) standby(32)} @The role of the IP address

addrNetId C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.3.1.1.5 @The network ID of the IP
address

addrState G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.3.1.1.6 ENUM{ up(2) down(4)
unknown(8)} @The state of the IP address

addrActiveNode C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.3.1.1.7 @The ID of the Node on
which this IP address is active

//NAME NETTABLE K 3600 @Data gathered from SNMP Object netTable

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.

netNodeId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.4.1.1.1 @The ID of the Node
this network is configured

netId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.4.1.1.2 @The ID of the network

netName D 64 1.3.6.1.4.1.2.3.1.2.1.5.4.1.1.3 @The name of network

netAttribute G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.4.1.1.4 ENUM{ public(2)
private(1) serial(4)} @The attribute of the network.

netState G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.4.1.1.5 ENUM{ up(2) down(4)
joining(32) leaving(64)} @The state of the network

netODMid C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.4.1.1.6 @The ODM id of the network
```

```
netType D 64 1.3.6.1.4.1.2.3.1.2.1.5.4.1.1.7 @The physical network type, e.g.
ethernet

netFamily G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.4.1.1.8 ENUM{ unknown(0) clinet(1)
clinet6(2) clhybrid(3)} @Family of the network.

//NAME CLSTRMGRTABLE K 3600 @Data gathered from SNMP Object clstrmgrTable

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.

clstrmgrNodeId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.5.1.1.1 @The node ID of
the Cluster Manager
clstrmgrVersion D 64 1.3.6.1.4.1.2.3.1.2.1.5.5.1.1.2 @The version of the Cluster
Manager

clstrmgrStatus G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.5.1.1.3 ENUM{ up(2) down(4)
suspended(16) unknown(8) graceful(32) forced(64) takeover(128)} @The state of the
Cluster Manager

//NAME CLLOCKDTABLE K 3600 @Data gathered from SNMP Object cllockdTable

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.

cllockdNodeId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.6.1.1.1 @The node ID of the
Lock Manager

cllockdVersion D 64 1.3.6.1.4.1.2.3.1.2.1.5.6.1.1.2 @The version of the Lock
Manager

cllockdStatus G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.6.1.1.3 ENUM{ up(2) down(4)
unknown(8) suspended(16) stalled(256)} @The state of the Lock Manager

//NAME CLINFOTABLE K 3600 @Data gathered from SNMP Object clinfoTable

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.
```

```
clinfoNodeId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.7.1.1.1 @The node ID running
the Client Information Daemon

clinfoVersion D 64 1.3.6.1.4.1.2.3.1.2.1.5.7.1.1.2 @The version of the Client
Information Daemon

clinfoStatus G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.7.1.1.3 ENUM{ up(2) down(4)
unknown(8) suspended(16)} @The state of the Client Information Daemon

//NAME APPTABLE K 3600 @Data gathered from SNMP Object appTable

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.

appNodeId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.8.1.1.1 @The node ID of the
application

appPid C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.8.1.1.2 @The process ID of the
application

appName D 64 1.3.6.1.4.1.2.3.1.2.1.5.8.1.1.3 @The name of the application

appVersion D 64 1.3.6.1.4.1.2.3.1.2.1.5.8.1.1.4 @The version of the application

appDescr D 64 1.3.6.1.4.1.2.3.1.2.1.5.8.1.1.5 @The description of the application

//NAME CLSMUXPD K 3600 @Data gathered from SNMP Object clsmuxpd

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.

clsmuxpdGets C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.9.1.0 @Number of get requests
received

clsmuxpdGetNexts C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.9.2.0 @Number of get-next
requests received

clsmuxpdSets C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.9.3.0 @Number of set requests
received

clsmuxpdTraps C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.9.4.0 @Number of traps sent

clsmuxpdErrors C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.9.5.0 @Number of errors
encountered
```

```
clsmuxpdVersion D 64 1.3.6.1.4.1.2.3.1.2.1.5.9.6.0 @Version of clsmuxpd program

//NAME EVENTTABLE K 3600 @Data gathered from SNMP Object eventTable

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.

eventId G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.10.2.1.1 ENUM{ swapAdapter(0)
swapAdapterComplete(1) joinNetwork(2) failNetwork(3) joinNetworkComplete(4)
failNetworkComplete(5) joinNode(6) failNode(7) joinNodeComplete(8)
failNodeComplete(9) joinStandby(10) failStandby(11) newPrimary(12)
clusterUnstable(13) clusterStable(14) configStart(15) configComplete(16)
configTooLong(17) unstableTooLong(18) eventError(19) dareConfiguration(20)
dareTopologyStart(21) dareConfigurationComplete(22) dareResource(23)
dareResourceRelease(24) dareResourceAcquire(25) dareResourceComplete(26)
resourceGroupChange(27) joinInterface(28) failInterface(29) wait(30)
waitComplete(31) migrate(32) migrateComplete(33) rgMove(34) serverRestart(35)
serverDown(36) siteUp(37) siteDown(38) siteUpComplete(39) siteDownComplete(40)
siteMerge(41) siteIsolation(42) siteMergeComplete(43) siteIsolationComplete(44)
nullEvent(45) externalEvent(46) refresh(47) topologyRefresh(48) clusterNotify(49)
resourceStateChange(50) resourceStateChangeComplete(51)
externalResourceStateChange(52) externalResourceStateChangeComplete(53)} @The
cluster event

eventNodeId C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.10.2.1.2 @The ID of the Node on
which the event occurs

eventNetId C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.10.2.1.3 @The ID of the Network on
which the event occurs

eventTime C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.10.2.1.4 @The time at which the
event occurred

eventCount C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.10.2.1.5 @A count of the event
used for indexing into the table

eventNodeName D 64 1.3.6.1.4.1.2.3.1.2.1.5.10.2.1.6 @The name of the Node on which
the event occurs

//NAME TRAPCLUSTERSTATE K 3600 @Data gathered from SNMP Object trapClusterState

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.
```

```
eventPtr C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.10.1.0 @Pointer to the most recent
event

//NAME RESGROUPTABLE K 3600 @Data gathered from SNMP Object resGroupTable

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.

resGroupId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.11.1.1.1 @The ID of the
Resource Group

resGroupName D 64 1.3.6.1.4.1.2.3.1.2.1.5.11.1.1.2 @The name of the Resource Group
resGroupPolicy G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.11.1.1.3 ENUM{ cascading(1)
rotating(2) concurrent(3) userdefined(4) custom(5)} @The State of the Resource
Group

resGroupUserPolicyName D 64 1.3.6.1.4.1.2.3.1.2.1.5.11.1.1.4 @The name of the
user-defined policy

resGroupNumResources C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.11.1.1.5 @The number of
resources defined in the group

resGroupNumNodes C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.11.1.1.6 @The number of
participating nodes in the group

resGroupStartupPolicy C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.11.1.1.7 @The Resource
Group's Startup Policy This can have the following values Online On HomeNode Only
- 1 Online On First Available Node - 2 Online Using Distribution Policy - 3 Online
On All Available Nodes - 4

resGroupFalloverPolicy C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.11.1.1.8 @The Resource
Group's Fallover Policy This can have the following values Fallover To Next
Priority Node On the List - 5 Fallover Using DNP - 6 Bring Offline - 7

resGroupFallbackPolicy C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.11.1.1.9 @The Resource
Group's Fallback Policy Fallback to Higher Priority Node in the List - 8 Never
Fallback - 9

//NAME RESTABLE K 3600 @Data gathered from SNMP Object resTable

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.

resourceGroupId C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.1 @The ID of the
resource group
```

```
resourceId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.2 @The ID of the
Resource

resourceName D 64 1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.3 @The name of this resource
resourceType G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.11.2.1.4 ENUM{
serviceLabel(1000) iPLabel(1000) htyServiceLabel(1001) fileSystem(1002)
volumeGroup(1003) disk(1004) rawDiskPVID(1004) aixConnectionServices(1005)
application(1006) concurrentVolumeGroup(1007) haCommunicationLinks(1008)
haFastConnectServices(1009)} @The Type of the Resource

//NAME RESGROUPNODETABLE K 3600 @Data gathered from SNMP Object resGroupNodeTable

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.

resGroupNodeGroupId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.11.3.1.1 @The ID of
the resource group

resGroupNodeId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.11.3.1.2 @Node ID of node
located within resource group

resGroupNodeState G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.11.3.1.3 ENUM{ online(2)
offline(4) unknown(8) acquiring(16) releasing(32) error(64) onlineSec(256)
acquiringSec(1024) releasingSec(4096) errorsec(16384) offlineDueToFallover(65536)
offlineDueToParentOff(131072) offlineDueToLackOfNode(262144) unmanaged(524288)
unmanagedSec(1048576)} @The State of the Resource Group

//NAME RESGROUPDEPENDENCYTABLE K 3600 @Data gathered from SNMP Object
resGroupDependencyTable

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.

resGroupDependencyId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.11.4.1.1 @The ID of
the Resource Group Dependency

resGroupNameParent D 64 1.3.6.1.4.1.2.3.1.2.1.5.11.4.1.2 @The name of the Parent
Resource Group

resGroupNameChild D 64 1.3.6.1.4.1.2.3.1.2.1.5.11.4.1.3 @The name of the Child
Resource Group

resGroupDependencyType D 64 1.3.6.1.4.1.2.3.1.2.1.5.11.4.1.4 @The type of the
resource group dependency.
```

```
resGroupDependencyTypeInt G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.11.4.1.5 ENUM{
globalOnline(0)} @The type of the Resource Group Dependency

//NAME SITETABLE K 3600 @Data gathered from SNMP Object siteTable

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.

siteId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.12.1.1.1 @The ID of the site

siteName D 64 1.3.6.1.4.1.2.3.1.2.1.5.12.1.1.2 @The name of this site

sitePriority G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.12.1.1.3 ENUM{ primary(1)
secondary(2) tertiary(4)} @The Priority of the Site

siteBackup G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.12.1.1.4 ENUM{ none(1) dbfs(2)
sgn(4)} @Backup communications method for the site

siteNumNodes C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.12.1.1.5 @The number of nodes in
this site

siteState G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.12.1.1.6 ENUM{ up(2) down(4)
joining(16) leaving(32) isolated(257)} @The State of the site

//NAME SITENODETABLE K 3600 @Data gathered from SNMP Object siteNodeTable

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.

siteNodeSiteId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.12.2.1.1 @The ID of the
cluster site

siteNodeNodeId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.12.2.1.2 @The node ID of
the node in this site

//NAME ADDR6TABLE K 3600 @Data gathered from SNMP Object addr6Table

//ATTRIBUTES

Agent_Info D 128 0.0 @Identifies the SNMP host name and community names for agents
to query.

Agent_Name D 64 KEY 0.0 @Identifies the SNMP host name relating to a particular
sample of data.
```

```
addr6NodeId C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.13.1.1.1 @The ID of the Node
this IP address is configured

addr6InetType G 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.13.1.1.2 ENUM{ unknown(0)
ipv4(1) ipv6(2) ipv4z(3) ipv6z(4) dns(16)} @The internet address type of
addrAddress

addr6OctetCount C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.13.1.1.3 @The number of
octets in addrAddress

addr6Address D 20 KEY 1.3.6.1.4.1.2.3.1.2.1.5.13.1.1.4 @The IP address

addr6PrefixLength C 2147483647 KEY 1.3.6.1.4.1.2.3.1.2.1.5.13.1.1.5 @The prefix
length

addr6Label D 64 1.3.6.1.4.1.2.3.1.2.1.5.13.1.1.6 @The IP label associated with the
IP address

addr6Role G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.13.1.1.7 ENUM{ boot(64) service(16)
persistent(8) sharedService(128) standby(32)} @The role of the IP address

addr6NetId C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.13.1.1.8 @The network ID of the IP
address

addr6State G 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.13.1.1.9 ENUM{ up(2) down(4)
unknown(8)} @The state of the IP address

addr6ActiveNode C 2147483647 1.3.6.1.4.1.2.3.1.2.1.5.13.1.1.10 @The ID of the Node
on which this IP address is active
```

Tivoli Monitoring Universal Agent TRAPCNFG for PowerHA SNMP monitoring

Now that you have converted the PowerHA MIB file to a data definition metafile for parsing the information into attributes and attribute groups, as shown in Example B-6 on page 480, you must define the trap configuration in the Tivoli Monitoring Universal Agent to receive and parse appropriate SNMP traps that are received from the PowerHA cluster nodes. Example B-7 shows a sample TRAPCNFG file.

Example B-7 TRAPCNFG configuration (/opt/IBM/ITM/aix526/um/work/TRAPCNFG)

```
risc6000clsmuxpd {1.3.6.1.4.1.2.3.1.2.1.5}
trapClusterSubState {1.3.6.1.4.1.2.3.1.2.1.5} 6 11 A 1 0 "Status Events"
SDESC
Fires whenever the cluster changes substate.
EDESC
trapClusterStable {1.3.6.1.4.1.2.3.1.2.1.5} 6 78 A 1 0 "Status Events"
SDESC
Specified node generated cluster stable event
EDESC
trapFailNetworkComplete {1.3.6.1.4.1.2.3.1.2.1.5} 6 69 A 1 0 "Status Events"
SDESC
```

```
Specified node generated fail network complete event
EDESC
trapRGState {1.3.6.1.4.1.2.3.1.2.1.5} 6 23 A 1 0 "Status Events"
SDESC
Fires whenever a resource group changesstate on a particular node.
EDESC
trapDareTopology {1.3.6.1.4.1.2.3.1.2.1.5} 6 84 A 1 0 "Status Events"
SDESC
Dynamic reconfiguration event for topology has been issued
EDESC
trapSiteIsolation {1.3.6.1.4.1.2.3.1.2.1.5} 6 104 A 1 0 "Status Events"
SDESC
Site is isolated
EDESC
trapAppState {1.3.6.1.4.1.2.3.1.2.1.5} 6 16 A 1 0 "Status Events"
SDESC
Fires whenever an application is added or deleted.
EDESC
trapSiteState {1.3.6.1.4.1.2.3.1.2.1.5} 6 18 A 1 0 "Status Events"
SDESC
Fires whenever a site changes state.
EDESC
trapSwapAdapter {1.3.6.1.4.1.2.3.1.2.1.5} 6 64 A 1 0 "Status Events"
SDESC
Specified node generated swap adapter event
EDESC
trapRGDel {1.3.6.1.4.1.2.3.1.2.1.5} 6 21 A 1 0 "Status Events"
SDESC
Fires whenever a resource group is deleted.
EDESC
trapResourceStateChange {1.3.6.1.4.1.2.3.1.2.1.5} 6 107 A 1 0 "Status Events"
SDESC
Resource State Change event has occurred on event node
EDESC
trapServerDown {1.3.6.1.4.1.2.3.1.2.1.5} 6 96 A 1 0 "Status Events"
SDESC
Server has failed on the event node
EDESC
trapFailStandby {1.3.6.1.4.1.2.3.1.2.1.5} 6 75 A 1 0 "Status Events"
SDESC
Specified node has failed standby adapter
EDESC
trapSiteDown {1.3.6.1.4.1.2.3.1.2.1.5} 6 98 A 1 0 "Status Events"
SDESC
Site failed
EDESC
trapRGDepDel {1.3.6.1.4.1.2.3.1.2.1.5} 6 31 A 1 0 "Status Events"
SDESC
Fires when a new resource group dependency is deleted.
EDESC
trapClusterConfigTooLong {1.3.6.1.4.1.2.3.1.2.1.5} 6 81 A 1 0 "Status Events"
SDESC
Specified node has been in configuration too long
EDESC
trapConfigStart {1.3.6.1.4.1.2.3.1.2.1.5} 6 79 A 1 0 "Status Events"
```

```
SDESC
Configuration procedure has started for specified node
EDESC
trapDareResource {1.3.6.1.4.1.2.3.1.2.1.5} 6 87 A 1 0 "Status Events"
SDESC
Dynamic reconfiguration event for resource has been issued
EDESC
trapDareResourceComplete {1.3.6.1.4.1.2.3.1.2.1.5} 6 90 A 1 0 "Status Events"
SDESC
Dynamic reconfiguration event for resource has completed
EDESC
trapRGChange {1.3.6.1.4.1.2.3.1.2.1.5} 6 22 A 1 0 "Status Events"
SDESC
Fires whenever the policy, number of nodes,or the number of resources of a
resourcegroup is changed.
EDESC
trapAddressTakeover {1.3.6.1.4.1.2.3.1.2.1.5} 6 19 A 1 0 "Status Events"
SDESC
Fires whenever IP address takeover occurs.
EDESC
trapExternalResourceStateChangeComplete {1.3.6.1.4.1.2.3.1.2.1.5} 6 110 A 1 0
"Status Events"
SDESC
External Resource State Change Complete event has occurred on event node
EDESC
trapDareTopologyComplete {1.3.6.1.4.1.2.3.1.2.1.5} 6 86 A 1 0 "Status Events"
SDESC
Dynamic reconfiguration event for topology has completed
EDESC
trapNetworkState {1.3.6.1.4.1.2.3.1.2.1.5} 6 13 A 1 0 "Status Events"
SDESC
Fires whenever a network changes state.
EDESC
trapSiteUpComplete {1.3.6.1.4.1.2.3.1.2.1.5} 6 101 A 1 0 "Status Events"
SDESC
Site join is complete on the event site
EDESC
trapDareResourceRelease {1.3.6.1.4.1.2.3.1.2.1.5} 6 88 A 1 0 "Status Events"
SDESC
Dynamic reconfiguration event for resource has been released
EDESC
trapAdapterSwap {1.3.6.1.4.1.2.3.1.2.1.5} 6 17 A 1 0 "Status Events"
SDESC
Fires whenever a address swap occurs.
EDESC
trapJoinNetwork {1.3.6.1.4.1.2.3.1.2.1.5} 6 66 A 1 0 "Status Events"
SDESC
Specified node has joined the network
EDESC
trapSwapAdapterComplete {1.3.6.1.4.1.2.3.1.2.1.5} 6 65 A 1 0 "Status Events"
SDESC
Specified node generated swap adapter complete event
EDESC
trapNodeState {1.3.6.1.4.1.2.3.1.2.1.5} 6 12 A 1 0 "Status Events"
SDESC
```

```
Fires whenever a node changes state.
EDESC
trapJoinNetworkComplete {1.3.6.1.4.1.2.3.1.2.1.5} 6 68 A 1 0 "Status Events"
SDESC
Specified node generated join network complete event
EDESC
trapConfigComplete {1.3.6.1.4.1.2.3.1.2.1.5} 6 80 A 1 0 "Status Events"
SDESC
Configuration procedure has completed for specified node
EDESC
trapJoinNodeComplete {1.3.6.1.4.1.2.3.1.2.1.5} 6 72 A 1 0 "Status Events"
SDESC
Specified node generated join node complete event
EDESC
trapResourceGroupChange {1.3.6.1.4.1.2.3.1.2.1.5} 6 93 A 1 0 "Status Events"
SDESC
rg_move event has occurred on the event node
EDESC
trapClusterNotify {1.3.6.1.4.1.2.3.1.2.1.5} 6 106 A 1 0 "Status Events"
SDESC
Cluster Notify event has occurred on event node
EDESC
trapSiteIsolationComplete {1.3.6.1.4.1.2.3.1.2.1.5} 6 105 A 1 0 "Status Events"
SDESC
Site isolation is complete on the event site
EDESC
trapSiteDownComplete {1.3.6.1.4.1.2.3.1.2.1.5} 6 99 A 1 0 "Status Events"
SDESC
Site failure complete on the event site
EDESC
trapJoinInterface {1.3.6.1.4.1.2.3.1.2.1.5} 6 92 A 1 0 "Status Events"
SDESC
Interface has joined on the event node
EDESC
trapFailNetwork {1.3.6.1.4.1.2.3.1.2.1.5} 6 67 A 1 0 "Status Events"
SDESC
Specified node generated fail network event
EDESC
trapNewPrimary {1.3.6.1.4.1.2.3.1.2.1.5} 6 15 A 1 0 "Status Events"
SDESC
Fires whenever the primary node changes.
EDESC
trapClusterUnstable {1.3.6.1.4.1.2.3.1.2.1.5} 6 77 A 1 0 "Status Events"
SDESC
Specified node generated cluster unstable event
EDESC
trapSiteMergeComplete {1.3.6.1.4.1.2.3.1.2.1.5} 6 103 A 1 0 "Status Events"
SDESC
Site merge is complete on the event site
EDESC
trapSiteUp {1.3.6.1.4.1.2.3.1.2.1.5} 6 100 A 1 0 "Status Events"
SDESC
Site is now up
EDESC
trapEventError {1.3.6.1.4.1.2.3.1.2.1.5} 6 83 A 1 0 "Status Events"
```

```
SDESC
Specified node generated an event error
EDESC
trapRGDepChange {1.3.6.1.4.1.2.3.1.2.1.5} 6 32 A 1 0 "Status Events"
SDESC
Fires when an resource group dependency is changed.
EDESC
trapServerRestart {1.3.6.1.4.1.2.3.1.2.1.5} 6 94 A 1 0 "Status Events"
SDESC
Server has been restarted on the event node
EDESC
trapFailNodeComplete {1.3.6.1.4.1.2.3.1.2.1.5} 6 73 A 1 0 "Status Events"
SDESC
Specified node generated fail node complete event
EDESC
trapDareTopologyStart {1.3.6.1.4.1.2.3.1.2.1.5} 6 85 A 1 0 "Status Events"
SDESC
Dynamic reconfiguration event for topology has started
EDESC
trapJoinNode {1.3.6.1.4.1.2.3.1.2.1.5} 6 70 A 1 0 "Status Events"
SDESC
Specified node generated join node event
EDESC
trapAddressState {1.3.6.1.4.1.2.3.1.2.1.5} 6 14 A 1 0 "Status Events"
SDESC
Fires whenever a address changes state.
EDESC
trapFailInterface {1.3.6.1.4.1.2.3.1.2.1.5} 6 91 A 1 0 "Status Events"
SDESC
Interface has failed on the event node
EDESC
trapEventNewPrimary {1.3.6.1.4.1.2.3.1.2.1.5} 6 76 A 1 0 "Status Events"
SDESC
Specified node has become the new primary
EDESC
trapSiteMerge {1.3.6.1.4.1.2.3.1.2.1.5} 6 102 A 1 0 "Status Events"
SDESC
Site has merged with the active site
EDESC
trapServerRestartComplete {1.3.6.1.4.1.2.3.1.2.1.5} 6 95 A 1 0 "Status Events"
SDESC
Server restart is complete on the event node
EDESC
trapDareResourceAcquire {1.3.6.1.4.1.2.3.1.2.1.5} 6 89 A 1 0 "Status Events"
SDESC
Dynamic reconfiguration event for resource has been acquired
EDESC
trapRGAdd {1.3.6.1.4.1.2.3.1.2.1.5} 6 20 A 1 0 "Status Events"
SDESC
Fires whenever a resource group is added.
EDESC
trapServerDownComplete {1.3.6.1.4.1.2.3.1.2.1.5} 6 97 A 1 0 "Status Events"
SDESC
Server has failed on the event node
EDESC
```

```
trapClusterState {1.3.6.1.4.1.2.3.1.2.1.5} 6 10 A 1 0 "Status Events"
SDESC
Fires whenever the cluster changes state.
EDESC
trapResourceStateChangeComplete {1.3.6.1.4.1.2.3.1.2.1.5} 6 108 A 1 0 "Status
Events"
SDESC
Resource State Change Complete event has occurred on event node
EDESC
trapRGDepAdd {1.3.6.1.4.1.2.3.1.2.1.5} 6 30 A 1 0 "Status Events"
SDESC
Fires when a new resource group dependency is added.
EDESC
trapFailNode {1.3.6.1.4.1.2.3.1.2.1.5} 6 71 A 1 0 "Status Events"
SDESC
Specified node generated fail join node event
EDESC
trapExternalResourceStateChange {1.3.6.1.4.1.2.3.1.2.1.5} 6 109 A 1 0 "Status
Events"
SDESC
External Resource State Change event has occurred on event node
EDESC
trapJoinStandby {1.3.6.1.4.1.2.3.1.2.1.5} 6 74 A 1 0 "Status Events"
SDESC
Specified node generated join standby event
EDESC
trapClusterUnstableTooLong {1.3.6.1.4.1.2.3.1.2.1.5} 6 82 A 1 0 "Status Events"
SDESC
Specified node has been unstable too long
EDESC
```

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- *IBM PowerHA SystemMirror 7.1.2 Enterprise Edition for AIX,* SG24-8106
- *IBM PowerHA SystemMirror Standard Edition 7.1.1 for AIX Update*, SG24-8030
- *Deploying PowerHA Solution with AIX HyperSwap*, REDP-4954

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- *RSCT Version 3.1.2.0 Administration Guide, SA22-7889*

Online resources

These websites are also relevant as further information sources:

- PowerHA SystemMirror Concepts

  http://public.boulder.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.powerha.concepts/hacmpconcepts_pdf.pdf
- PowerHA SystemMirror system management C-SPOC

  http://public.boulder.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.powerha.concepts/ha_concepts_install_config_manage.htm/
- HyperSwap for PowerHA SystemMirror in the IBM Knowledge Center

  http://publib.boulder.ibm.com/infocenter/aix/v6r1/index.jsp?topic=%2Fcom.ibm.aix.powerha.pprc%2Fha_hyperswap_main.htm
- IBM PowerHA SystemMirror HyperSwap with Metro Mirror

  https://www.ibm.com/developerworks/aix/library/au-aix-hyper-swap/#!

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

IBM

Redbooks

Guide to IBM PowerHA SystemMirror for AIX, Version 7.1.3

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

Guide to IBM PowerHA SystemMirror for AIX Version 7.1.3

Outlines the latest PowerHA enhancements

Describes clustering with unicast communications

Includes migration scenarios

This IBM Redbooks publication for IBM Power Systems with IBM PowerHA SystemMirror Standard and Enterprise Editions (hardware, software, practices, reference architectures, and tools) documents a well-defined deployment model within an IBM Power Systems environment. It guides you through a planned foundation for a dynamic infrastructure for your enterprise applications.

This information is for technical consultants, technical support staff, IT architects, and IT specialists who are responsible for providing high availability and support for the IBM PowerHA SystemMirror Standard and Enterprise Editions on IBM POWER systems.

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

SG24-8167-00 ISBN 0738439819