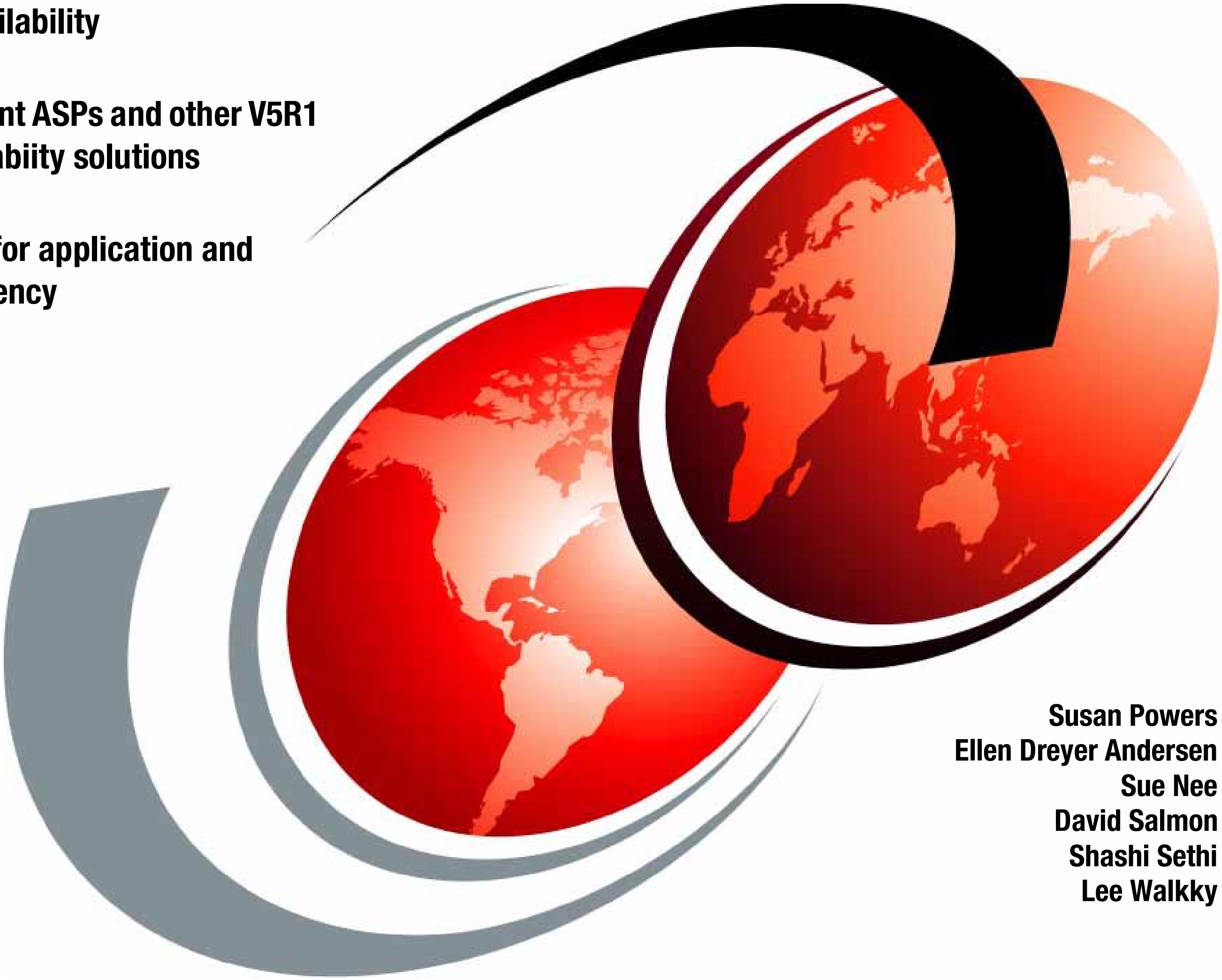
Clustering and IASPs for Higher Availability

on the IBM eserver iSeries Server

Moving the iSeries server beyond 99.9% availability

Independent ASPs and other V5R1 high availabiity solutions

Solutions for application and data resiliency



Susan Powers
Ellen Dreyer Andersen
Sue Nee
David Salmon
Shashi Sethi
Lee Walkky

Redbooks

ibm.com/redbooks

International Technical Support Organization

Clustering and IASPs for Higher Availability on the IBM @server iSeries Server

April 2002

SG24-5194-01

Take Note! Before using this information and the product it supports, be sure to read the general information in “Special notices” on page xv.

Second Edition (April 2002)

This edition applies to OS/400 Version 5, Release 1.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JLU Building 107-2
3605 Highway 52N
Rochester, Minnesota 55901-7829

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Chapter 14. Lakeview Technology MIMIX 249

Chapter 15. Vision Solutions Vision Suite 255

Figures

Tables

Special notices

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

IBM trademarks

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

e (logo)®
AIX®
AS/400®
AS/400e™
Balance®
ClusterProven™
DB2®
DB2 Universal Database™
DFS™
e (logo)®
Enterprise Storage Server™
IBM®
iSeries™
MQSeries®
Netfinity®
Operating System/400®
OS/400®
Parallel Sysplex®
PartnerWorld®
Perform™
pSeries™
Redbooks™
RS/6000®
S/390®
Service Director™
SP™
SP1®
SP2®
System/38™
TCS®
Tivoli®
xSeries™
Lotus®
Domino™
Redbooks (logo)™

Other company trademarks

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

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

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

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

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

Preface

With OS/400 V5R1, IBM @server iSeries servers support two methods of clustering. *Replication technology* is one method. The other method is *switchable disk technology*, which is referred to as *independent auxiliary storage pools (IASPs)* on the iSeries server.

This IBM Redbook presents an overview of cluster and switched disk technology available at OS/400 Version 5 Release 1. It explains the concepts and architecture surrounding iSeries clusters. It introduces you to the @server brand initiative – ClusterProven for iSeries – and explains how it applies to iSeries customers and independent software vendors. Application resiliency can be delivered by exploiting OS/400 cluster technology and cluster management services such as those provided by IBM High Availability Business Partners. It is available through IBM cluster middleware providers. Considerations for this application design are also introduced in this redbook.

This redbook is written for IBM customers, technical representatives, and Business Partners who plan business solutions and systems that are continuously available. You should use this book to gain a broad understanding of the cluster architecture available with OS/400 Version 5, Release 1, where clustering is viable. You should also use it to learn how to plan and implement clustering and independent ASPs.

Note: This redbook discusses high availability solutions beyond a single-system iSeries solution. Solutions for single-system availability are discussed in *The System Administrator's Companion to AS/400 Availability and Recovery*, SG24-2161.

The team that wrote this redbook

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

Susan Powers is a Senior I/T Specialist at the International Technical Support Organization, Rochester Center. Prior to joining the ITSO in 1997, she was an AS/400 Technical Advocate in the IBM Support Center with a variety of communications, performance, and work management assignments. Her IBM career began as a Program Support Representative and Systems Engineer in Des Moines, Iowa. She holds a degree in mathematics, with an emphasis in education, from St. Mary's College of Notre Dame.

Ellen Dreyer Andersen is a Certified IT Specialist in IBM Denmark. She has 22 years of experience working with the AS/400 and System/3x platforms. Since 1994, Ellen has specialized in AS/400e Systems Management with a special emphasis on performance, ADSTAR Distributed Storage Manager for AS/400, and high availability solutions.

Sue Nee currently works in the @server Executive Briefing Center for the iSeries in Rochester, Minnesota. She has had a variety of assignments in IBM, starting in the field working with IBM customers. She has worked in Competitive Marketing, focusing on the midrange server marketplace and has managed AS/400 data center operations for AS/400 Support Family Services: Performance Management/400, AS/400 Alert, and AS/400 Support Line. Her area of expertise in the Briefing Center is systems management and high availability computing.

David Salmon is a Senior I/T Availability Professional in IBM Global Services, Australia. He has 27 years experience in IBM. He began his career began as a hardware Customer Engineer on midrange systems, moving into the software area after 15 years to serve as a Branch Program Support Representative specializing in System/38 and AS/400 systems. Working as a Systems Consultant, David is experienced in Client Access/400 and in recovering customer systems. His current assignment is as an iSeries and AS/400 System Consultant in the Business Continuity and Recovery Services Centre in Sydney.

Shashi Sethy works for IBM Global Services out of Rochester, Minnesota, USA. He has over 20 years of consulting experience in the IT industry, the last ten years of which have been spent at IBM Rochester. He consults with many large corporations around the world in diverse areas of the iSeries server. As the iSeries server has evolved over the years, so have his areas of specialization. Currently, he is an active iSeries specialist in three areas – Clustering and IASP on the iSeries, iSeries performance including Application design and SQL optimization, and finally the use of MQSeries on the iSeries platform.

Lee Walkky works for the IBM @server Executive Briefing Center in Rochester, Minnesota, where he specializes in high availability and systems management. He has been there since June 2001. He began his career with IBM in OS/400 development for Common Data Management. In 1997, he left IBM and joined Lakeview Technology, where he worked in development and eventually became a High Availability Services Consultant. In 1999, Lee returned to IBM Global Services to implement the high availability environment for IBM Server Group Manufacturing. Lee has a degree in Computer Information Systems from Winona State University.

Thank you to the following people for their invaluable contributions to this project:

Michael Brinker
Bob Gintowt
Vicki Morey
Amartey Pearson
Ron Peterson
George Romano
IBM Rochester team

Mike Snyder, Lead Architect for iSeries Clustering
IBM Rochester Laboratory

Sue Baker
Eric Hess
Chuck Stupca
IBM Rochester Technical Services

R. L. (Bob) Blanscett
IBM UNIX Executive Briefing Center - Austin

Mike Warkentin, Senior Product Sales Specialist
Ilze Valdmanis, Senior Product Developer
DataMirror Corporation

Mary Lieser, Manager of Product Development
Glenn Van Benschoten, Product Marketing Director
Lakeview Technology

Dave Brown, Chief Scientist
Johannes Grobler, Chief Engineer
Vision Solutions, Inc.

Notice

This publication is intended to help system administrators understand the availability, backup, and recovery techniques for a high availability solution on iSeries servers. The information in this publication is not intended as the specification of any programming interfaces that are provided by high availability vendors, such as DataMirror Corporation, Lakeview Technology, and Vision Solutions, Inc. See the PUBLICATIONS section of the IBM Programming Announcement for OS/400 (5722-SS1) for more information about what publications are considered to be product documentation.

Comments welcome

Your comments are important to us!

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

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to the address on page ii.

Summary of changes

This section describes the technical changes made in this edition of the book compared to the first edition. This edition may also include minor corrections and editorial changes that are not identified.

Summary of Changes
for SG24-5194-01
for *Clustering and IASPs for Higher Availability on the iSeries Server*
as created or updated on April 16, 2002.

April 2002, Second Edition

The second edition of this redbook reflects the addition, deletion, and modification of new and changed information in iSeries clustering and availability technology. The original publication, *AS/400 Clusters: A Guide to Achieving Higher Availability*, SG24-5194-00, is based on OS/400 V4R4. The current edition is based on clustering and availability functions added in OS/400 V4R5 and V5R1.

New information

Several chapters (and appendices) are added in the second edition of this redbook to address new functions added since OS/400 V4R4. The new chapters include:

- Chapter 5, “Implementing and managing clusters with IBM solutions” on page 59

  This chapter discusses solutions provided by IBM to implement and manage clusters. These include the Operations Navigator Simple Cluster Management GUI, APIs, and commands available in QUSRTOOL library.

- Chapter 6, “Independent ASPs explained” on page 99, and Chapter 7, “Operations Navigator independent ASP GUI” on page 125

  These chapters describe the “what”, “why”, and “how” of the new iSeries switched disk functionality introduced with OS/40 V5R1.

- Chapter 9, “Making applications continuously available” on page 169

  This chapter describes an application view of continuously available. It also provides simple programming examples.

- Chapter 10, “Sample cluster environment and applications” on page 183

  This chapter illustrates a cluster configuration and describes program examples to support program resiliency.

- Appendix A, “Advanced clusters explained” on page 275

  This chapter is directed for the reader interested in more technical description of iSeries clustering architecture, or a deeper discussion than what is provided in other sections of the redbook.

- Appendix B, “Referenced lists” on page 283

  To improve the readability of the redbook, several lists and tables are moved to this appendix.

- Appendix C, “iSeries cluster resources” on page 291

  The education and services available from IBM is listed in this appendix.

Changed information

Much of the original redbook is revised with this current edition. In general, the redbook is changed from an executive overview of clustering functions, to an implementor’s view of the tasks involved to implement and support a higher availability clustering solution on the iSeries server.

Part 1

iSeries high availability fundamentals

The iSeries server is proven to be one of the most reliable servers in the marketplace. However, reliability is not availability. Unplanned downtime does happen – hardware failures can occur, operators do make errors, applications may fail, and power can give out. Unplanned downtime happens. Yet the largest share of downtime can be due to foreseeable, yet unavoidable tasks, such as system backups, software upgrades, and application and database management, maintenance or reorganization that require planned downtime to complete.

Clustering builds on the solid foundation of iSeries single-system high availability. An iSeries clustering solution is designed to address the problems caused by downtime and further enhance system availability.

This part introduces the reader to availability topics as a basis to understand iSeries solutions for higher availability.

1

Introduction

Clustering is designed as a high availability solution. Clustering involves a primary and backup system (or systems) linked together in a network. It supports switching from one server to another in the event of a system outage (planned and unplanned).

High availability computing can be achieved with an iSeries clustering implementation. This redbook focuses on clustering as a means to achieve high availability with an iSeries implementation. This chapter provides fundamental information about availability and the clustering architecture for the iSeries server. It serves as an introduction to the remainder of the redbook.

1.1 A brief overview about clustering

Clustering technology has been in the computing industry for years. It is used for high availability computing, horizontal scalability, and load balancing.

Most of the early cluster solutions are designed to solve a problem with limited scalability of the server. Coupling servers in a cluster provides a means to scale horizontally and grow capacity as business needs require.

With the scalability of servers available in the computing industry today, the need to cluster for horizontal growth is less. The new imperative is high availability.

High availability drives the need to cluster. Globalization of business, server consolidation, and Web based computing – these are the dynamics that bring server availability to the forefront of IT strategic planning. These factors place demand in an environment where the server must be available nearly 24 X 7 for business transaction processing or Web serving.

The users of a business' I/T infrastructure extend beyond its employees. A server outage potentially affects not only employees of the business, but also trading partners, customers, and perhaps the public at large. The visibility of a computing system outage is magnified. In this era of e-commerce, information availability is of the utmost importance for business survival.

With these dynamics as a backdrop, high availability computing is no longer viewed as applicable to only the largest, most sophisticated businesses. Cluster architecture provides support for customers who want to make their businesses continuously available.

The level of availability required in today's I/T environment can only be achieved by building redundancy into critical I/T resources, especially servers. Server redundancy is achieved through clustering. The primary goal of iSeries clustering is to achieve unprecedented system availability approaching 99.999% and beyond.

You may wonder, "Clustering is not new to the iSeries. The iSeries servers support dual systems with failover capability using data replication products from DataMirror, Lakeview Technology, and Vision Solutions. So what's new about iSeries and clustering that did not exist in the past?" What is new is that clustering functionality is built right into the system, something that did not exist in the past.

This new clustering support for the iSeries provides:

- A "heartbeat" monitor for systems in a cluster through a function in OS/400 called *Cluster Services*. If a failure occurs in one of the nodes, Cluster Services posts an error to an exit program to begin preparation for failover. This sophistication of the management of failover was not present prior to V4R4, when clustering was first introduced as part of OS/400.
- A GUI interface makes it easier to setup and manage cluster implementation.
- Application recoverability via an API interface that maintains a program's state when switching between nodes in a cluster.
- *Device resiliency*, which allows disk devices to be switched between nodes in a cluster and run applications on a different node.

This redbook provides an overview of this new clustering support for the iSeries introduced at V4R4 and that has been enhanced at V5R1. It also guides you through the steps required to set up replication-based clustering and clustering using independent auxiliary storage pools (IASPs). Cluster management and problem determination are also outlined.

1.2 Defining availability

An *outage* is a period when the information system is not available. During a scheduled outage, the system is planned to be made unavailable to users. Scheduled outages are used to run batch work, save the system, or apply program temporary fixes (PTFs). An unscheduled outage is usually caused by a failure of some type. Unscheduled outages typically cannot be predicted.

Clustering technologies and solutions are implemented to minimize the impact of outages. Availability solutions are designed for differing levels of system availability.

There is no industry standard to define precisely what high availability is. Different vendors and solution providers apply different meanings to terms like “continuous operations,” high availability,” and “continuous computing”. This redbook uses these definitions of the terms to discuss the availability levels offered by iSeries servers and clustering:

- **High availability**: Systems that reduce or eliminate unplanned outages.
- **Continuous operations**: Systems that reduce or eliminate planned outages.
- **Continuous availability**: Systems that reduce or eliminate both planned and unplanned outages.

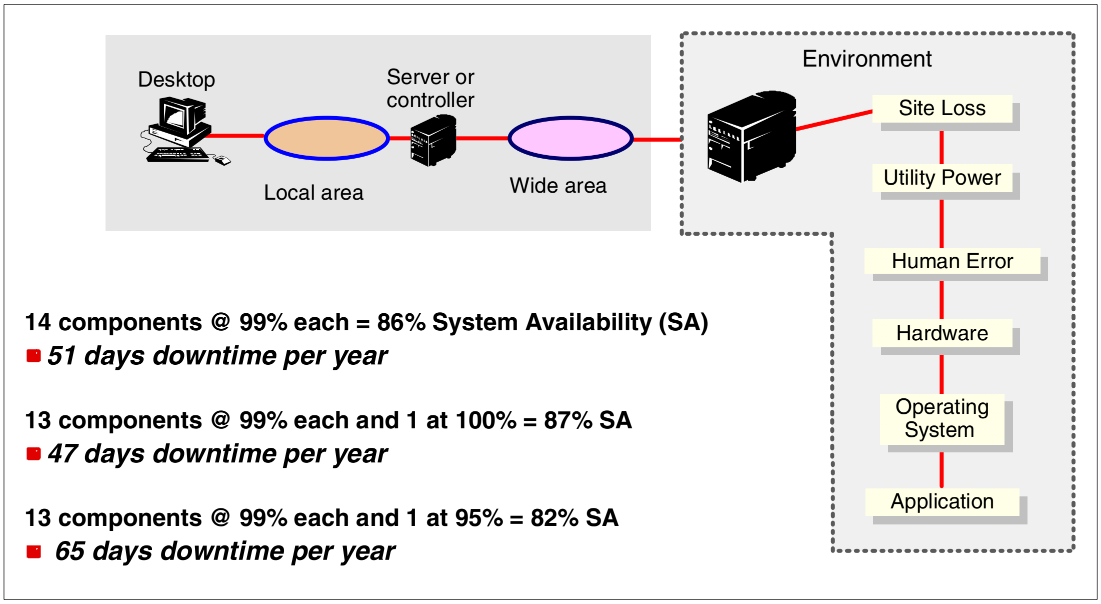
The iSeries server can achieve all three levels of availability.

Note: These availability definitions are consistent with the usage in the redbook *The System Administrator's Companion to AS/400 Availability and Recovery*, SG24-2161. This book is commonly referred to as “the companion redbook” and describes availability for a single-system iSeries (AS/4000e) implementation.

1.2.1 Factors impacting availability

Important: Server availability is just one component of a critical business process. For an effective high availability implementation, consider all of the elements involved, such as the wide area network (WAN) and local area network (LAN) resources, connections to other servers, and design for resiliency.

A component’s availability rating impacts the potential downtime of the solution. Availability decreases as more components are added to the I/T environment. The downtime numbers identified in Figure 1-1 illustrate this point.



14 components @ 99% each = 86% System Availability (SA)

- ***51 days downtime per year***

13 components @ 99% each and 1 at 100% = 87% SA

- ***47 days downtime per year***

13 components @ 99% each and 1 at 95% = 82% SA

- ***65 days downtime per year***

Figure 1-1 Elements impacting downtime

Table 1-1 correlates the availability ratings of 90 through 99.9999% to the equivalent duration of an outage measured in time.

Table 1-1 Availability percentage translated to potential business downtime

Availability percentage	**Total outage per year**
99.9999	32 seconds
99.999	5 minutes
99.99	53 minutes
99.9	8.8 hours
99	87 hours (3.6 days)
90	876 hours (36 days)

A 10% difference in availability rating makes a difference of seconds, minutes, hours, and indeed days.

To illustrate this reality, consider this scenario. For this example, we assume:

- A server supports a critical business process.
- The server is available 99% of the time.
- An application running on that server uses a network connection through a router.
- The availability of router is 95%.

To measure the availability for this critical business process, consider each component separately. The combined availability of the server and the router factors calculate to be 94%, which means a probability of 21 days of downtime per year. In contrast, if only the server's 99% availability is factored in, downtime is only 3.5 days per year.

If a probability of 21 days downtime per year cannot be tolerated by the business, the availability of the router must be improved. With 95% availability, the router is the weak link in the chain of factors contributing to availability.

2

Downtime

In larger companies, planning for downtime has been an inherent part of the I/T structure for many years. Recovery models would focus primarily on hot or cold sites, or redundant computer centers. With a large initial investment, many companies would elect to improve single system availability rather than buy into redundant systems. As hardware and software costs have declined, specifically processors, memory, disk, and communication bandwidth, entry into the recovery market is now more feasible for smaller companies.

The purpose of high availability is to avoid downtime. High availability implementation requires a business investment. Although iSeries servers hold some of the highest reliability and availability ratings of any server in the marketplace, iSeries customers still need to prepare for the possibility of downtime. (See Chapter 3, “Availability technology” on page 15, for information on single-system availability.)

This chapter discusses the implications of downtime from a business viewpoint, with an approach to analyze the impact of downtime, and how downtime can be affected by application design. The topics in this chapter apply to all platforms.

2.1 Scheduled and unscheduled downtime

Downtime – whether planned or unplanned – is, at the very least, an inconvenience to a business. With business' relying on their I/S infrastructure almost 24x7 today, managing downtime becomes a primary focus of I/T organizations. Clustering provides a means to reduce or eliminate downtime.

Planned downtime is more of a known entity. Although it is an inconvenience at times, business operations can adjust to accommodate the outage. A business must ask itself, "How long can we tolerate an unplanned outage without significantly impacting business?"

The answer determines what I/T infrastructure investment is required to manage unplanned downtime. If the answer is "for several days", then a high availability solution involving clustering for fast failover and recovery to a backup system probably is not required. But if the answer is "not for very long", then a further investment in high availability solutions is required.

The business impact of downtime is measured in terms of lost worker productivity, lost sales, lost manufacturing time, or customer satisfaction. These impacts are tangibly quantified as business costs.

To determine what investment in high availability is required, a business must weigh the costs of system downtime against the costs associated with implementing a high availability solution to reduce or avoid an outage. An example of this business impact analysis is presented in 2.2, "Example: Impact analysis of an outage" on page 9.

With current technology, most system outages are planned; that is, the system is down for a planned activity such as maintenance. Other system outages are unplanned, caused primarily by these factors as illustrated in Figure 2-1:

- Application failure
- Operator errors
- Hardware, power, operating system, disaster

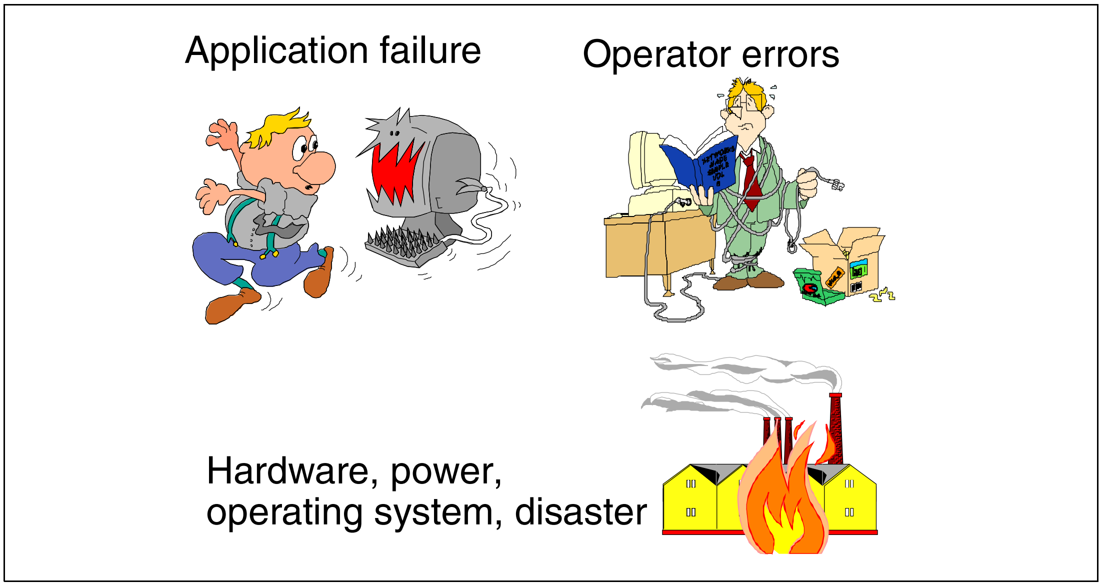
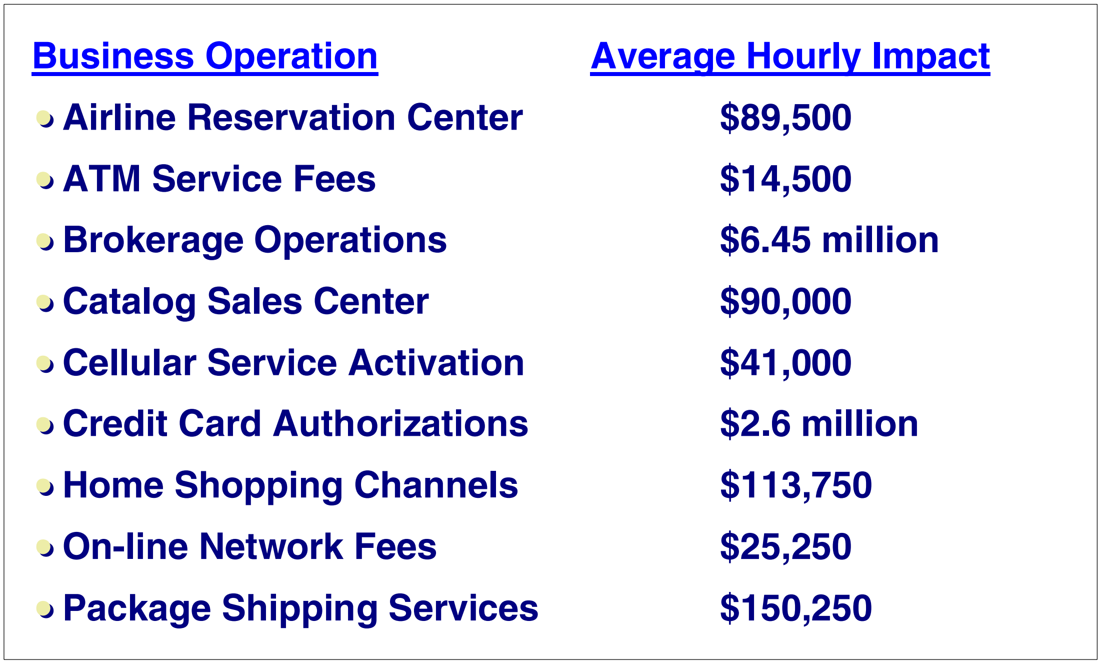


Figure 2-1 Causes of unplanned downtime

If a business builds a plan to address the scheduled outages, it addresses the unscheduled outages.

If a business builds a plan to address the impact of scheduled outages, it addresses some of the unscheduled outages.

Figure 2-2 lists the estimated hourly costs of downtime experienced by businesses in several industries.



Business Operation	Average Hourly Impact
Airline Reservation Center	$89,500
ATM Service Fees	$14,500
Brokerage Operations	$6.45 million
Catalog Sales Center	$90,000
Cellular Service Activation	$41,000
Credit Card Authorizations	$2.6 million
Home Shopping Channels	$113,750
On-line Network Fees	$25,250
Package Shipping Services	$150,250

Figure 2-2 Estimated costs of downtime

Downtime can be costly. High availability computing is a form of insurance. The higher the cost an outage is to a business, the easier it is to cost justify the insurance that a high availability solution provides.

2.2 Example: Impact analysis of an outage

This section presents a summary of an analysis of the impact to a business of the worst type of system outage – a site disaster. The business described is a worldwide manufacturing company that has implemented an ERP package across all its locations. The name of the company is not revealed. The numbers are real.

The analysis takes into account three levels of recovery implementation, the business impact to one business unit and the entire company, and the costs (in dollars) associated with implementing only the physical infrastructure. The costs associated with manual processes, employee inefficiencies, lost sales, any lost market value of the company, the restart of applications, and synchronization with a manual system are not included.

The levels of recovery implementation considered to develop the business impact model are:

- **Level 1**: No infrastructure is in place to initiate the restore of data from backup tapes. Agreements are in effect to have hardware shipped in a timely manner if a disaster event is declared.

  The business impact at Level 1, over a three-week period, represents a loss of approximately 2.5% of the gross revenue of the company.

The factors in level 1 take into account the amount of time to recover all server platforms, the operating systems and data. The time required to perform a system re-test and to resynchronize the application is considered for the ERP application only.

- **Level 2**: A minimum infrastructure is in place to allow a faster start to restore data from backup tapes. There is major transaction loss. Data is restored after the failure.

  The business impact at Level 2, over a ten day period, represents a loss of approximately 1.7% of the gross revenue of the company.

- **Level 3**: Continuous availability. There is no transaction loss and little impact to the business.

  The impact of 30 minutes or less of application outage is rated as minimal. The costs to achieve this minimized business risk are not cost prohibitive when compared to Level 2.

For a more complete determination of the cost of a business outage due to a site disaster, quantify the cost of these consequences:

- Problems with data integrity
- A loss of productivity due to inconsistent access to data and applications
- A loss of business as a result of lost sales
- The affects to the company image

Other potential negative impacts to the business due to an outage that are not included in this example cost analysis are:

- A consequential loss of external business
- A loss in market value

Table 2-1 identifies the recovery time, the impact of the duration of the recovery in terms of costs, and the cost of the business investment for the three levels of recovery implementation that were assessed.

Table 2-1 ERP disaster recovery options by level

Option	Description	Recovery time	Single business unit lost revenue *	Business impact lost revenue **	Disaster recovery implementation
Level 1	React at the time of disaster	3 weeks	Over 1 million	150 million	None
Level 2	Minimum infrastructure build today; data restored after disaster	10 days	$750,000	100 million	$775,000
Level 3	Continuous availability	30 minutes or less	Minimum	Minimum	Level 2 costs plus $150,000

* Source: Single business unit
** Source: Cumulative Financial Impacts and Exposures. These numbers represent the losses for all global business units.

While the cost to the business of moving to recovery Level 2 appears high, compare this amount to a potential loss of $100 million, which is less than one percent of the potential loss. It is a small price to pay for the business to survive. As such, a continuously available solution is a relatively small cost. Business managers should plan to reach Level 3, rather than stay on recovery Level 2.

The implementation time differs for Level 2 and Level 3. When a business is serious about availability, Level 2 is viewed as a tactical business solution. Level 2 recovery can provide protection until the more complex options at Level 3 are implemented.

Studies have shown that a loss of data and impact to business are:

- 43% of companies experiencing disasters never re-open. 29% close within two years (McGladrey and Pullen)
- One out of 500 data centers has a severe disaster each year (McGladrey and Pullen)
- A company that experiences a computer outage which lasts more than ten days never fully recovers financially.
- 50% of the businesses experiencing a computer loss are out of business within five years.

For more information, refer to *Disaster Recovery Planning: Managing Risks and Catastrophe in Information Systems* by Jon Toigo.

The advantages far outweigh the disadvantages and risk represented at each availability level. Table 2-2 illustrates the impact to the business for each level of availability.

Table 2-2 Business impact advantages by level

Level	Advantage	Cost to implement	Disadvantage	Cost to business	Risk protection*
1	None	None	Lost revenue	Can destroy company	None
2	Significant benefit Some downtime	Large investment	Substantial loss of revenue	Annual maintenance and cost of support	Low
3	High customer satisfaction Automated processes No downtime for system upgrades	Higher cost than Level 2	Highest cost	Not significantly higher than Level 2	Extremely high
* Insurance coverage					

While the initial expenditure for a highly available system is viewed as prohibitive, the resulting savings is even greater.

2.3 Application architecture and downtime

Modern applications are typically multitiered and involve edge (cache) servers, Web servers, application servers, and database servers, as illustrated in Figure 2-3. This component approach to application design introduces a higher probability of failure and subsequent potential for downtime.

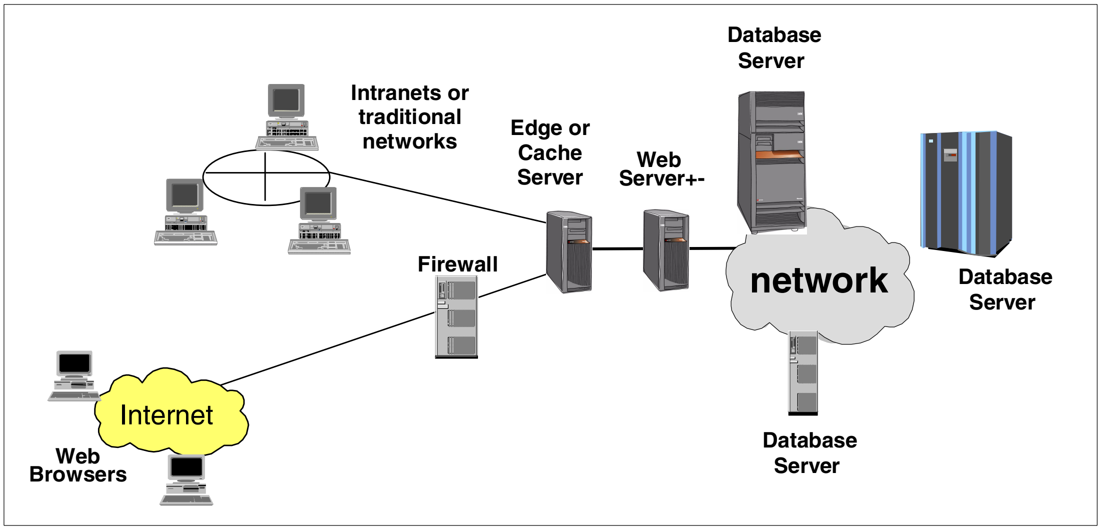


Figure 2-3 Typical multi-tier Web application

Today's application environment has evolved over time and is driven by the introduction of industry standards for databases, programming languages, communication protocols, messaging services, and other key technology components. These standards provide an application framework to build very flexible solutions with a lot of interconnections.

New standards continue to emerge. For example, XML provides for a whole new type of client to access server resources through wireless devices. With wireless access, users are not restricted to their office or home to make a connection to a server. They can carry their server access with them. This sort of pervasive computing adds pressure to I/T staff and solutions, to reduce or eliminate downtime altogether.

Application recoverability can be a large component to address unscheduled downtime. In the event of an unscheduled outage, if an application does not have good recoverability characteristics, transactions are lost or partially completed. This compromises the integrity of the database. The time it takes to sort out partially completed transactions and missing database records can considerably lengthen the overall recovery process.

In a high availability environment, an application must be designed with recoverability as part of the overall objective. Include commitment control in the design to allow a partially completed transaction to be rolled back.

Adding commitment control to an existing application is a task that many businesses are reluctant to do, in spite of the recoverability benefits that are to be gained. Often an application is developed over time and the original program designers are no longer available. It becomes a challenge to undertake any major changes because how long it takes, or the potential impact of the changes, is unknown.

A benefit of the clustering support introduced with OS/400 V4R4 for the iSeries server is that clustering provides a framework to provide better recoverability for applications. This framework supports several levels of application recoverability, starting simply with the use of an exit program to record which screen or step an application is executing when the system fails, and sending the application state information to the secondary system when the primary fails. This allows an application to resume at a known point when it starts up on the secondary system.

This level of application recoverability can be accomplished without making changes to the application itself. Therefore, iSeries clustering architecture allows an improvement to the application's recoverability without being intrusive to the application.

Important: Full transaction level recovery still needs to be addressed with commitment control.

Application design and recoverability are explained further in Chapter 9, "Making applications continuously available" on page 169.

3

Availability technology

iSeries servers earn some of the highest reliability and availability ratings of any server in the market place today. There are many availability features inherent within OS/400 and iSeries hardware. From its inception onward, the iSeries server is designed to run applications to support core business processes. The iSeries is built for business.

Because it is built to support business, features are designed to avoid unscheduled system downtime whenever possible and to quickly restore the system to an operational state should a failure occur.

The impact of an unscheduled outage of the iSeries server components can be minimized, because many operations can continue while error reporting occurs. The errors logged allow maintenance to be deferred and scheduled for maintenance at a less disruptive time.

System backups are the most frequent cause for scheduled downtime on a system.

For scheduled outages, the philosophy and design of the iSeries server are to provide tools, utilities, processes, and hardware resources to make a scheduled outage as brief as possible. Examples of this include support for fast tape technology (3590 and Linear Tape Output (LTO) devices), and leveraging OS/400's multi-threading capability by initiating multiple save jobs in parallel to more than one tape drive. These features reduce the time it takes to backup a system.

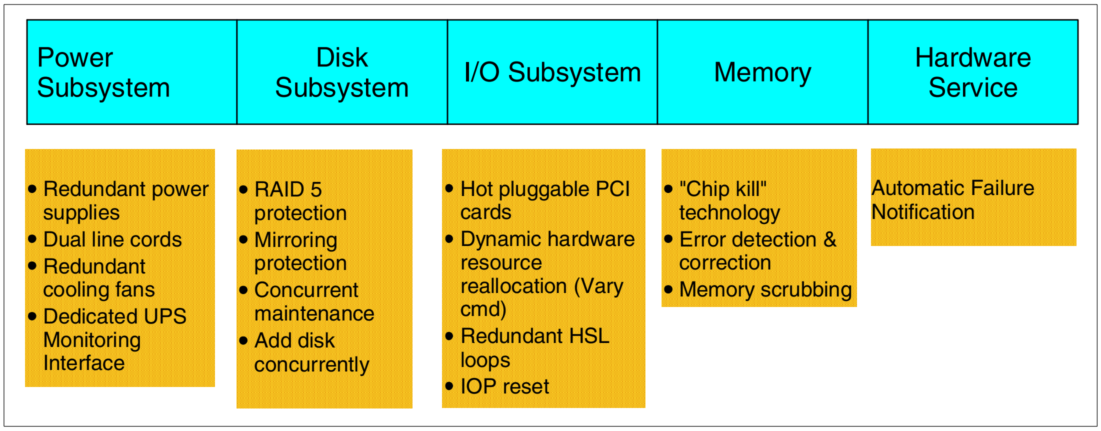
OS/400 itself maintains message logs, job logs, and the system history log to ensure they do not become too large and perhaps impact system performance or operations. OS/400 reclaims virtual addresses and temporary disk space used by system jobs and applications to reduce the requirement of a system IPL to reclaim and clean up resources.

The limit of what can be achieved in a single system environment is reached between a 99.9% and 99.99% availability. Achieving higher availability (99.999% and above) is only possible using a multiple system approach. With the iSeries server's implementation of clustering, availability tends to move closer to the 100% target.

This chapter describes the availability technology built into the iSeries server. It supplements the information found in the redbook *The System Administrator's Companion to AS/400 Availability and Recovery*, SG24-2161, which explains iSeries availability from a single system perspective.

3.1 iSeries server hardware availability features

Figure 3-1 identifies the availability features built into the iSeries server hardware. These features are described in more detail in this section.



Power Subsystem	Disk Subsystem	I/O Subsystem	Memory	Hardware Service
• Redundant power supplies • Dual line cords • Redundant cooling fans • Dedicated UPS Monitoring Interface	• RAID 5 protection • Mirroring protection • Concurrent maintenance • Add disk concurrently	• Hot pluggable PCI cards • Dynamic hardware resource reallocation (Vary cmd) • Redundant HSL loops • IOP reset	• "Chip kill" technology • Error detection & correction • Memory scrubbing	Automatic Failure Notification

Figure 3-1 iSeries server hardware availability features

- **Power subsystem**
  - *Redundant power supplies, cooling fans, dual line cords*

    Redundant power supplies and cooling fans are options available for iSeries servers. Some models of the system can be ordered with dual line cords.

    These features allow power to be supplied from more than one source, with one power source acting as a backup in the event of a disruption to the alternate power source.
  - *Dedicated UPS interface*

    The iSeries server provides a program interface to monitor and manage the switch to a Uninterruptible Power Supply (UPS) source in the event of a power outage. The system sends a message (that can be monitored for) when it detects power loss. A power handling program can monitor for power-related messages and manage the switchover to a UPS.
- **Disk subsystem**
  - *Device parity protection (RAID-5)*

    Device parity protection (RAID-5) is a hardware availability function that protects data from loss due to a disk unit failure or because of damage to a disk. The overall goal of device parity protection is to provide high availability and to protect data as inexpensively as possible.

    To protect data, the disk controller or input/output processor (IOP) calculates and saves a parity value for each bit of data. Conceptually, the disk controller or IOP computes the parity value from the data at the same location on each of the other disk units in the device parity set. When a disk failure occurs, the parity value and values of the bits in the corresponding locations on the other disks are used to reconstruct the data. The system continues to run while the data is reconstructed.

  - *Mirrored protection*

    Mirrored protection is an availability function that protects data from being lost due to failure or because of damage to a disk-related component. Data is protected because the system keeps two copies of data on two separate disk units. When a disk-related component fails, the system continues to operate without interruption. The mirrored copy of the data is used until the failed component is repaired.

    Mirroring can be extended to include mirroring the disk IOPs and the busses that the disk units are attached to so the disk subsystem can continue to function even if a disk IOP or a bus fails.

  - *Concurrent maintenance*

    The iSeries disk subsystem allows maintenance to be performed on a disk drive that is part of a mirrored pair or a RAID-5 set while the system remains operational. Disks can be added concurrently, meaning disk capacity can be increased without disruption to system operations. Because the system manages storage automatically, newly added drives are immediately available for use. There is no requirement to partition the drives or move data to them in order for the system to utilize the drives. The system manages all space as one virtual address. Other than configuring the disks as new hardware devices, special setup is not required to make a new disk operational.

- **I/O subsystem**

  - *Hot pluggable PCI cards*

    Hot plugging is made possible by the existence of a power control to individual cards slots. PCI IOPs or IOAs can be added, removed, or replaced while the system remains active.

  - *Dynamic hardware resource reallocation*

    Each hardware device on the iSeries server has a device description associated with it. The description contains the name of the specific hardware component that the hardware resource is associated with.

    If a hardware device fails and there is a backup device for it installed in the system, the device description can be modified to point to the backup device. It can then be substituted for the failing device.

  - *Redundant HSL loops*

    High Speed Link (HSL) is a new fibre bus structure introduced for iSeries servers. HSL is a 1 Gigabyte per second bus that includes a redundant path capability. If a system bus cable is broken or unplugged, I/O traffic is routed through an alternate path, therefore, avoiding a system outage.

  - *IOP reset*

    The iSeries I/O architecture uses intelligent I/O processors (IOPs) to control hardware adapters. Should a failure occur in one of these IOPs, it can be reset (or “re-booted”) with the system VARY command. This avoids the need to IPL the system to recover from an I /O error.

- **Memory**

  iSeries memory represents “Chip Kill” technology. If a segment of memory fails, the iSeries simply makes unavailable the range of addresses, including the defective address or addresses. A message is sent to the system operator and the hardware error logs are updated with data related to the failure.

  Therefore, the system can remain active should a part of main storage fail. Maintenance can be deferred, which allows the system to tolerate memory failures without bringing the system down.

The system also performs a background “scrub” of memory, to detect and correct single and double bit errors.

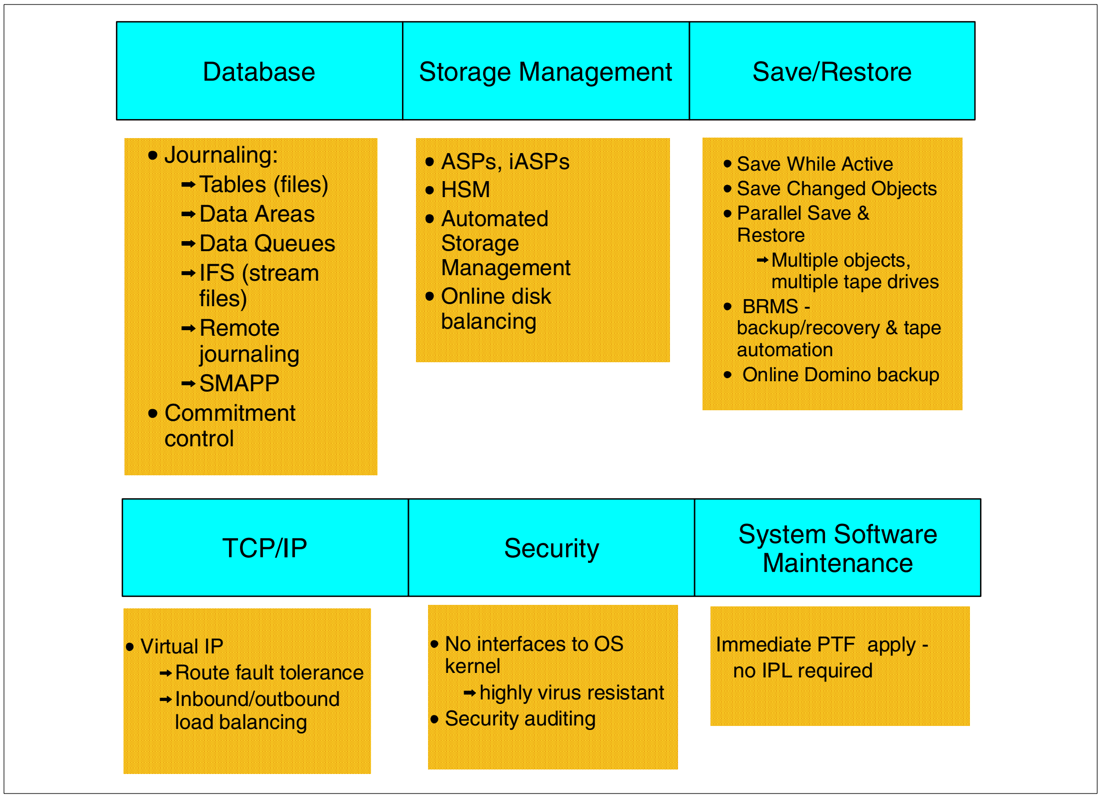
- **Hardware service**
  - *Hardware failure notification*

    With iSeries Service Director, the system “phones home” to a service machine when it detects key hardware component failures. A customer can optionally choose to have a repair engineer dispatched automatically when a hardware failure is logged.

    There are many cases recorded where a service engineer comes to a customer’s premises in response to a hardware problem detected by Service Director, and the customer is not even aware of the problem because the system was able to continue operations.

3.2 OS/400 and system software availability features

This section provides additional detail about the availability features built into the iSeries server software. The features that are addressed are listed in Figure 3-2.



Database

- Journaling:
  - ➡Tables (files)
  - ➡Data Areas
  - ➡Data Queues
  - ➡IFS (stream files)
  - ➡Remote journaling
  - ➡SMAPP
- Commitment control

Storage Management

- ASPs, iASPs
- HSM
- Automated Storage Management
- Online disk balancing

Save/Restore

- Save While Active
- Save Changed Objects
- Parallel Save & Restore
  - ➡Multiple objects, multiple tape drives
- BRMS - backup/recovery & tape automation
- Online Domino backup

TCP/IP

- Virtual IP
  - ➡Route fault tolerance
  - ➡Inbound/outbound load balancing

Security

- No interfaces to OS kernel
  - ➡highly virus resistant
- Security auditing

System Software Maintenance

Immediate PTF apply - no IPL required

Figure 3-2 iSeries software availability features

- **Database – DB2 Universal Database for iSeries**
  - *Journaling*

    iSeries journaling was initially introduced to record changes made to database files. In the event of a system outage, the journal is used to reconstruct the file based on changes recorded in the journal receiver.

iSeries journaling has evolved over time, as has the style of computing that the system supports. Journaling support is enhanced to include byte stream files (Integrated File System files), data areas, and data queues.

Remote journaling was introduced to the system at V4R2. With remote journaling, journal receiver entries are replicated to a backup or remote system.

Remote journaling can be setup to run in synchronous or asynchronous mode. When remote journaling is synchronous, a database update for the source system is not completed until the target system makes the journal entry in its receiver.

Remote journaling can be used in conjunction with database replication for high availability. You can find more information about remote journaling in the redbook *AS/400 Remote Journal Function for High Availability and Data Replication*, SG24-5189.

The system also provides a journal for security auditing purposes, as described under the bullet “Security” on page 21.

- *Commitment control*

  Some applications involve multi-step transactions to update the database. It is imperative that you complete *all* steps within the transaction before you commit the database update. The iSeries provides commitment control for this transaction environment. Commitment control is an application-level function that defines the transaction boundary. It is used in conjunction with database journaling.

  In the event of a system failure, commitment control uses journal entries to “roll back” an entire transaction. Therefore, a partial update to database files is avoided.

  An example of the need for commitment control is a financial application that moves funds between accounts. In order for the transaction to be considered complete, the debit and credit of the accounts involved must both be reflected in the database.

- **Storage management**

  - *Auxiliary storage pools (ASP)*

    iSeries single level storage treats all storage as one large virtual address space (this includes main store memory as well as disk). There is no concept of a disk volume or data set partition. However, the system provides the capability to separate this contiguous address space into smaller disk “pools” to make system backup and recovery faster and to provide Hierarchical Storage Management facilities. These pools are called auxiliary storage pools.

    Conceptually, each ASP on the system is a separate pool of disk units for single-level storage. The system spreads data across the disk units within an ASP. If a disk failure occurs, you need to recover only the data in the ASP that contains the failed unit.

    The user of ASPs can reduce system backup time. To do this, create ASP to include individual applications and data. A single ASP can then be backed up without impacting business operations while other applications that operate from different ASPs stay online.

    Introduced with V5R1, independent ASPs (IASPs) take the concept of ASPs further by making the ASP switchable between systems in a cluster. At V5R1, IASPs can contain only IFS objects. iSeries intends to support database objects in IASPs in the future.

  - *Hierarchical Storage Management*

    Hierarchical Storage Management (HSM) is a set of APIs supplied with OS/400.

    Beginning with V4R4, the iSeries Backup Recovery Media Services (BRMS) licensed program offers an HSM component. BRMS provides automated backup and recovery support for database and IFS files. It also provides automation for system recovery.

  HSM moves data across a hierarchy of storage, allowing data that is not heavily used to move to less costly storage. Retrieval of the data is transparent to users and programs. When the object is referenced, BRMS retrieves it for the user or program.

  HSM also helps reduce system back up time, as seldom used data is moved out of the system ASP and can be saved outside the backup window used for daily saves of critical business data.

  – *Automated storage management*

    The iSeries server has long been known for its low cost of ownership. A contributing factor is that the iSeries server does not need a database administrator (DBA) to track storage utilization and worry about moving data around to balance or enhance disk subsystem performance.

    Automated storage management is also an availability feature in that the database does not need to be made unavailable to perform this type of maintenance. OS/400 storage management automatically spreads data across all available disk arms to balance disk arm utilization. It also automatically allocates additional storage as files, libraries, and other objects grow. There is no need to take the database or a file offline to extend its size.

  – *Online disk balancing*

    If a large number of disk drives are added at once, run the Start ASP Balance (STRASPBAL) CL command to redistribute data across the disk arms and rebalance arm utilization. There is no need to partition data sets or to move data between volumes as required with other databases to balance performance.

- **Save and restore**

  OS/400 provides a very comprehensive set of save and restore capabilities. These capabilities include:

  – *Save-while-active*

    Save-while-active provides a means to save an object to tape while the system remains active. Any application using a file or library being saved while the system is active must temporarily stop processing before the save can occur. Save-while-active then establishes a checkpoint image of the object and begins the save to tape while the application resumes execution.

    An advantage to save-while-active is that the entire system does not need to be brought down for back up. We recommend that you end all subsystems to ensure any database updates are written from memory to disk before the save is initiated.

  – *Save changed objects*

    OS/400 keeps a description for every object that exists on the system. Within this description, there is a time stamp that records the last time the object is changed and when it is last backed up. OS/400 save commands use this time stamp to provide the ability to save only objects that have been changed since the last save to tape. This reduces the amount of data saved and the time required to perform a system backup.

  – *Parallel save and restore*

    OS/400 allows parallelism within the system save commands. A system with multiple tape drives can initiate several save commands in parallel. Use the **`include`** and **`omit`** parameters to direct saves for specific libraries to different tape drives. Use this same approach to restore system objects using the restore procedures.

  - *Backup Recovery and Media Services (BRMS)*

    BRMS provides an automated means to manage tape libraries and to set up system save policies. Save policies can be setup for daily, weekly, and other schedules to ensure critical enterprise data is saved to tape media. BRMS tracks which system objects are saved and the date of the save, and reports objects that are not saved in the operation. BRMS creates a “recovery report”, which lists the steps required to restore a system in the event of an outage where the system must be recovered from backup media. BRMS uses the parallel save and restore support provided in OS/400.

- **TCP/IP**

  iSeries servers support a full function TCP/IP communications stack. The support is built into TCP/IP to facilitate high availability computing in a network environment. A description of these functions follows.

  - *Virtual IP*

    iSeries support for virtual IP allows the system to assign an IP address without designating it to a physical hardware device. All IP traffic can be routed through this virtual address. Each virtual address can have more than one physical communications adapter and/or system behind it. This way, if a physical card adapter or system fails, traffic can be rerouted to maintain availability. A client can be transparently re-routed. There is no need to re-establish or reconfigure the link to the alternate system.

    Virtual IP can also be used for load balancing and to direct sessions across communications adapters in a system. This helps to evenly distribute traffic for workload management.

- **Security**

  - With the well-known instances today of viruses and server hacking, to have a secure server that is not vulnerable to attack is a key component of availability.

    OS/400 has no open interfaces to the system kernel, which means the iSeries is highly resistant to hacking and viruses. The iSeries provides security auditing and uses system journaling support to log security entries. System security auditing can log activities with user profiles, objects on the system, and jobs.

- **System Software Maintenance**

  - To achieve higher levels of availability when applying PTFs, the iSeries adopts a philosophy to apply PTFs immediately (if possible), and not require a system IPL for the PTF to take effect.

3.2.1 Clusters for high availability

When business demands a system to be operational 24 hours a day, 365 days a year (24 x 365), a continuous availability solution is required. A viable solution for iSeries customers involves Cluster Resource Services (CRS). Cluster Resource Services is part of the OS/400 operating system and runs on each system in the cluster. CRS provides failover and switchover capabilities for systems used as database servers or application servers.

When a system outage or a site loss occurs, the functions provided on a clustered server system can be switched over to one or more designated backup systems that contain a current copy (replica) of the critical resource. The failover can be automatic. Or the operator can control how and when the transfer takes place by initiating a manual switchover.

Figure 3-3 shows a basic cluster. There are four node systems, A though D. The nodes are connected through a network. Systems A, B, and C are local to each other, and System D is at a remote location.

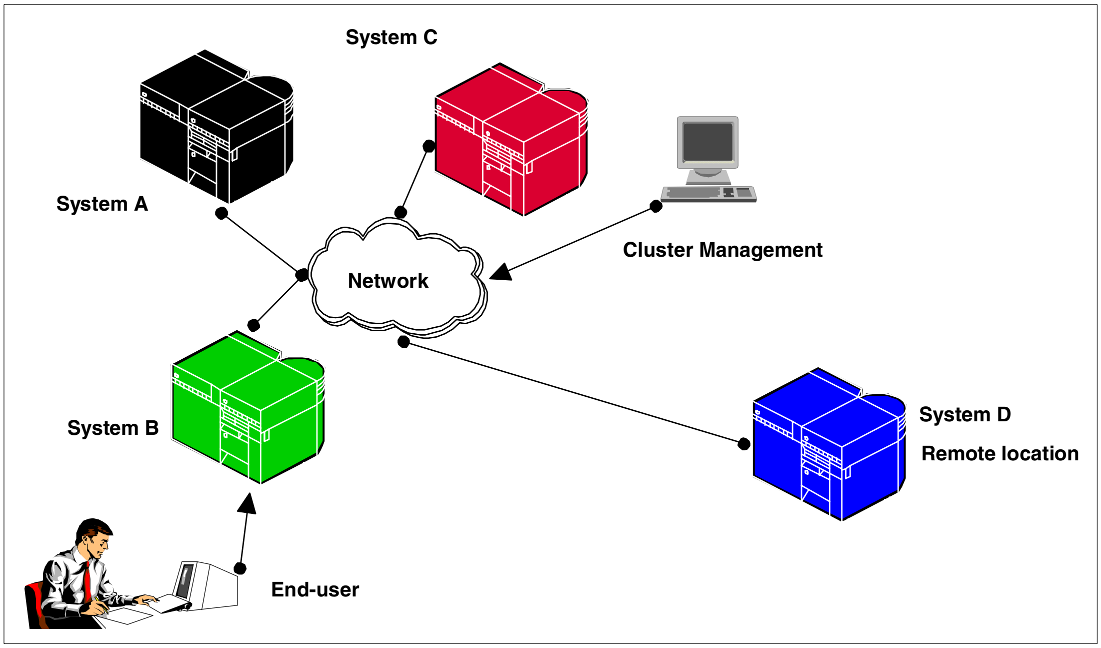


Figure 3-3 Basic cluster

The cluster management tool controls this cluster from anywhere in the network. End users work on servers in the cluster without knowing or caring from which server their application executes.

In the event of a failure, Cluster Resource Services provides a switchover. The switch causes minimal impact to the end user or applications running on a server system. Data requests are automatically rerouted to the new primary system. Multiple data replications of the same data are easily maintained.

Clusters contain more than two nodes. A system's resilient data (replicated data) can be grouped together to allow different systems to act as the backup system for each group's resilient data. Multiple backup systems are supported. If a system fails, Cluster Resource Services provides the means to automatically re-introduce or rejoin systems to the cluster and to restore their operational capabilities.

Hardware and software requirements for clusters

Any iSeries model that can run OS/400 Version 4 Release 4 or later is compatible for cluster implementation. OS/400 V4R4 or later must be installed, and Transmission Control Protocol/Internet Protocol (TCP/IP) must be configured on iSeries servers to implement clustering. Purchase a cluster management package from a cluster middleware business partner to provide the required replication functions and cluster management capabilities.

See Part 3, “Cluster middleware business partners” on page 227, for information on business partner solutions.

3.3 Cluster with replication technology

Replication technology is the database’s ability to make a synchronized copy of data and objects from one system to another. On the iSeries server, this is achieved with journaling and commitment control. Customers who understand the need for highly available systems implement these features on their systems and in their applications.

Journaling is the cornerstone of the high availability middleware provided by IBM HABPs. Journaling allows changes in the database to be recorded and stored. These changes can be transferred to the backup system by a communications method or using tape media.

Commitment control is implemented at an application level. It provides transaction boundary points. When a point is reached, the transaction is committed to the database. In the event of a failure, any incomplete transactions can be rolled back to the last complete transaction. Incomplete transactions still need to be re-keyed on the backup machine, but this scenario considerably adds to the recoverability of the application.

A few application providers have implemented commitment control in their applications. This position changes as the application providers deliver continuously available applications.

3.4 Cluster with switched disk

Disk drives can be switched from one system to another (see Figure 3-4). Local access to the data is only available from the owning system.

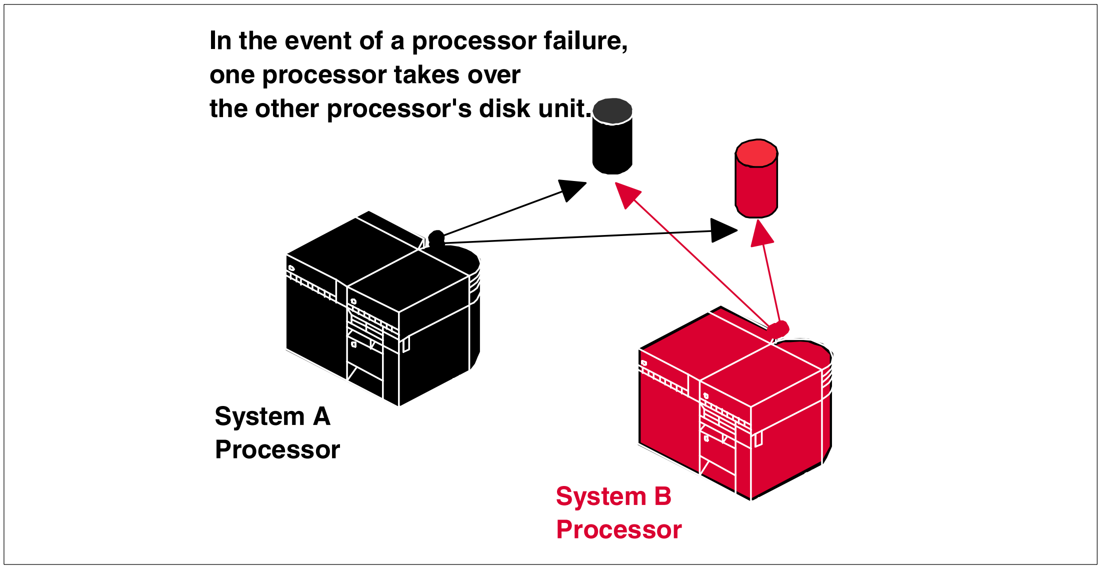

In the event of a processor failure,
one processor takes over
the other processor's disk unit.
System A
Processor
System B
Processor

Figure 3-4 Switched disk cluster

Some operating systems implement switched disk technology to improve the reliability from the single system model. Microsoft Cluster Services implements switched disk technology. With switched disk technology, if the processor fails, another processor complex takes over the disk and the associated database. This model is less expensive than dual systems because there is no duplication of disk units (DASD) and adapters. However, unless the hardware is unreliable, there is no significant difference between this and the single system model.

What happens to the business transactions in the switched disk model? In general, the following actions occur:

- The applications fail
- A switch is made
- The applications are restarted

If a switched system has on-line transaction processing (OLTP) applications and a failure occurs, there can be transactions that are only partially complete. To maintain database integrity, roll back the incomplete database transactions and restart the application. Re-key the incomplete transactions after the roll back completes.

For non-OLTP based applications, whether standalone or server-based, there can be less of a problem, depending on when the user last saved their work. For example, there is less of an impact to the database for a typical word processor or spreadsheet user, because their transactions typically only affect their open files. Disruptions to OLTP applications are typically more pervasive and require additional capabilities in database functionality.

The iSeries server implemented switched disk with OS/400 V5R1 technology. Refer to Chapter 4, “iSeries clusters explained” on page 31, for a further discussion.

3.5 Shared disk

In a shared disk setup, disk drives are attached to multiple systems simultaneously. Local access is available from all systems sharing the disk. Figure 3-5 illustrates a shared disk cluster.

The first design of shared disk technology allows every server to access every disk. This requires expensive cabling and switches, plus specialized operating system functions and specialized applications.

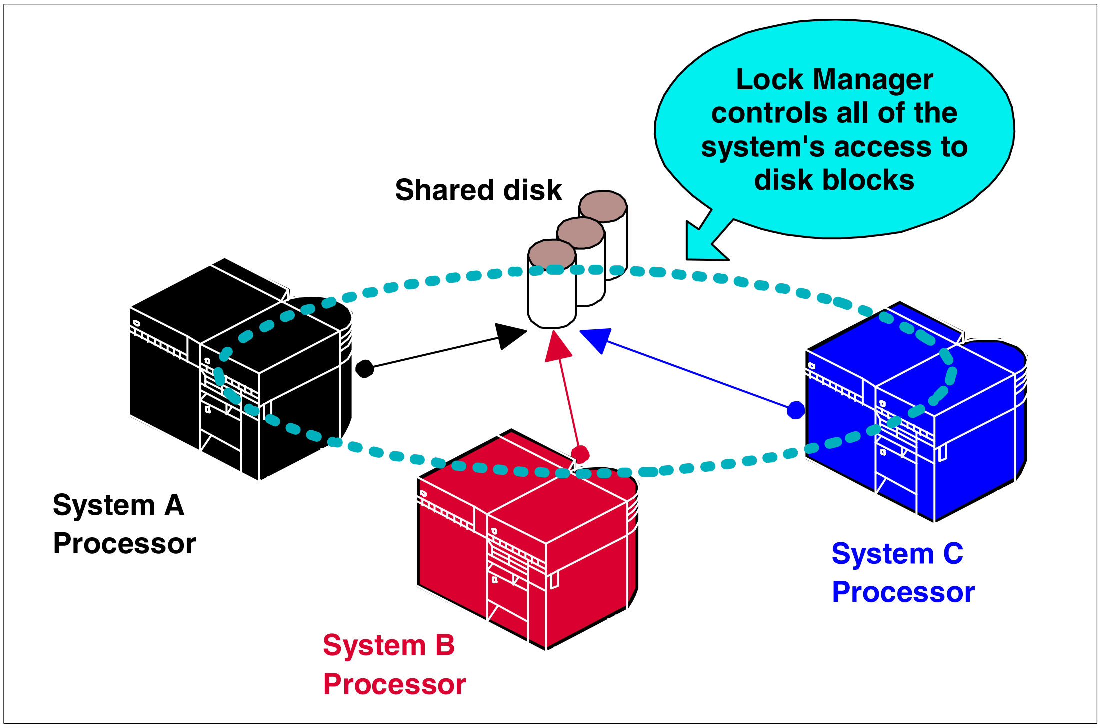


Figure 3-5 Shared disk

With today’s standards, such as small computer systems interface (SCSI), the need for expensive cabling and switches is eliminated. However, shared disk still requires specially modified applications.

In Figure 3-5, Systems A, B, and C are writing to and reading from the same disk. To manage this, the three systems have a form of DASD block management code. This code controls who has current access to a block of storage.

In this example, System A currently has a lock on block 123. Then System B requests block 123 on the shared DASD. The lock manager asks System A to give up block 123. When System A gives up the block, the lock manager changes the ownership of block 123 to System B. System B now has control of the block and can write all over it. At any time, Systems C or A can request the block back or can compete for other blocks. The lock manager can reside on any or all of the three systems.

IBM S/390 Parallel Sysplex successfully uses shared disk technology. This function has developed over time, with a significant investment in the system and applications to manage this function.

The iSeries server does not implement true shared disk functions. Single level storage and symmetric multi-processing (SMP) have some analogies to shared disk, where multiple applications run on multiple processors and in one storage pool. The user does not have to be concerned about where the data resides. The system takes care of managing the data, spreading the data across all the disks. OS/400 also takes care of object lock and task management.

There are more examples of the underlying iSeries functions that provide such high single system availability, largely taken for granted for many years.

3.6 Separate server

In a separate server cluster or dual system environment, data and objects are replicated from one system to another (as illustrated in Figure 3-6).

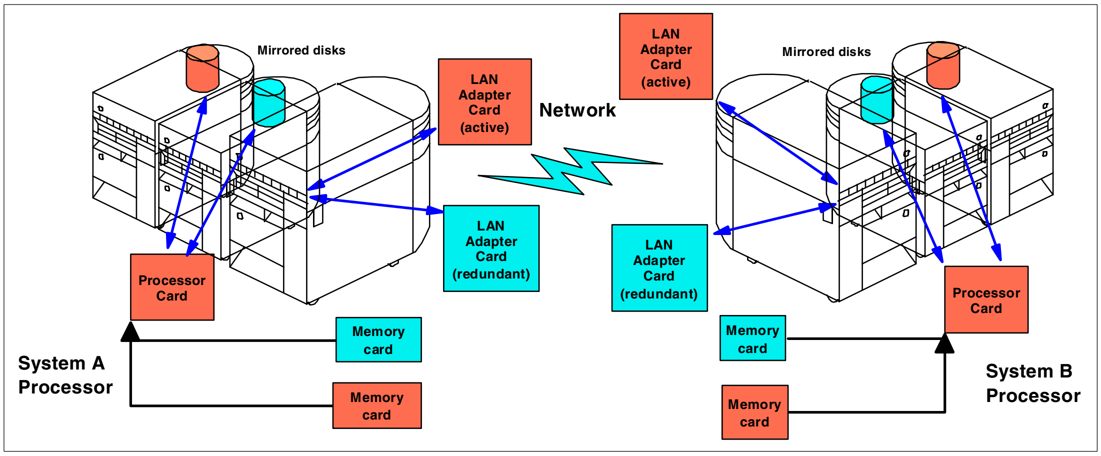


Figure 3-6 Separate server cluster

The primary resources in the iSeries server are its processors, memory (main storage), I/O buses, and IOPs. Each logical partition represents a division of these resources in the iSeries server. Each partition is logical (as opposed to physical), because the division of resources is virtual. The sharing of resources offered in a cluster that is made-up of logical partitions is illustrated in Figure 3-7.

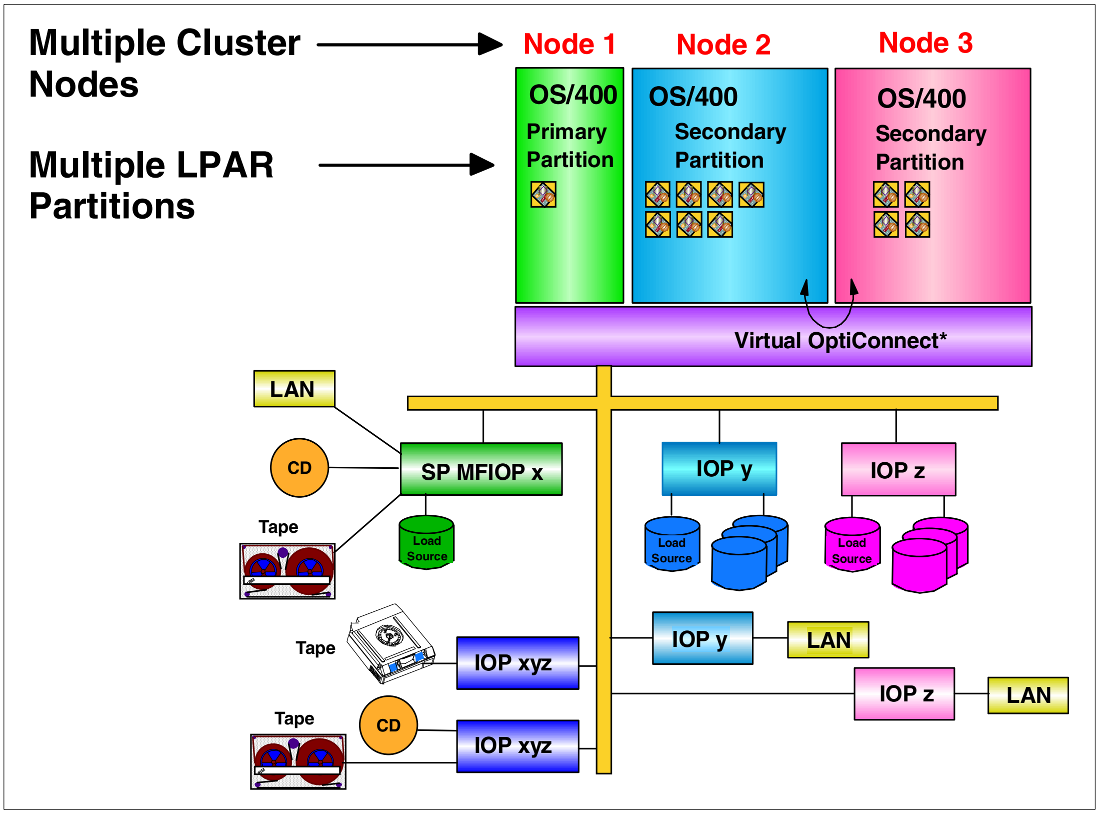


Figure 3-7 Cluster created by logical partitioning

OS/400 is licensed once for the entire system by its normal processor group, regardless of the number of partitions defined. License management across partitions is not supported in OS/400 V4R4. Install OS/400 on each partition. Releases prior to V4R4 are not supported in a logical partition.

Each logical partition operates as an independent logical system. Each partition shares a few physical system attributes such as the system serial number, system model, and processor feature code. All other system attributes can vary among partitions. For example, each partition has dedicated processors, main storage, and I/O device hardware.

An LPAR solution does not offer a true failover capability for all partitions. If the primary partition fails, all other partitions also fail. If there are multiple secondary partitions backing each other up, a failover can occur between partitions.

The secondary partitions are nodes and are a cluster solution. They are not a separate server implementation. An LPAR solution cannot provide the same level of availability as a two or more node cluster solution.

3.7 iSeries cluster middleware

Cluster middleware is the name given to the group of applications that provide the replication and management of application data between iSeries servers and that provide cluster management utilities.

The following cluster middleware providers offer data resiliency tools. Beginning with OS/400 V4R4, they are developing application resiliency offerings.

- DataMirror
- LakeView Technology
- Vision Solutions

Refer to the Cluster Management GUI described in Part 3, “Cluster middleware business partners” on page 227, to learn more about the cluster middleware solutions that are available for the iSeries server.

Part 2

iSeries clusters and higher availability

During the 1990s, the focus of information technology was server high availability and data resiliency. *Data resiliency* is when applications handle a copy of the data together with information about the currency of the data. While mirroring and RAID-5 increase the availability of the data source, data replication tools serve to enable a data resilient solution.

However, solutions that focus only on data availability cannot be available every hour of the day. The focus must include the application and the data together. That is why clustering technology was introduced in V4R4. Switching between systems in a cluster requires application resiliency and transaction signalling, as well as data availability.

Part 2 describes the components involved in the comprehensive solution for higher availability called *iSeries clustering*.

4

iSeries clusters explained

To explore the implementation of iSeries clustering, it is important to first understand iSeries clustering technology and capabilities. This chapter provides information that is fundamental to understanding what clustering is. It outlines the available methods to implement clustering and the reasons to invest in clustering.

For those of you who are looking for a deeper understanding of clustering, the underlying technology is described in Appendix A, “Advanced clusters explained” on page 275.

4.1 Cluster basics and definitions

A *cluster* can be defined as a configuration or a group of independent servers that appear on a network as a single machine. Stated another way, a cluster is a collection of complete systems that work together to provide a single, unified computing resource.

The cluster is managed as a single system or operating entity. It is designed specifically to tolerate component failures and to support the addition or subtraction of components in a way that is transparent to users.

The major benefits that clustering offers a business are:

- Continuous or high availability of systems, data, and applications
- Simplified administration of servers by allowing a customer to manage a group of systems as a single system or single database
- Increased scalability and flexibility by allowing a customer to seamlessly add new components as business growth develops

Attributes normally associated with the concept of clustering include:

- High availability and continuous availability
- Simplified single system management
- Scalability and flexibility
- High-speed interconnect communication
- Shared resources
- Workload balancing
- Single system image

> **Important:** It is important to note that there are several implementations or interpretations of what a cluster is. Different computer manufacturers have different cluster solutions. Most of these cluster solutions were designed to solve a limited horizontal growth in distributed systems. The design concept for iSeries clustering is that a number of systems closely coupled together can provide the capacity required for a growing business.

Sometimes horizontal support is referred to as *load balancing*. When a client job addresses a server in a cluster to get some work done, it is automatically directed to the server with less workload running at that time.

Some application software packages running on the iSeries server can also accomplish load balancing. An example is SAP.

4.2 iSeries clustering

OS/400 technologies provide a firm foundation for iSeries architecture. Clustering is the latest OS/400 improvement in iSeries technology for high availability.

The clustering technology introduced with OS/400 V4R4 builds on legacy AS/400 availability support for single-systems, such as journaling, commitment control, mirroring, and OptiConnect. iSeries high availability is offered with clustering support, for example:

- Peer relationships between cluster nodes help ensure no cluster-wide outage.
- Heartbeat monitoring and efficient cluster communications provide low overhead internode processing and early detection of potential node outages.

- The distributed activity groups are used to synchronize activities and objects across cluster nodes.
- Cluster engine services provide reliable, ordered messaging, and group membership services.
- The job structure of Cluster Resource Services (CRS), interjob communications, and internode communications provide a single, consistent view of cluster nodes, and cluster resource status.
- Through cluster partition handling, the system determines the difference between many failure and partition conditions without user intervention.

The explanation of these concepts comprises the content of this redbook.

OS/400 V5R1 provides key features to build the foundation for clustering support, as highlighted in 4.4, “OS/400 V5R1 clustering enhancements” on page 35.

OS/400 clustering delivers a standard for transferring applications, and their associated data, programs, and users, from one iSeries to another. An iSeries clustering solution offers continuous availability to meet the operational business demands 24 hours a day, 365 days a year (24 x 365). The foundation for this solution, called *OS/400 Cluster Resource Services*, is part of the OS/400 operating system. CRS provides failover and switchover capabilities for iSeries servers that are used as database servers or application servers.

If a system outage or a site loss occurs, the functions that are provided on a clustered server system can be switched over to one or more designated backup (idle standby) systems that contain a current copy (replica) of the critical resource. If a system failure should happen, the failover can be automatic. Or an operator can control how and when the transfer takes place by manually initiating a switchover.

The iSeries cluster uses a separate server, as well as a shared-nothing model. That is, cluster resources are not physically shared between multiple systems, or critical resources can be replicated between nodes. The resource is accessible from other nodes by shipping function requests to the system that is currently hosting the resource of interest. At any given moment, each resource is owned, or hosted, by a single system.

See Chapter 3, “Availability technology” on page 15, for a discussion of single-system availability options.

4.3 Why you want clustering

Small outages, tolerated just a few years ago, can now mean a significant loss of revenue and of future opportunities for a business. The most important aspect of clustering is *high availability*, that is, the ability to provide businesses with resilient processes. A well-managed iSeries cluster can provide the highest levels of availability of any individual server in the industry.

Clusters are a very effective solution for continuous availability requirements on an iSeries server, providing fast recovery for the widest range of outages possible, with minimal cost and overhead.

The concept of high availability in the sense of *disaster recovery* is an important consideration. However, disasters are not the only reason why high availability is so important. Disasters or unplanned outages account for only 20% of all outages. The majority of outages consists of planned ones, such as a shutdown to perform an upgrade or complete a total system backup. A relatively straightforward action, like the backup of databases and other objects, actually accounts for 50% of all planned outages.

Some of you may think that a backup of the server is not an outage. But iSeries users are not interested in such technicalities. If access to their data on the system is not possible, the user is most concerned about when the system is available again so that work can continue.

Actually, in many environments, any downtime creates a problem. According to one IBM study, the iSeries server averages 61 months between hardware failures. However, even this stellar record can be cause for availability concerns. Stated another way, 61 months between hardware failures means that nearly 67 percent of all iSeries servers can expect some type of hardware failure within the first five years. Such industries as health care and banking invest in redundant hardware and high-availability software solutions that minimize downtime by replicating the production environment on a backup system (an idle standby system). High availability of the system is also a big concern for e-business and Web-based solutions.

Regardless of what causes an outage and whether it is a planned or an unplanned one, the users – or in some cases, the customers – only see the system as unavailable, with all of the consequences to the business that this can have.

4.3.1 Hidden benefits of iSeries clusters

Clusters can be an efficient solution to availability issues, and in some situations, for server capacity and performance issues. Depending upon the availability requirements of a business, the backup iSeries server in a cluster can serve in an idle or an active mode:

- **Idle backup iSeries server**

  As an idle backup, the only active workload on the backup iSeries server is the cluster middleware product. The idle backup server provides data replication and cluster management.

  In the event of an outage, the backup server stands ready for a switchover, to assume primary iSeries production processing responsibilities. Backup mode typically provides for the fastest recovery in the event of an outage on the primary iSeries server, since no time is required to manage or end the workload on the backup iSeries server.

- **Active backup iSeries server**

  As an active backup, other work takes place on the backup iSeries server in addition to the cluster middleware. The active backup is productive throughout the day.

  By using replicated databases on the backup iSeries, “read only” types of workloads can be relocated to the backup iSeries to use the backup server’s available CPU and system resources.

  It is important that only “read only” work is performed against the replicated databases to avoid interference with the data replication processes. Functions that impact the availability, capacity, and performance of the primary iSeries server are candidates to be moved to the backup iSeries server.

For example, the active backup iSeries server can serve to:

– *Provide query, reporting, and inquiry capabilities at any time of the day*

  Ad hoc queries, reporting, Web-enabled customer inquiry, and Business Intelligence can have an impact on the primary (production) iSeries server performance. You should move these types of jobs to the backup iSeries server to use its processing power and replicated databases, without impacting the primary iSeries server.

– *Perform system maintenance*

  When the iSeries server is in a backup server role, system maintenance can be done without impacting business production. Examples of system maintenance include:

  - *Nightly saves*

    Even with fast, multiple tape drives and OS/400 save options, such as save-while-active, backup windows that require quiescing production work can become a problem.

    Use the data replicated to the backup iSeries so that the backup activity itself can be relocated to the backup server. This reduces the planned downtime on the primary iSeries server.

  - *Perform PTF maintenance*

    The backup iSeries server is available to assume production work, while the primary iSeries has regular PTFs applied. Return production work to the primary iSeries server once PTF maintenance is complete. This allows a minimal interruption to business.

  - *Perform release updates or system upgrades*

    The backup iSeries server can assume production work while the primary iSeries has its software (such as OS/400 and other licensed programs) upgraded to a new release, or when a hardware upgrade is performed.

    The data replication method of clustering supports a wide variety of versions and releases between systems, which can virtually eliminate the possibility that both systems must be upgraded at the same time.

  - *Eliminate development and testing from the production iSeries server*

    Periodically, an outage can occur due to application development and testing on the primary (production) iSeries server. Whether it is a planned or unplanned outage (such as accidental alteration or deletion of production data), it can be an outage that the business cannot tolerate.

    Assign application development and testing to the backup iSeries, where an outage does not immediately impact availability.

The ability to use a backup iSeries server in a cluster for the capabilities mentioned in this section depends on the availability requirements, configurations, spare capacity, and performance of the backup iSeries server, and the training of operations staff on the clustering solution. You should work closely with a cluster middleware consultant or an IBM Integrated Technology Services (ITS) consultant during the design phase of the cluster to ensure that failover and switchover capabilities meet the objectives of the business.

4.4 OS/400 V5R1 clustering enhancements

Clustering is an iSeries availability solution offered with OS/400 V4R4 and later. The functions and support added with V5R1 are listed in this section.

- Switchable independent auxiliary storage pools

  Independent auxiliary storage pools (IASP), also known as “switchable disk”, are described fully in Chapter 6, “Independent ASPs explained” on page 99.

- Resilient cluster device

  A resilient cluster device is represented by a system object. It is a hardware resource that can be switched between systems in the event of a planned system outage or a unplanned system failure.

  The resilient cluster device available with OS/400 V5R1 is the IASP.

- Device domain

  A device domain is a subset of cluster nodes across which a set of resilient devices can be “shared”. A device domain prevents conflicts that could cause resilient device switching to fail. The Resilient Cluster Device IASP can be active on one system at a time.

- Simple Cluster Management GUI

  A Simple Cluster Management GUI interface is accessible through Operations Navigator. It is offered as Option 41 of OS/400.

  Use the IBM Simple Cluster Management GUI to create and manage a two-node, switched disk cluster. The GUI allows the user to create and manage a cluster that uses switchable IASPs to ensure data availability. Simple Cluster Management features a wizard that steps the user through the creation of a simple, two-node cluster.

  Additional cluster management can be accomplished using the Simple Cluster Management GUI to perform tasks such as to:

  - Add a node to an existing one-node cluster
  - Add a switchable hardware group to a cluster
  - Add a switchable software product to a cluster
  - Change the cluster description
  - Change the exit program name
  - Change the takeover IP address for a switchable software product
  - Delete a cluster
  - Start a cluster
  - Stop a clustering
  - Switch cluster resources from the primary node to the backup node
  - View messages relative to cluster activity

Cluster performance tuning

APIs are available for basic tuning of cluster operations, such as allowing the user to set the tuning parameters to a predefined set of values identified for high, low, and normal time-out and messaging intervals.

Distribute Information

The Distribute Information (QcstDistributeInformation) API provides a mechanism to send information from one node in the CRG recovery domain to other nodes in the recovery domain. This can be a useful mechanism to communicate application activity or to send small amounts of information related to the application to affected nodes.

Cluster versioning

A cluster version represents the level of function available on the cluster. Versioning is a technique that allows the cluster to contain systems at multiple release levels and fully interoperate by determining the communications protocol level to use.

See 5.3.3, “Cluster versions” on page 96, and A.2.6, “Cluster versions” on page 281, for a further discussion.

Cluster partition improvements

Improved handling and recovery for cluster partitions is offered with OS/400 V5R1. In addition to better detection of some failover conditions, Cluster Resource Services provides an easier way to change the status of partition nodes to failed.

See A.2.5, “Cluster partition state” on page 279, for a further discussion.

Example commands and exit program

A set of example commands are provided in QUSRTOOL that can be used to create and manage a cluster in some environments. An example application CRG exit program is also included in the QUSRTOOL library. The sample source code can be used as the basis for writing an exit program.

> **Note:** Member TCSTINFO in the QUSRTOOL/QATTINFO file has more information on the cluster management commands. The TCSTAPPEXT member in the QATTSYSC file has an example exit program written in ILE C.

The tools in QUSRTOOL are meant for customers who want to create switchable applications and some advanced cluster functions. To create a simple cluster with switchable disks, the Simple Cluster Management GUI interface is sufficient.

See 5.3, “Using QUSRTOOL CL commands and OS/400 APIs to implement an iSeries cluster” on page 87, for a further discussion.

4.5 Prerequisites for iSeries clustering

The base functions for clustering are provided in V4R4 hardware. Clustering with switched disk support requires OS/400 V5R1 hardware. Therefore, the prerequisites for an iSeries clustering solution depend on IASPs and the type of IASPs that are to be implemented.

Specifically, the prerequisites for clustering are:

- **Hardware**
  - Two or more V4R4 systems or a logically partitioned system with a minimum of two partitions. For a logically partitioned system, each partition participating in the cluster must be at V5R1.
  - To support a switchable IASP in a standalone disk tower, V5R1 HSL adapter cards are required. Examples of adapters that support clustering include:
    - #7002 HSL Enabler
    - #2739/#9749 Optical Bus Adapter
    - #9691 Bus Adapter
- **Software**
  - TCP/IP Connectivity Utilities (5722-TC1)
  - Client Access Express (5722-XE1)
  - OS/400 HA Switchable Resources (Option 41 of 5722-SS1) to support the Operations Navigator GUI or switched disks

Note: HSL OptiConnect (Option 23 of 5722-SS1) is not required for clustering or IASPs. It is required for system-to-system communication, for example, Distributed Data Management (DDM).

These prerequisites are identified in a Help panel of Operations Navigator clustering, as shown in Figure 5-6 on page 68.

4.6 Cluster Resource Services

Cluster Resource Services is a component of OS/400. The functions provided by CRS are:

- Tools to create and manage clusters, the ability to detect a failure within a cluster, and switchover and failover mechanisms to move work between cluster nodes for planned or unplanned outages.
- A common method for setting up object replication for nodes within a cluster. This includes the data objects and program objects necessary to run applications that are cluster enabled.
- Mechanisms to automatically switch applications and users from a primary to a backup node within a cluster for planned or unplanned outages.

The iSeries clustering framework is built around a set of system APIs, system services, and exit programs. This clustering architecture calls for teamwork between IBM and Business Partners to provide the total solution.

IBM clustering initiatives include alliances with cluster middleware business partners and independent software vendors (ISVs) and the development of standards for cluster management utilities. See Chapter 8, “ClusterProven for iSeries applications” on page 161, to understand these standards.

4.6.1 OS/400 integrated cluster services

This section discusses the Cluster Resource Services provided within OS/400, as illustrated in Figure 4-1.

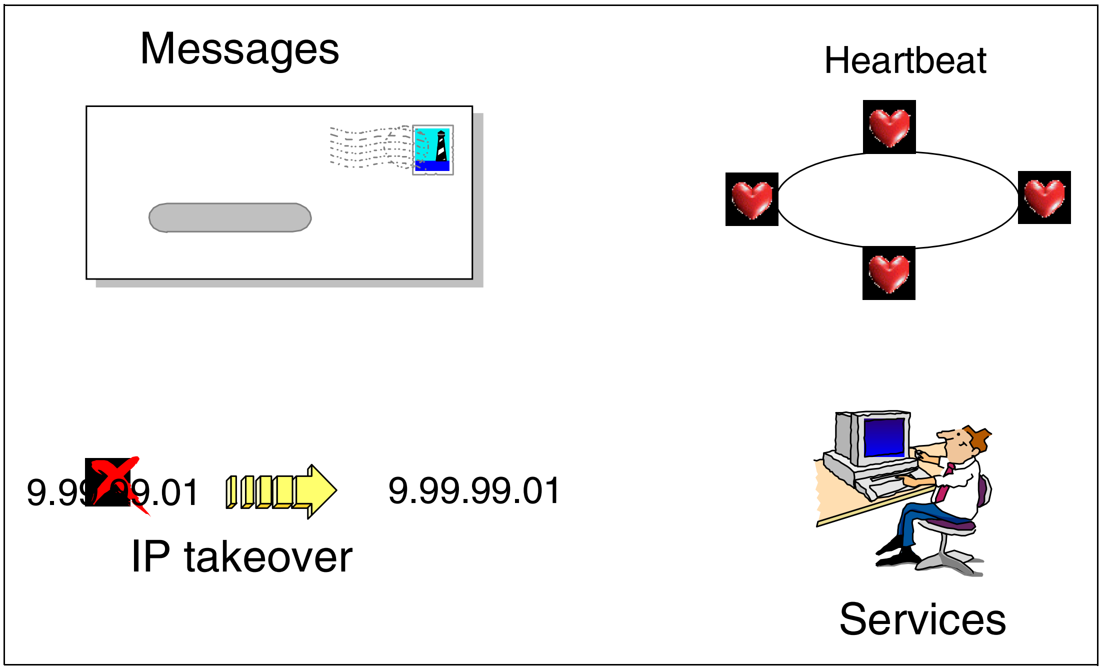


Figure 4-1 OS/400 integrated cluster services

Message function

The message function of Cluster Resource Services keeps track of each node in a cluster and ensures that all nodes have consistent information about the state of cluster resources. Reliable messaging uses retry and time-out values that are unique to clustering. These values are preset and can be changed with an API. The values are used to determine how many times a message is sent to a node before a failure or partition situation is signaled.

For a local area network (LAN), the amount of time it takes to go through the number of retries before a failure or partition condition is signaled is approximately 45 seconds. For a remote network, more time is allowed to determine whether a failure or partition condition exists. Estimate approximately four minutes and 15 seconds for a remote network.

See A.2.2, “Distributed activities” on page 277, for further information.

Heartbeat monitoring

Heartbeat monitoring ensures that each node is active. When the heartbeat for a node fails, the condition is reported so the cluster can automatically fail over resilient resources to a backup node. A heartbeat message is sent every three seconds from every node in the cluster to its upstream neighbor. In a network, the nodes expect acknowledgment to their heartbeat from the upstream node as well as incoming heartbeats from the downstream node, thus creating a heartbeat ring. By using routers and relay nodes, the nodes on different networks can monitor each other and signal any node failures.

If a node fails or a break occurs in the network, heartbeat monitoring tries to re-establish communications. If communications cannot be reestablished within a designated time, heartbeat monitoring reports the failure to the rest of the cluster.

See A.2.1, “Heartbeat and cluster communication” on page 276, for further information.

IP takeover

IP takeover is the ability of a backup system to take over the IP address of a primary system in case of a failure on the primary system.

OS/400 cluster service jobs

Cluster service jobs are a set of multithreaded jobs supplied with OS/400. When clustering is active on an iSeries, the jobs run in the QSYSWRK subsystem. The jobs run using the QDFTJOBD job description. Should any Cluster Resource Services job fail, no job log is produced. In order to provide a job log, change the LOG parameter of the job description to a logging level that produces job logs.

See A.2.3, “Job structure for Cluster Resource Services” on page 278, and A.2.4, “Cluster engine services” on page 279, for further information.

4.6.2 Cluster Resource Services structure

On the iSeries server, the clustering infrastructure is called Cluster Resource Services. Figure 4-2 shows the key elements of OS/400 Cluster Resource Services and their relationship.

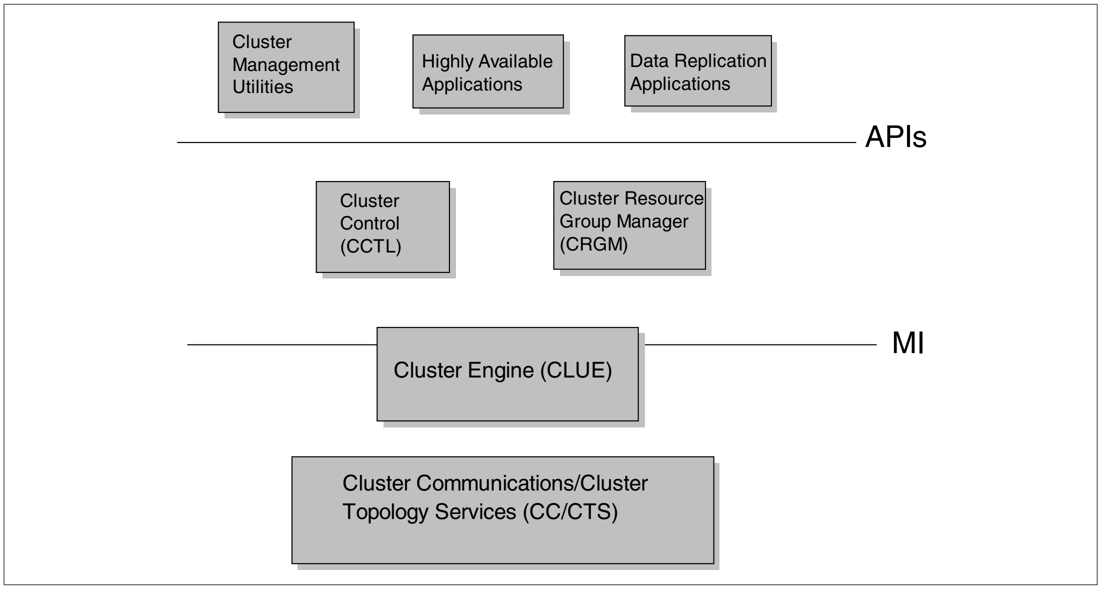

Cluster Management Utilities
Highly Available Applications
Data Replication Applications
APIs
Cluster Control (CCTL)
Cluster Resource Group Manager (CRGM)
MI
Cluster Engine (CLUE)
Cluster Communications/Cluster Topology Services (CC/CTS)

Figure 4-2 Cluster Resource Services structure

The two boxes labeled Cluster Control and Cluster Resource Group Manager in Figure 4-2 represent OS/400 services that provide APIs. The APIs enable business partners, independent software vendors, and application providers to deliver a cluster management utility, data resilience through replication, and resilient (highly available) applications.

The APIs are documented in *System API Reference*, SC41-5801, which is available from the iSeries Information Center at:
http://publib.boulder.ibm.com/pubs/html/as400/onlinelib.htm

Cluster control

Cluster control provides configuration, activation, and management functions for the cluster and the nodes in the cluster. The cluster definition, configuration, and state information is maintained in a persistent internal object called a *cluster information object*. This object exists on each node in the cluster.

Upon request, cluster control starts clustering on a node and coordinates the process of joining that node into the cluster. This process ensures that all nodes are equally aware of the action and have the same content in their cluster information object. Cluster control also manages the merging of cluster partitions.

For further information, see A.2.5, “Cluster partition state” on page 279.

Cluster Resource Group Manager

Cluster Resource Group Manager provides object management functions to create, delete, and modify CRG objects.

A CRG is an OS/400 object that defines and controls the behavior for a group of cluster resources across a recovery domain. Conceptually, the CRG is a distributed object. It exists on all nodes in the defined recovery domain. Each node in the recovery domain has a defined role of primary, backup, or replicate. The nodes in the recovery domain and their respective roles are defined in the CRG object.

When a cluster event occurs that affects that CRG, a user-specified exit program is called on every active node in the recovery domain. A cluster event can add a node to the cluster, change a recovery domain, or cause a node to go offline.

The CRG exit program is identified in the *CRG object. Since the exit program provides resource-specific processing for the cluster event, it can be considered the resource manager for the group of resources associated with that CRG. There can be many CRGs on a node, each potentially with a different recovery domain.

The cluster control and Cluster Resource Group manager components use lower-level system services (OS/400 Licensed Internal Code) to ensure consistency, such that:

- The content of all control objects are logically identical across the affected nodes.
- Cluster activity is coordinated across the affected nodes.

The two boxes labeled Cluster Engine and Cluster Communications/Cluster Topology Services in Figure 4-2 identify the functions that provide these system services.

The *Cluster Engine* provides reliable group communications for the distributed processing needed by the other cluster components to achieve coordinated, distributed, and synchronized activity across multiple cluster nodes. The cluster engine services include group membership services and group messaging services. Most of the cluster engine is implemented below the Machine Interface (MI) to achieve high efficiency, better performance, and better integration with other communication components in the streams stack.

See A.2.4, “Cluster engine services” on page 279, for further information.

Cluster communications

Cluster communications provides low-level internode communications support for the rest of the Cluster Resource Services. It implements the reliable first in, first out (FIFO) ordered multicast message that takes advantage of the IP multicast support of the underlying network when it is available.

This component guarantees that a multicast message is eventually delivered to all its targets, except in the case of failures.

When cluster communications fails to deliver a message to a target (after exhausting all retry attempts and alternative paths), it considers the target node unreachable (failed or disconnected). In the case where the local node fails before completing a multicast message, there are no guarantees that all targets receive the multicast message.

In addition to multicast messages, cluster communications also supports unreliable unordered messaging, reliable FIFO point-to-point messaging, and unreliable point-to-point messaging. The components used can define many multicast groups, dynamically modify membership of each multicast group, and refer to each multicast group via an identifier (for example, when sending messages). This allows cluster communications to plan message distribution and to maximize parallelism for processing unrelated multicast messages sent in the cluster.

Cluster communications is implemented in the streams stack below the MI to achieve high efficiency, better performance, and better integration with other communication components.

Cluster topology services

Cluster topology services provides a cluster view over existing IP network connectivity. It maintains the knowledge of currently active cluster nodes and cluster nodes known to be partitioned. Two paths can be defined to each node in the cluster. The first path to the node specified on the cluster control API is considered the preferred (primary) path.

A *partition* is the situation where the connection between nodes is lost, but none of the nodes fail. This situation is described in more detail in A.2.5, “Cluster partition state” on page 279.

Cluster topology services continuously checks connectivity of the various network paths and allows a seamless switch to the alternative path when the preferred path is not available. It also allows a seamless switch back to the preferred path when it becomes available again. Cluster topology services periodically checks connectivity to partitioned nodes to see if connectivity is re-established. When successful, cluster topology services notifies cluster control and the cluster engine, which then attempt to merge partitions back into the cluster.

Part of the continuous check performed by cluster topology services is *heartbeating,* which performs periodic checks on liveness and connectivity of the locally reachable cluster nodes and delivers failure notifications. When a previously connected node becomes unreachable, cluster topology services notifies the cluster engine. The cluster engine then removes the node from the locally visible cluster or declares a partition.

For an in-depth discussion of the structure of OS/400 cluster services, refer to A.1, “Underlying technologies” on page 276.

4.7 Cluster components

A cluster is made of these elements:

- Cluster nodes
- Cluster resources
- Recovery domains
- Cluster management support and clients

These elements are illustrated in Figure 4-3 and explained in the following sections.

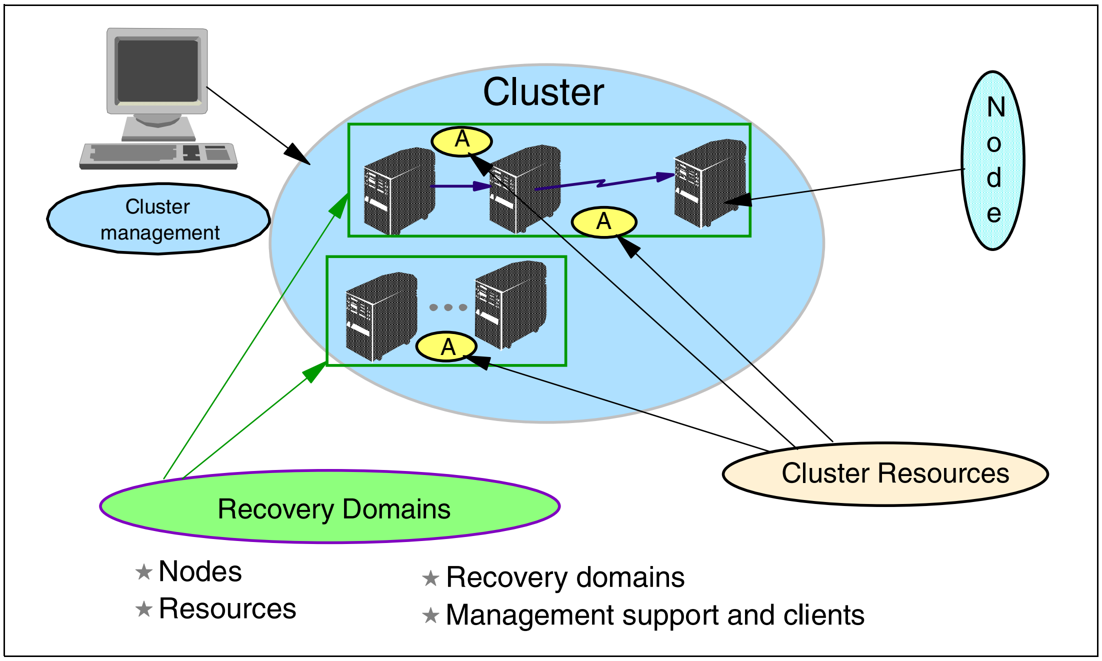


Figure 4-3 Components of clustering

4.7.1 Cluster nodes

A *cluster node* is any iSeries server or partition that is a member of a cluster. Cluster nodes must be interconnected on an IP network.

A cluster node name is an eight-character cluster node identifier. Each node identifier is associated with one or more Internet Protocol (IP) addresses that represent an iSeries server. Any name can be given to a node. However, for simplicity, make the node name the same as the system name.

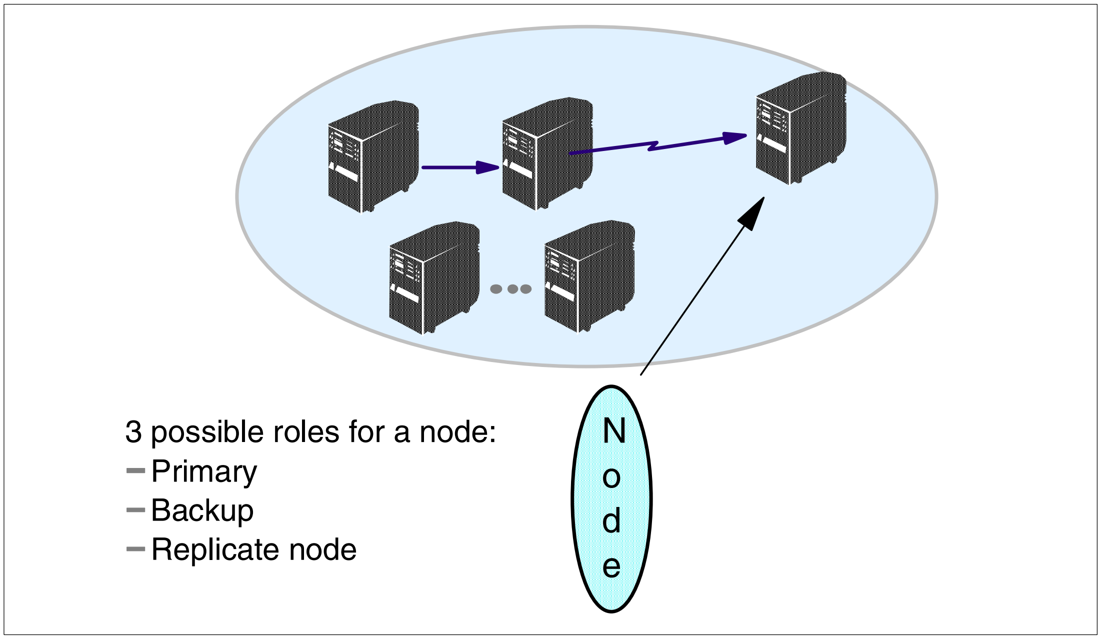
Cluster communications that run over IP connections provide the communications path between cluster services on each node in the cluster. The set of cluster nodes that are configured as part of the cluster are referred to as the *cluster membership list*.

A cluster consists of a minimum of two nodes. The environment can be extended to a cluster with a maximum of 128 nodes.

A node of a cluster can fill one of three possible roles within a recovery domain, as shown in Figure 4-4. The roles and associated functions are:

- **Primary node**
  - Point of access for resilient device
  - Contains principal copy of any replicated resource
  - Current owner of any device resource
  - All CRG objects fail over to a backup node
- **Backup node**
  - Can take over the role of primary access at failure of the current primary node
  - Contains copy of cluster resource
  - Copies of data are kept current via replication

- **Replicate node**
  - Has copies of cluster resources
  - Unable to assume the role of primary or backup (typically used for functions such as data warehousing)

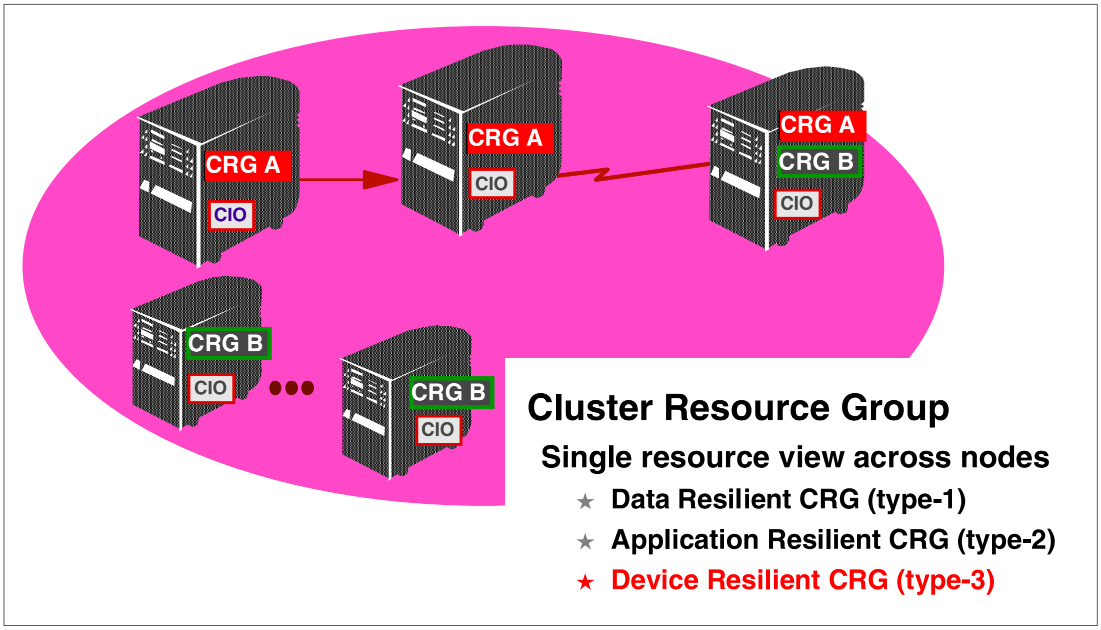


3 possible roles for a node:
- Primary
- Backup
- Replicate node

Figure 4-4 Cluster nodes

4.7.2 Cluster Resource Groups (CRG)

A *Cluster Resource Group* is an OS/400 external system object that is a set or grouping of cluster resources. The Cluster Resource Group (and replication software) is a foundation for all types of resilience. See Figure 4-5 for an illustration.



Cluster Resource Group

Single resource view across nodes

- ★ **Data Resilient CRG (type-1)**
- ★ **Application Resilient CRG (type-2)**
- ★ **Device Resilient CRG (type-3)**

Figure 4-5 Cluster Resource Group

Resources that are available or known across multiple nodes within the cluster are called *cluster resources*. A cluster resource can conceptually be any physical or logical entity (database, file, application, device, and so forth). Examples of cluster resources include iSeries objects, IP addresses, applications, and physical resources. The objects labeled CRG A and CRG B in Figure 4-5 represent cluster resources. When a cluster resource persists across an outage, that is any single point of failure within the cluster, it is known to be a *resilient resource*. As such, the resource is resilient to outages and accessible within the cluster even if an outage occurs to the node currently “hosting” the resource.

Cluster nodes that are grouped together to provide availability for one or more cluster resources are called the *recovery domain* for that group of cluster resources. A recovery domain can be a subset of the nodes in a cluster, and each cluster node may actually participate in multiple recovery domains.

Resources that are grouped together for purposes of recovery action or accessibility across a recovery domain are known as a *Cluster Resource Group*. The Cluster Resource Group defines the recovery or accessibility characteristics and behavior for that group of resources.

A CRG describes a recovery domain and supplies the name of the Cluster Resource Group exit program that manages cluster-related events for that group. One such event is moving the users from one node to another node in case of a failure.

There are three Cluster Resource Group object types that are used with Cluster Services at V5R1:

- **Data resilient**: A data resilient CRG enables data resiliency, so that multiple copies of data can be maintained on more than one node in a cluster.
- **Application resilient**: An application resilient CRG enables an application (program) to be restarted on either the same node or a different node in the cluster.
- **Device resilient**: A device resilient CRG enables a hardware resource to be switched between systems. The device CRG is represented by a (device) configuration object as a device type of *independent ASP* (IASP). Device resilient CRGs are supported with OS/400 V5R1.

Each CRG definition object specifies the cluster exit program to be called. The exit program is responsible for handling the action codes passed to it by the Cluster Resource Group Manager. Action codes are managed in the APIs that interact with the applicable CRG. And the Cluster Resource Group exit program manages the movement of the access point of a resilient resource.

Exit programs are written or provided by high availability business partners and by cluster-aware application program business partners. See 4.7.5, “Exit programs” on page 47, for a further discussion.

4.7.3 Recovery domains

A *recovery domain* is a subset of nodes in the cluster that are grouped together in a Cluster Resource Group for purposes such as performing a recovery action. Each Cluster Resource Group has a recovery domain that is a subset of the nodes in the cluster. See Figure 4-6 for an illustration.

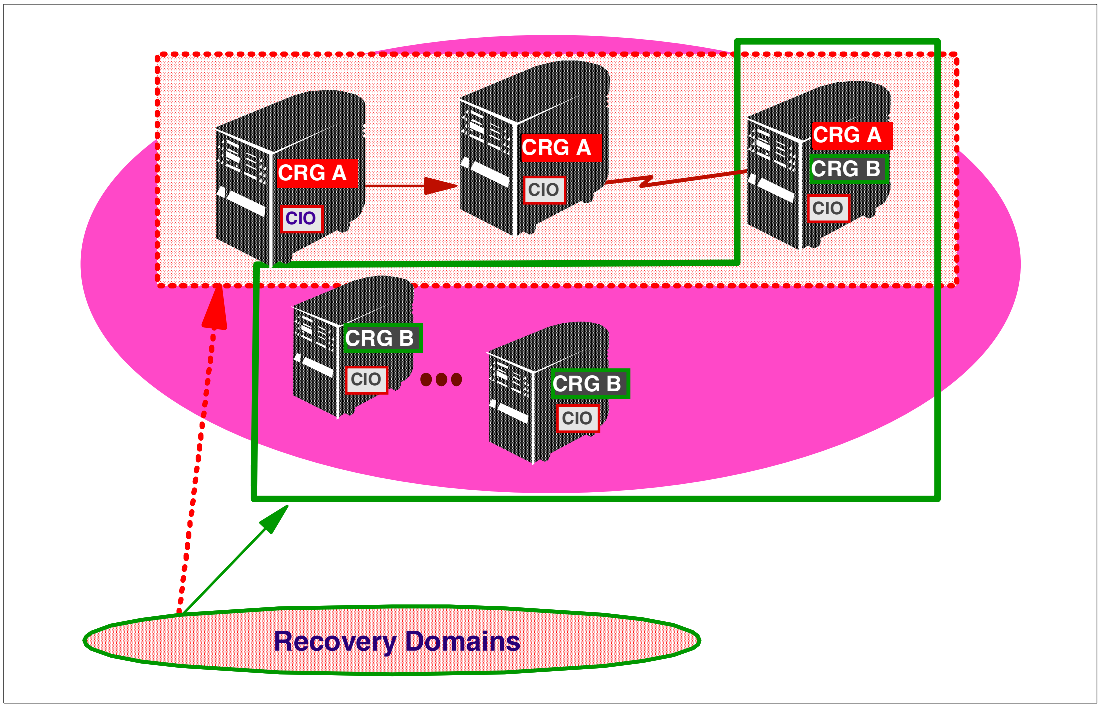


Figure 4-6 Recovery domain

Here are some facts about recovery domains:

- The nodes within a recovery domain participate in any recovery actions for the resources of the domain.
- Different CRGs may have different recovery domains.
- As a cluster goes through operational changes (for example nodes end, nodes start, nodes fail), the current role of a node may change. Each node has a preferred role that is set when the CRG is created.
- A recovery domain can be a subset of the nodes in a cluster, and each cluster node may participate in multiple recovery domains.

4.7.4 Device domains

The construct known as a *device domain* (Figure 4-7) is a subset of cluster nodes that share a set of resilient devices. A resilient device might be an independent ASP.

A function of a device domain is to prevent conflicts that would cause the failure of an attempt to switch a resilient device between systems.

Resources involved in a device domain include the structures used to identify and manage the content of the structures across the multiple systems involved in the domain. These structures are described in Chapter 6, “Independent ASPs explained” on page 99.

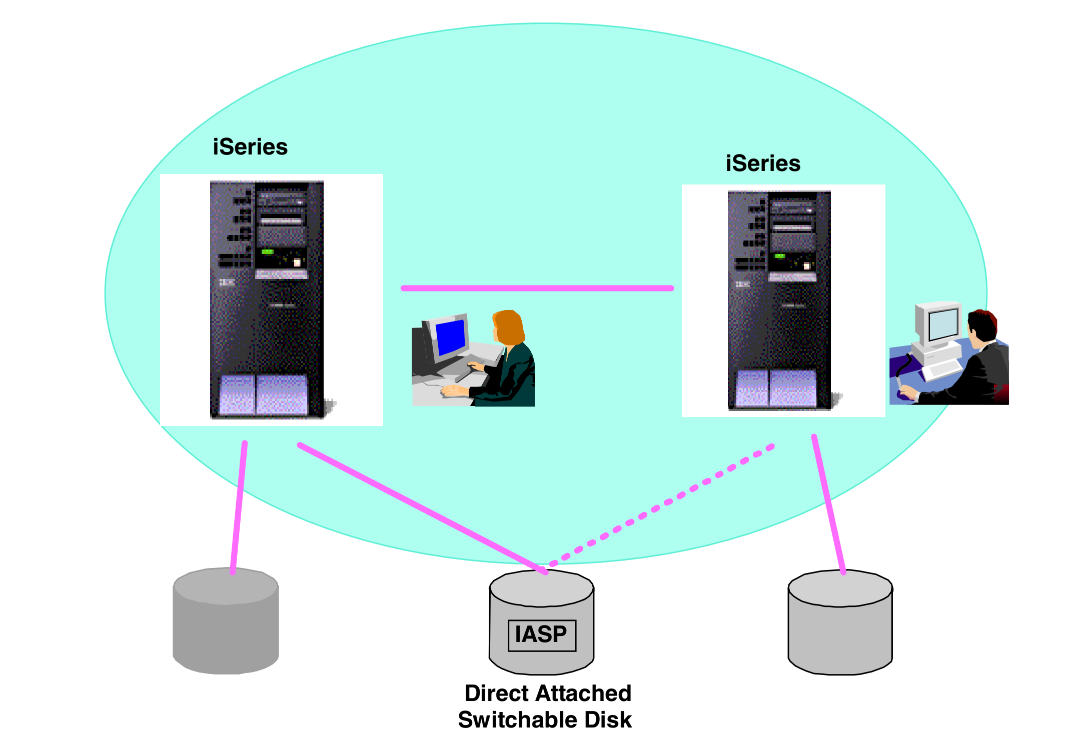
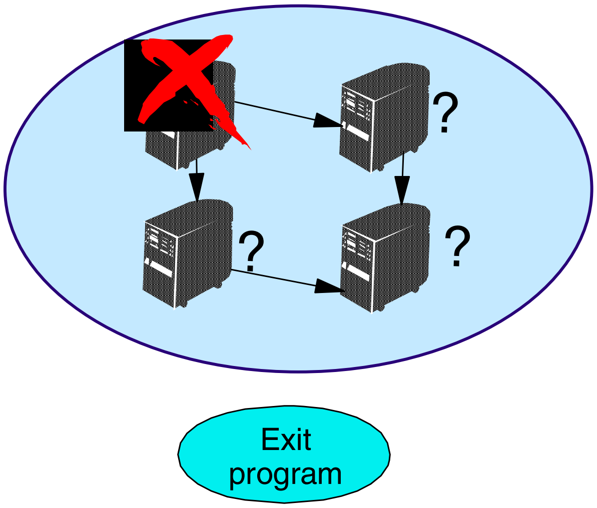


Figure 4-7 Device domain

4.7.5 Exit programs

The main purpose of exit programs (Figure 4-8) is to "tell" each node in the cluster what to do in case of a failure on the primary system.



- Called when changes occur in the recovery domain
- Specify which action to take on the other nodes

Figure 4-8 Exit program

When a change occurs in the recovery domain, the exit program associated with the CRG is called on all the active nodes in the recovery domain. Changes range from a system failure to a planned switchover from one system to another, to the addition of a new node to the recovery domain.

The exit program is also called when other events happen, such as when the CRG is started or ended or when an exit program fails. When an exit program is initiated, OS/400 passes the program an action code indicating the event that caused the program call.

Exit programs with data CRGs

An exit program associated with a data CRG must ensure that as much data as possible (for example, any final journal entries) is transferred to the backup system in the event of a switchover. On the backup system (the new primary system), all outstanding journal entries must be applied. Any other objects must be synchronized as well. When a new node is added to a recovery domain, the exit program may handle the initial data replication.

For example, when a switchover is initiated for a data CRG, the cluster middleware software for data replication writes any remaining journal entries to the backup system. (Remote journaling eliminates this step). Then, when the exit program is called on the backup system (changing its role to primary), the cluster middleware software applies any outstanding journal entries and synchronizes the non-database objects. The software establishes tasks to replicate data from the new primary system to the next backup node, if necessary. Cluster middleware providers are committed to enhancing their products to use the new clustering support (for example, to call the appropriate functions from the exit program), which lets them take advantage of system services such as heartbeat monitoring.

In most cases, a cluster middleware provider supplies data CRG exit programs that take care of all the functions mentioned previously. This means that when a customer wants OS/400 functions combined with data replication, the link between these is already provided.

Exit programs with application CRGs

Exit programs associated with application CRGs are particularly critical, because the application must be restarted on the backup system. OS/400 supplies the exit program with information about the node status change, but all application-specific details, such as current users, record pointers, and even which programs are active, must be handled by the application developer.

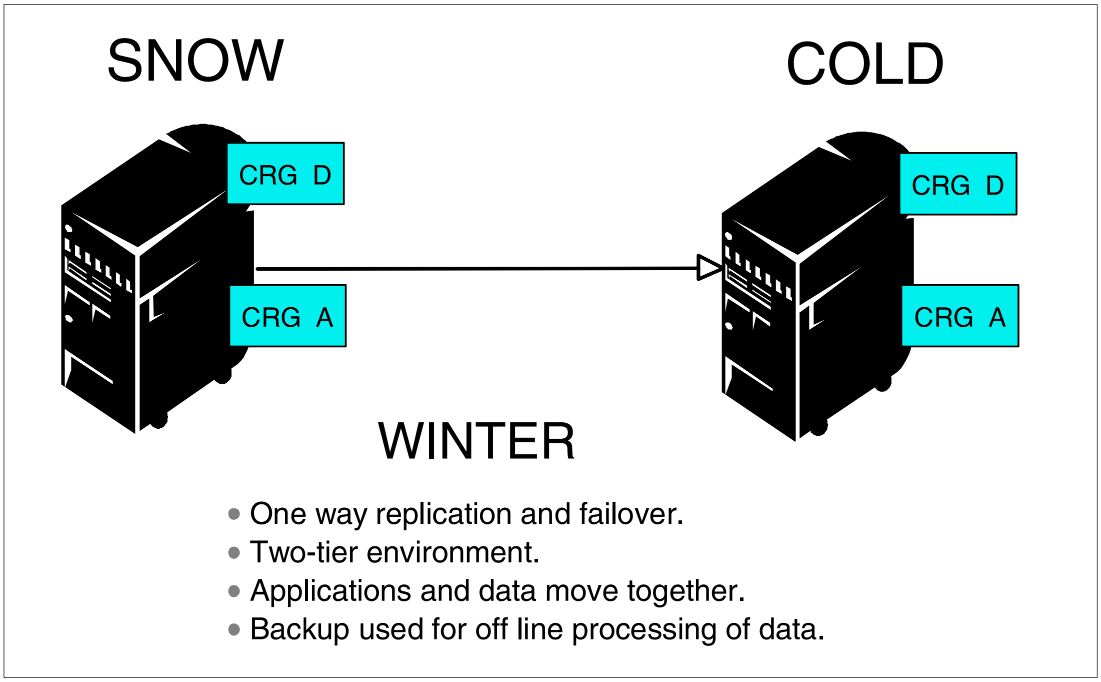
Note: It is not mandatory for a device CRG to have an associated exit program. It is for application and data CRGs.

4.8 Cluster implementation example

To illustrate the concepts described in this chapter thus far, this section provides examples of cluster implementation: a simple two-node cluster and a four-node cluster.

4.8.1 Simple two-node cluster example

For this simple cluster example, there is a cluster named WINTER with two cluster nodes, named SNOW and COLD. See Figure 4-9 for an illustration.



- One way replication and failover.
- Two-tier environment.
- Applications and data move together.
- Backup used for off line processing of data.

Figure 4-9 Simple two-node cluster

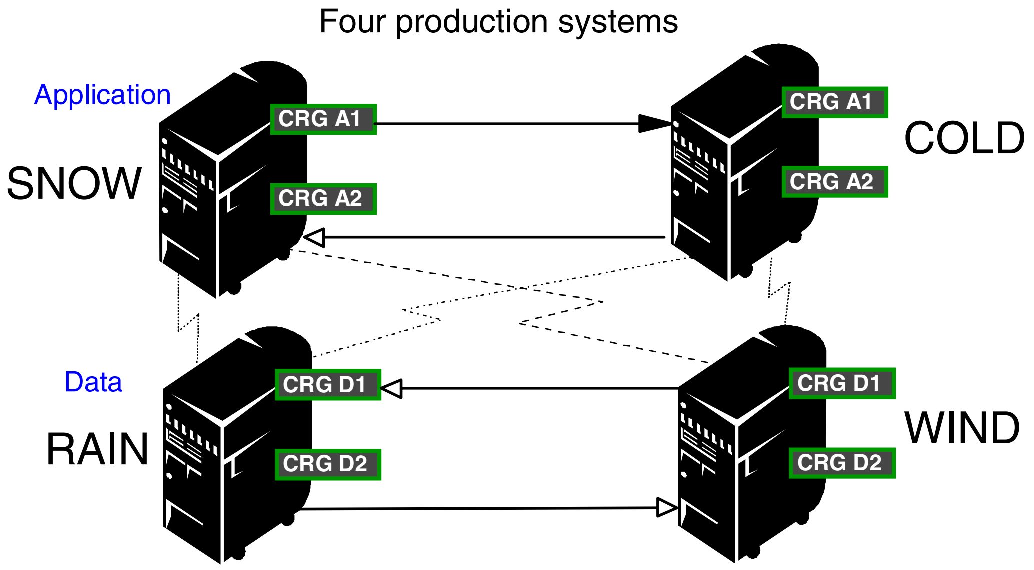
Node SNOW operates as the primary node for two Cluster Resource Groups called CRG A and CRG D. CRG A is an *application* Cluster Resource Group. CRG D is a *data* Cluster Resource Group. Node COLD is the first (and only) backup for both of the luster resource groups.

Data that is associated with CRG D and pertinent application information associated with CRG A are replicated from the node named SNOW to the node named COLD. If Node SNOW fails or needs to be taken down for administrative reasons, then Node COLD becomes the primary node for both Cluster Resource Groups CRG A and CRG D. Node COLD takes over the Internet Protocol address defined for CRG A.

> **Note:** While Node SNOW is down, system availability is exposed because there is no backup if Node COLD also fails. When Node SNOW recovers and rejoins the cluster, it becomes the backup for both Cluster Resource Groups. At that time, replication is from the node named COLD to the node named SNOW.
>
> To make SNOW the primary node, perform an administrative switchover.

4.8.2 Four-node mutual takeover cluster example

A four node example shows the additional flexibility that is possible with an iSeries cluster. In this example, there are two Application Cluster Resource Groups (CRG A1 and CRG A2) and two Data Cluster Resource Groups (CRG D1 and CRG D2). The data associated with CRG D1 is the critical data for the application associated with CRG A1. The data associated with CRG D2 is the critical data for the application associated with CRG A2. See Figure 4-10 for an illustration.



- Two-way replication and failover; three-tier environment.
- Applications and data move independently.
- Backup used for normal production of different workloads.

Figure 4-10 Four-node mutual takeover cluster

Because this is a three-tier environment, the applications exist on the second tier (Node SNOW and Node COLD) and the data is separated into the third tier (Node RAIN and Node WIND).

For CRG A1, Node SNOW is the primary and Node COLD is the backup. At the same time, Node COLD is the primary for CRG A2 and Node SNOW is its backup. For CRG D1, Node WIND is the primary and Node RAIN is the backup. Also, Node RAIN is the primary for Data CRG D2 and Node WIND is its backup. This enables mutual take over capability at both the application and data levels.

All four nodes are used for normal production. The nodes are also used to back up other systems in the cluster. The two applications and their associated data would always be available in this cluster.

The outage of any single node does not disrupt availability. In addition, the simultaneous outage of a node at the application level with a node at the data level does not disrupt availability.

You can find further examples in Chapter 10, "Sample cluster environment and applications" on page 183.

4.9 Resiliency

To achieve continuous availability, more than robust system availability is needed. Critical data and critical applications must also be *resilient* to outages.

A complete resiliency solution is achieved when the critical data and the critical applications are made to be resilient resources and are always available.

Clustering depends on two interrelated concepts: *data resiliency* and *application resiliency*. Both must be accessible across the cluster even when the normal hosting system for the resource fails

Data resiliency ensures that the backup system has all the information necessary to run critical production jobs when control is transferred from the primary system. Data resiliency requires synchronizing objects across the nodes in the Cluster Resource Group. Cluster middleware business partners have many tools to deliver iSeries data resiliency. IBM supports the business partner tools rather than to create a contending data resiliency solution.

Existing high-availability solutions synchronize data files, programs, and related objects such as data areas, job queues, and user profiles. The cluster middleware solutions use a combination of custom applications and OS/400 functions (for example, remote journaling). All these functions are needed to support clustering.

A solution that focuses only on data cannot be available 24 x 365. Switching between systems requires application resiliency and transaction signalling. This is why clustering technology was introduced at V4R4 and why the focus is now expanded to include the application and the data together in a comprehensive solution called the cluster.

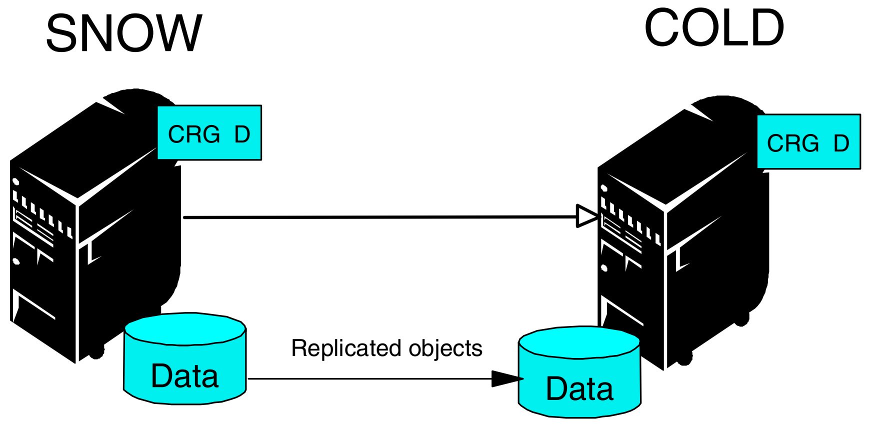
Application resiliency ensures that the services provided by the application are accessible to end users of the cluster. Resilience is provided through an IP address takeover and a restart of the application on the backup system. Application resiliency can be defined as the ability to run an application on more than one node in a cluster.

Ideally, when an application switches from one node to another, the user experiences no disruption at all and is not even aware that the job has been switched to a different server. Realistically, the disruption the user experiences can range from a slight delay to an extensive application restart. The user may have to sign on to the new server, restore or resynchronize data, restart the application, and re-enter any partially completed transactions. The more resilient an application is, the more this disruption is minimized.

For a full discussion of application resiliency, see Chapter 8, “ClusterProven for iSeries applications” on page 161.

4.9.1 Data resiliency in a simple two-node cluster example

Figure 4-11 shows an example of a cluster with two nodes: SNOW and COLD. SNOW is the primary machine. The Data CRG “D” represents data resiliency. Data objects are replicated between the two systems.

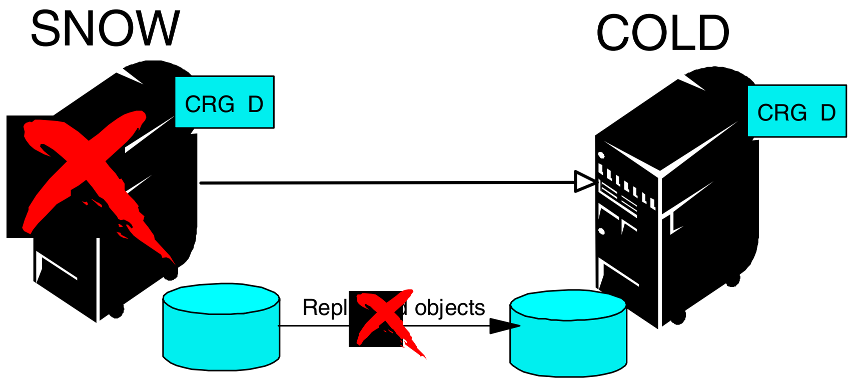


Data CRGs represent data resiliency
Data objects are replicated between systems

Figure 4-11 Data resiliency in a simple two-node cluster

Figure 4-12 illustrates what happens in the event of a failure at Node SNOW.

Assume that SNOW encounters a failure



- When SNOW becomes inactive, COLD must assume a current node role of primary
- No replication takes place

Figure 4-12 Failure in a simple two-node cluster

When Node SNOW encounters a failure, Node COLD must take over as the primary node.

> **Note:** In either instance, the cluster runs exposed in that some cluster resources are not replicated while a node is down. Resolve this by having more than one backup for any critical cluster resource.

4.9.2 Application resiliency and IP address takeover example

Figure 4-13 shows an example of a cluster with two nodes. The WINTER_cluster has data resiliency, meaning that data is available at both systems. In this case, production jobs can run on the backup system (COLD) in case of a failure at the primary system (SNOW).

The level of resiliency beyond data resiliency is application resiliency. Application resiliency means that the application can transfer the user to the backup system in case of a failure in the primary system.

An important feature in application resiliency is *IP takeover*. Figure 4-13 illustrates a setup for an IP address takeover. The client uses an IP address associated with an application CRG.

The IP address specified in the CRG must not be in use on any other node in the cluster. In this example, the TCP/IP interface with the same address as that of Node SNOW is inactive on Node COLD, the backup system. The IP address is started on the node that has the current role of primary for the application CRG.

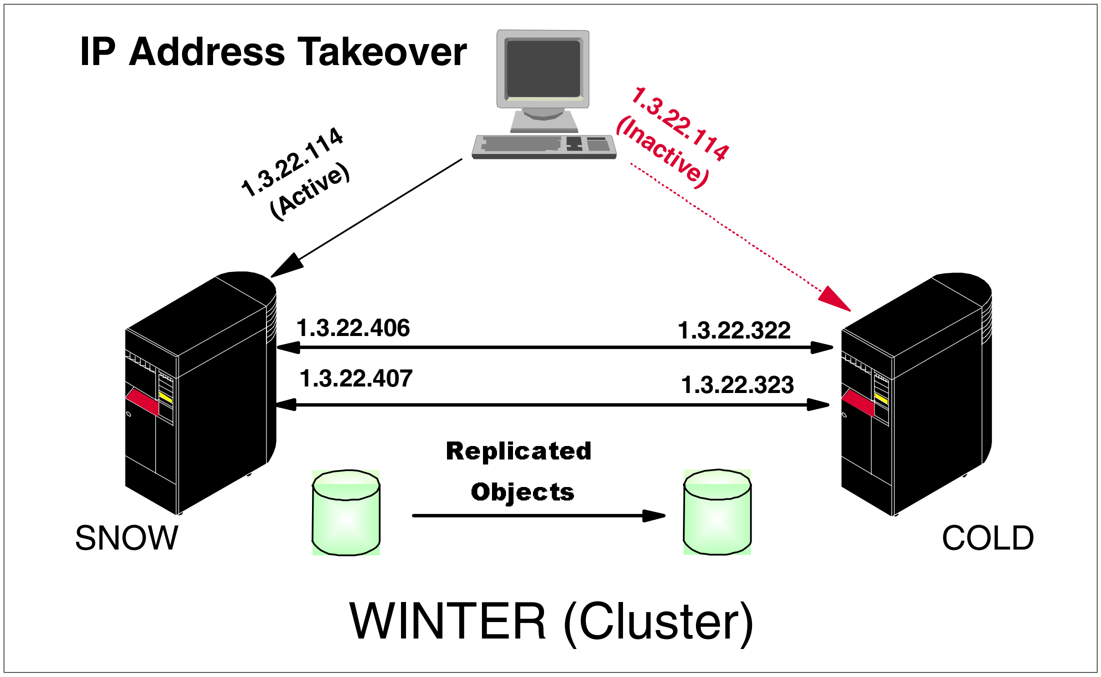


Figure 4-13 Application resiliency and IP address takeover in a simple two-node cluster

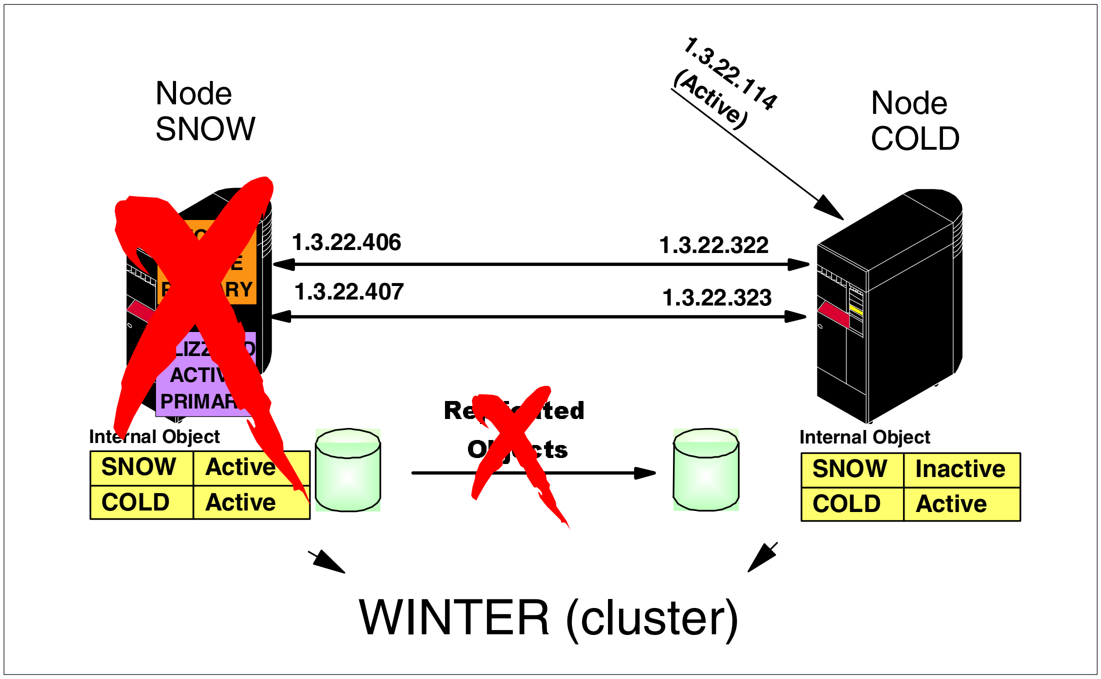
4.9.3 Managing application resiliency

Figure 4-14 illustrates what happens to the user in case of a failure at the System (Node) named SNOW. This is an example of application resiliency and IP takeover after failover processing is complete.

Node SNOW fails, and Node COLD assumes the role of primary. The TCP/IP interface 1.3.22.114 is now active on Node COLD, and the users are switched to Node COLD.

Note: In this case, the risk involved in a failure at Node COLD is a concern, because it has no backup and no data is being replicated from Node COLD to another backup machine.

When Node SNOW is back up again, objects are replicated from Node COLD (temporarily the primary system) back to Node SNOW.



Internal Object

SNOW	Active
COLD	Active

Internal Object

SNOW	Inactive
COLD	Active

Figure 4-14 Application resiliency and IP address takeover after a failure

Once replication is finished to the point where the two nodes are in sync, another cluster node operation can switch the users back to run on the node named SNOW, to again become the current primary.

> **Important:** Application resiliency is important. It is equally important to recognize that in order to obtain continuous availability, the applications have to be designed in a way that allows them to return to their previous known failure state. In other words, the job state and the application state have to be maintained.

4.9.4 ClusterProven applications

A ClusterProven application is one that allows the user be switched automatically to the backup system in case of a failure on the primary system. It also lets the user resume work in the same screen the user was working in at the primary machine.

To understand ClusterProven applications for the iSeries server, refer to Chapter 8, “ClusterProven for iSeries applications” on page 161.

4.10 iSeries cluster solution components

The infrastructure of clustering support is comprised of several functions. These universal functions involve:

- Support to enable the base means of clustering, to define the:
  - Cluster
  - Cluster membership list
  - Means to access each cluster node

- Support for groups of nodes to:
  - Define groups of cluster resources
  - Manage groups of cluster resources
  - Ensure node accessibility
  - Determine when an outage occurs
  - Distribute information or messages across groups of cluster nodes
- Support for the infrastructure of clustering to:
  - Ensure that internal cluster information and external objects are synchronously updated across all affected cluster nodes

Figure 4-15 illustrates the components of an iSeries total cluster solution.

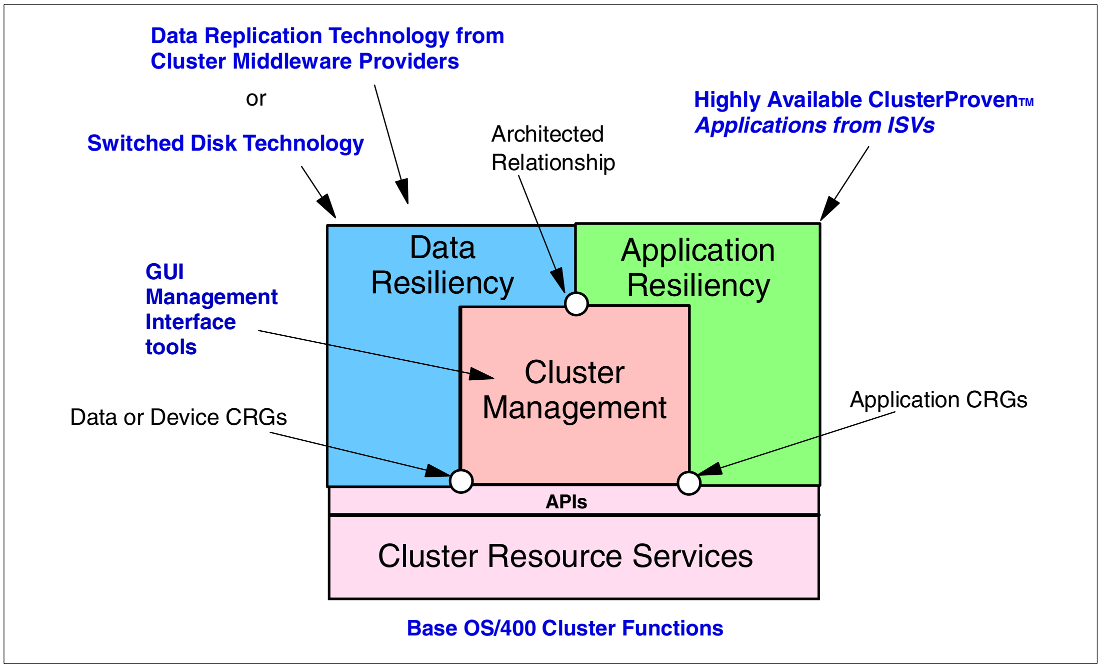


Figure 4-15 iSeries cluster solution components

For a total iSeries clustering solution, note these points:

- Data resiliency is provided by data replication solutions from cluster middleware providers or switched disk technology.
- Application resiliency is provided by highly available ClusterProven™ applications from ISVs.
- Cluster management is performed with a GUI management interface from a cluster middleware provider or within IBM in OS/400 V5R1.
- Cluster Resource Services are provided by IBM in OS/400.

Notice the inclusion in Figure 4-15 of various cluster middleware business partners. The interdependent relationships between IBM, cluster middleware providers, and application developers are central to the rollout of clustering with OS/400. Each participant's role is clearly defined in 11.6, “Roles and responsibilities when implementing clustering” on page 211.

4.11 iSeries clusters: Cluster architecture versus data replication

Although it is possible to build a “cluster like” environment without taking advantage of the ClusterProven Architecture and ClusterProven Applications, it is much more difficult.

For example, when building the business clustering environment, you must:

1. Select a tool to replicate critical data from the production system to the backup system.

   Replication tools are available from one of the IBM cluster middleware business partners, DataMirror, Lakeview Technology, or Vision Solutions. See Part 3, “Cluster middleware business partners” on page 227, for a discussion of these solutions.

2. Assuming the physical environment is prepared (that is, a backup server is installed and connectivity is established), determine what needs to be replicated to the backup system.

   This requires a thorough knowledge of the application design. This can be a very complex task that can take a long period of time. In fact, often times it is a work in progress as time goes on. Because of this, the integrity and reliability of the High Availability environment can be jeopardized. Do not be fooled. This is a part of the project that can take weeks, even months, to complete.

3. Ensure that everything to be replicated is journaled. If not, create the journal environment and begin journaling on those files and objects.

   With journaling, the cluster middleware product knows what objects to replicate, based on the changes made.

4. Synchronize data between the systems.
5. Replicate the data.

The project is not done at this point. There is plenty more to do to create a “cluster like” environment. For example, you need to perform these tasks:

- With data replicated to the backup system, in both planned and unplanned situations, programs are needed to switch the users from the production system to the backup system.
- Programs are needed to restart the application following a switch.
- In an unplanned scenario, a manual process must be in place to determine what transactions are lost and never make it to the backup system.
- Depending on the application, some manual steps can be needed to restart the application and to determine where certain processes should begin.
- Implement monitors for the production system to alert the operator of a failure of the system. The operator can then determine if a switch is necessary.

  A clustering solution with replication services has the architecture for automatic failovers. Use replication services to avoid a need for operator (manual) intervention for a switchover.

Therefore, as described, it is possible to create a “cluster like” environment.

However, to build the final solution around the data replication involves a lot of work. That is the fundamental difference between a “cluster like” environment and iSeries clusters. iSeries cluster architecture takes the tasks that otherwise would be built and maintained in-house, and makes them part of the architecture. That is:

- An application that is ClusterProven provides the cluster middleware providers with all of the objects the application needs replicated.

- Functions to monitor the cluster are provided.
- There is a GUI environment to manage the clusters in the entire enterprise rather than switching between character-based screens on each system.
- All of the processes to restart the application are provided via application CRG exit programs.

Stated another way, IBM works with the cluster middleware providers and ISVs to make a customer's HA environment more reliable by moving critical function out of user-written high-level language (HLL) programs and into OS/400. These relationships will continue to grow to enhance and strengthen the cluster architecture, which in turn, makes the HA environment more manageable and reliable.

5

Implementing and managing clusters with IBM solutions

IBM and IBM Cluster Middleware Business Partners team together to provide state-of-the-art Cluster Resource Services functions, along with a graphical user interface (GUI) for cluster management. This relationship is illustrated in Figure 5-1 and further described in this chapter.

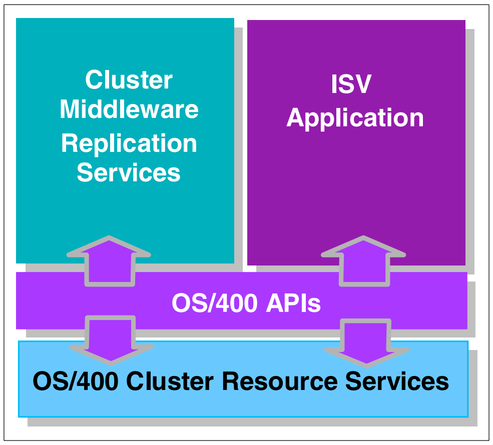

Cluster Middleware Replication Services
ISV Application
OS/400 APIs
OS/400 Cluster Resource Services

Figure 5-1 iSeries cluster implementation open architecture

OS/400 Cluster Resource Services (CRS) provides a set of integrated capabilities to maintain cluster topology, perform heartbeating, and create and administer cluster configuration and Cluster Resource Groups (CRGs). CRS provides reliable messaging functions that keep track of each node in the cluster, and ensure that all nodes have consistent information about the state of cluster resources. In addition, Cluster Resource Services provides a set of application program interfaces (APIs) and facilities that can be used by iSeries application providers and customers to enable cluster architecture and enhance application availability.

A set of clustering commands is provided in the QUSRTOOL library. QUSRTOOL library is shipped as part of OS400 V5R1. For each clustering API available with OS/400, there is a corresponding cluster command available in the QUSRTOOL library.

The QUSRTOOL cluster commands are most useful in simple clustering environments. With the commands available in QUSRTOOL, a customer can easily set up a cluster to test cluster enabled applications.

Note: Cluster middleware solutions are the preferred method to implement a more advanced clustering solution.

This chapter describes the steps necessary to create and manage clusters. The discussion covers the IBM-supplied methods available with OS/400 V5R1 – the GUI that is part of Operations Navigator and the cluster APIs and commands that are provided in the QUSRTOOL library.

Note: For more information, you can find a cross-reference of the QUSRTOOL command name with the corresponding clustering API in B.3, “Cluster APIs and related QUSRTOOL commands” on page 286.

5.1 Using the Operations Navigator GUI or commands, APIs to implement, manage iSeries clusters

Either the Operations Navigator GUI, or commands and OS/400 APIs, can be used to implement and control iSeries clusters. Each method has its positive and negative aspects.

Consider these points to help determine the better solution for clusters on the iSeries server:

- The Simple Cluster Management GUI is the easiest interface to create simple clusters.

  A two-node cluster can be created with the Operations Navigator GUI. Switchable resources can be added to the cluster and the resources can be managed. The status of a cluster can be determined with minimum effort with the GUI. And the GUI supports a switchover function.

- The clustering APIs and the commands supplied with QUSRTOOLS are more effective when clustering needs extend beyond a simple two-node cluster environment.

  Consider the Cluster APIs or commands supplied with QUSRTOOL library when clustering needs are more complex. The APIs and QUSRTOOL commands allow the user to create up to 128 nodes and create and manage CRGs. Device domains can have resilient switchable devices.

- Use a high availability business partners clustering implementation when cluster requirements and the configuration are more complex.

  Managing a complex cluster can be time consuming. The more complex the cluster configuration and requirements are, the more high availability replication is warranted.

  The cluster middleware providers offer sophisticated cluster management tools to create and manage a complex multi-node cluster with less effort than using APIs and commands, or GUI.

  Refer to Part 3, “Cluster middleware business partners” on page 227, for a description of the cluster middleware providers offerings for cluster management on the iSeries server.

For a quick comparison, the basic functions of cluster support are listed in Table 5-1.

Table 5-1 Clustering functions

Clustering function	OS/400 cluster APIs	Operations Navigator cluster GUI
Create cluster	Yes	Yes
Add application CRG	Yes	Yes
Add data CRG	Yes	No
Add device CRG (switchable hardware)	Yes	Yes
Manage clusters	Limited	Application and device management. Not data.
Extensive help text	No	Yes
Managed nodes	128	2
Wizards	No	Yes

In effect, API usage is not for the ordinary user. A GUI interface is necessary to manage IASPs. And the best solution when data and application resiliency are the goal, is to either develop code in-house, or implement an application business partner offering.

Important: Whatever option is selected to implement a clustering solution, do not mix methods.

If a simple two-node cluster is created with the Simple Cluster Management GUI, continue using the GUI for all activity related to that cluster. Do not attempt to use the APIs and QUSRTOOL commands to further enhance the cluster.

Similarly, if the APIs or QUSRTOOL commands are used to create a more complex cluster, do not attempt to manage or enhance the cluster using the Simple Cluster Management GUI. The GUI tool often combines multiple steps completed with a simple click of a mouse. Many functions are “hidden” by the IBM GUI actions.

If a business partner solution is selected, do not attempt to manage or enhance the cluster using the Simple Cluster Management GUI or the QUSRTOOL commands. Unpredictable results can occur when mixing the GUI with cluster APIs and QUSRTOOL commands. IBM does not provide support for problems encountered in this circumstance.

5.2 Using the Operations Navigator GUI to implement and manage iSeries clusters

When switched disk is used, the GUI interface of Operations Navigator is the recommended interface to manage a cluster of two iSeries’ servers. The other supported interface is described in 5.3, “Using QUSRTOOL CL commands and OS/400 APIs to implement an iSeries cluster” on page 87, and 5.3.1, “Using OS/400 cluster management APIs to implement a cluster” on page 93.

The basic steps to implement a simple cluster using Operations Navigator GUI are:

1. Complete the environment plan.
2. Set up the communications environment for the cluster.
3. Start Management Central.
4. Create the cluster.

The steps are explained in this section. You can find more details on step 3 in “IBM Simple Cluster Management GUI considerations” on page 64. You can also find more information on step 4 in 5.2.2, “Using the Simple Cluster Management GUI to create the cluster” on page 64.

Completing the environment plan

To develop a plan to prepare the environment for clustering, consider these success factors:

- Clearly state the objectives of clustering.

  Make sure that there is a well-defined plan to implement the cluster.
- Identify which nodes are to be in the cluster.
- Be certain that the required prerequisite software is loaded on both systems to be in the cluster.

  Refer to 4.10, “iSeries cluster solution components” on page 54, to understand the software requirements for an iSeries clustering solution.
- Ensure that user profiles exist on all nodes that are to be in the cluster.

  The user profile must incorporate *IOSYSCFG authority.

Setting up the communications environment for the cluster

Prior to implementing a cluster solution, document all system names and IP addresses of the nodes to participate in the cluster. To set up the communications environment for the cluster, follow these steps:

1. Start the TCP/IP servers Internet Daemon (*INETD) and Management Central server (*MGTC).
2. Change the Allow Add to Cluster (ALWADDCLU) network attribute to `*ANY` or `*RQSAUT`.

> **Note:** *ANY means that any other system is allowed to add this system as a node in a cluster. *RQSAUT means that after an exchange of digital certificates is done to validate the cluster add request, another system is allowed to add this system as a node in a cluster.

Starting Management Central

To enable cluster management from Management Central, you must complete these actions:

1. Change the Central System in Management Central to be one of the nodes to participate in the cluster.

> **Note:** The central system is typically the primary node in the cluster, but is not required to be so.

2. Use Operations Navigator to define both nodes as valid connections.

Using the Simple Cluster Management GUI to create the cluster

Follow these steps:

1. Proceeding through the GUI, read all screens carefully and fully.
2. If error screens are encountered, click the **Details** button.
3. Follow the suggestions mentioned in the error text. Errors are typically easily fixed and a “try again” option is provided after the error is answered.

5.2.1 Clustering configuration tasks

To prepare for configuring simple clusters with the Operations Navigator GUI, it is helpful that certain conditions exist in the iSeries setup and configuration. Prior to creating a cluster, make sure the tasks in these checklists are completed.

TCP/IP requirements

__1.TCP/IP must be started on every node chosen to be in the cluster (STRTCP).

__2.Configure the TCP loopback address of 127.0.0.1. It must show a status of *Active*. Use the Network Status command (NETSTAT) to verify this on every node in the cluster.

__3.The IP addresses used for clustering to a given node must show a status of *Active*. Use NETSTAT to verify this on the status of the subject node.

__4.All cluster IP addresses must be defined with contiguous-bit subnet masks. The subnet address must be the same for each node of the cluster. That is, each IP address participating in the cluster must have the same subnet mask.

__5.The status of the QUSER user profile must be enabled.

__6.TCP Server *INETD must be active on all nodes in the cluster (STRTCPSVR *INETD). Verify this by checking for a presence of a QTOGINTD (user QTCP) job in the Active Jobs list on the subject node. *INETD provides a port in the TCP connection list that "listens" for various clustering functions.

__7.The local and any remote nodes must be able to PING using the IP addresses used for clustering to ensure network routing is active.

__8.Ports 5550 and 5551 are reserved for IBM clustering and must not be used by other applications. Use the NETSTAT command to view the port usage. Port 5550 is opened by clustering and is in a "Listen" state once *INETD is started.

Resilient device requirements

__1.If resilient devices are to be switched between logical partitions on a system, enable Virtual OptiConnect for the partitions. This is done at the Dedicated Service Tools (DST) signon.

__2.If a tower on an HSL loop is switched between two systems, and one of the systems has logical partitions, enable HSL OptiConnect for the partitions. This is done at the Dedicated Service Tools (DST) signon.

__3.When switching resilient devices between logical partitions that are on a system bus, configure the bus as "own bus shared" by one partition. Configure all other partitions that participate in the device switching as "use bus shared".

__4.When switching a tower on a HSL loop between two different systems, configure the tower as switchable.

__5.When a tower is added to an existing HSL loop, all systems on that same loop must be restarted.

__6.Install OS/400 Option 41 (HA Switchable Resources) on all cluster nodes to be in the device domain.

__7.A valid license key must exist on all cluster nodes to be in the device domain. Note that any use of the IBM Simple Cluster Management GUI requires this option.

Security requirements

__1.The Allow Add to Cluster (ALWADDCLU) network attribute must be appropriately set on the target node if trying to start a remote node. Set ALWADDCLU to `*ANY` or `*RQSAUT` depending on the environment.

__2.If ALWADDCLU is set to *RQSAUT, install OS/400 Option 34 (Digital Certificate Manager) and a Cryptographic Access Provided Product (5722-AC2 or AC3).

__3.The user profile invoking the Cluster Resource Services APIs must exist on all cluster nodes and must have IOSYSCFG authority.

__4.The user profile to run the exit program for a CRG must exist on all recovery domain nodes.

Job considerations

__1.Jobs can be submitted by the Cluster Resource Services APIs to process requests. The jobs either run under the user profile to run the exit program specified when creating the CRG, or under the user profile that requested the API (for varying on devices in a Device Resiliency CRGs only). The user must ensure that the subsystem which services the job queue associated with the user profile is configured as *NOMAX for the number of jobs that can be run from that job queue.

__2.Jobs are submitted to the job queue specified by the job description that is obtained from the user profile defined for a CRG. The default job description causes the jobs to be sent to the QBATCH job queue. Since this job queue is used for many user jobs, the exit program job may not run in a timely fashion. Consider using a unique job description with a unique user queue. The number of maximum programs should be set to a value greater than one.

__3.The value of the routing data supplied in the job description determines in which main storage pool, and with what run time attributes, the exit program executes. The default job description values result in jobs that execute in a pool with other batch jobs with a run priority of 50. The default options may not produce the desired performance for exit programs of clustering jobs. The subsystem initiating the exit program jobs (which is the same subsystem that is using the unique job queue) should assign the exit program jobs to a pool that is not used by other jobs initiated by the same subsystem or other subsystems. In addition, assign a run priority of 15 so that the exit program jobs run before most user jobs.

IBM Simple Cluster Management GUI considerations

__1.Install OS/400 Option 41 (HA Switchable Resources) on all cluster nodes to be in the device domain.

__2.A valid license key must exist for each cluster node in the device domain.

__3.Start all host servers. Use this command:

```
STRHOSTSVR SERVER(*ALL)
```

__4.Start the Management Central server (*MGTC). Use this command:

```
STRTCPSVR SERVER(*MGTC)
```

Follow these steps to work with the clusters function in Management Central:

1. Open the main Operations Navigator window by clicking **Start-> Programs-> IBM AS/400 Client Access Express-> AS/400 Operations Navigator**.
2. Expand **Management Central**.
3. Select **Clusters**.

5.2.2 Using the Simple Cluster Management GUI to create the cluster

The IBM Simple Cluster Management GUI is provided as part of Operations Navigator. Use it to create a simple two-node cluster.

Considerations when using the IBM Simple Cluster Management GUI include:

- The Operations Navigator GUI appears as a special Management Central system group against which cluster functions can be performed.
- It provides system-by-system functions.
- It supports a maximum of two nodes in the cluster.
- Only application and device CRGs are supported with Operations Navigator GUI. Data CRGs are not supported.

Follow these steps to create a cluster using the Simple Cluster Management GUI:

1. Sign on to Management Central on the system that is to be the managing system.

   > **Note:** The managing system does not have to be one of the systems that participates in the cluster.

   The Operations Navigator initial window is displayed in Figure 5-2.

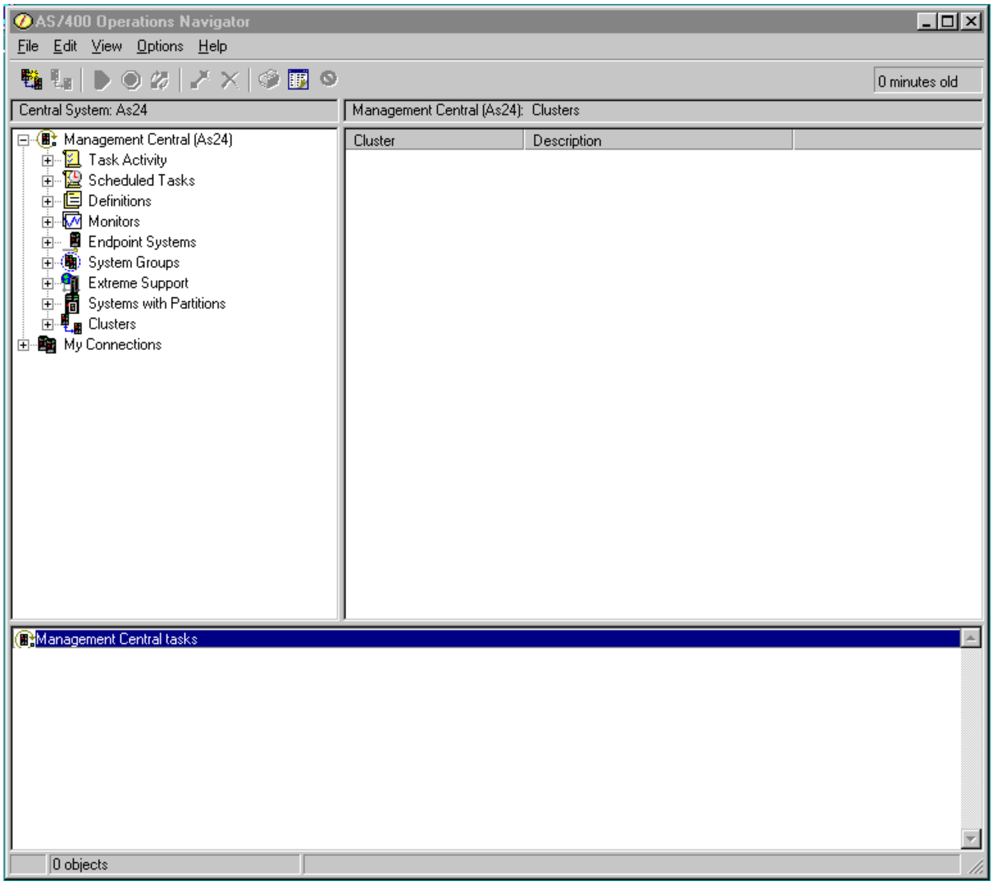


Figure 5-2 Operations Navigator initial window

2. Right-click **Clusters** and select **New Cluster** from the drop-down menu. This is the starting point for creating a cluster. See Figure 5-3.

   > **Note:** You may experience a short delay as the Wizard is loaded from the iSeries server.

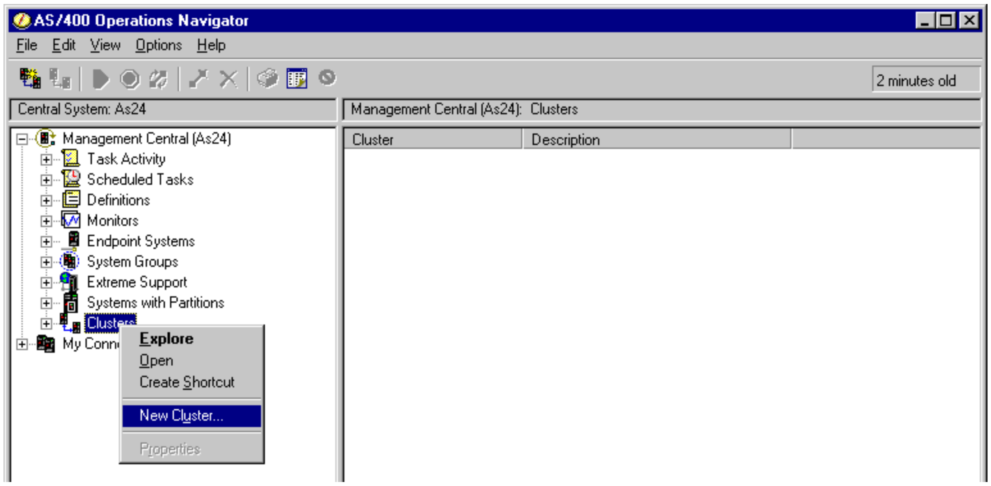
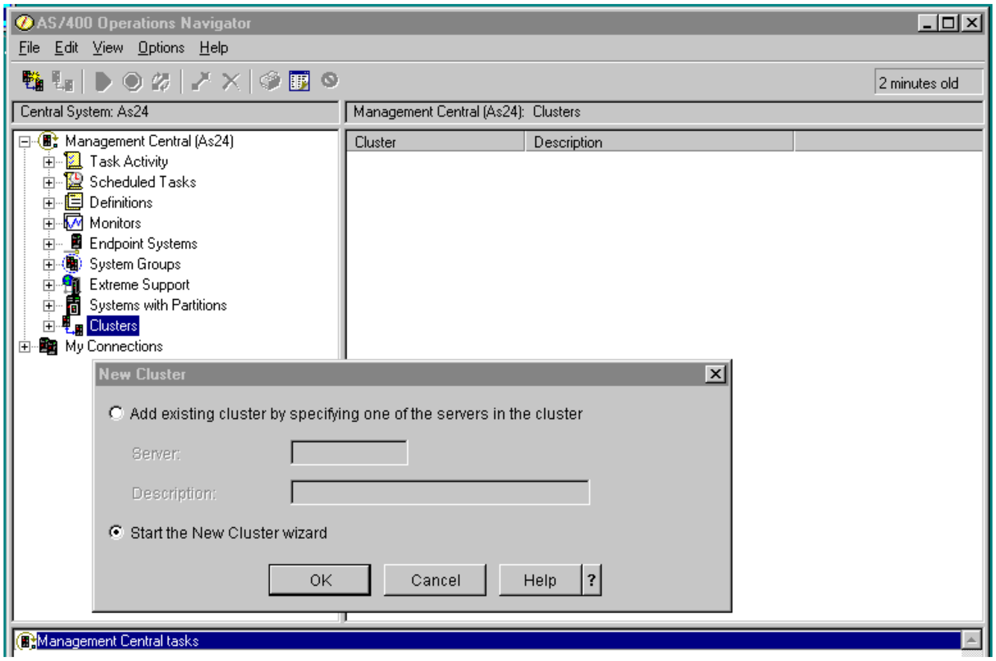


Figure 5-3 Creating a new cluster

3. On the New Cluster window, select **Start the New Cluster Wizard** and then click **OK** as shown in Figure 5-4.



○ Add existing cluster by specifying one of the servers in the cluster

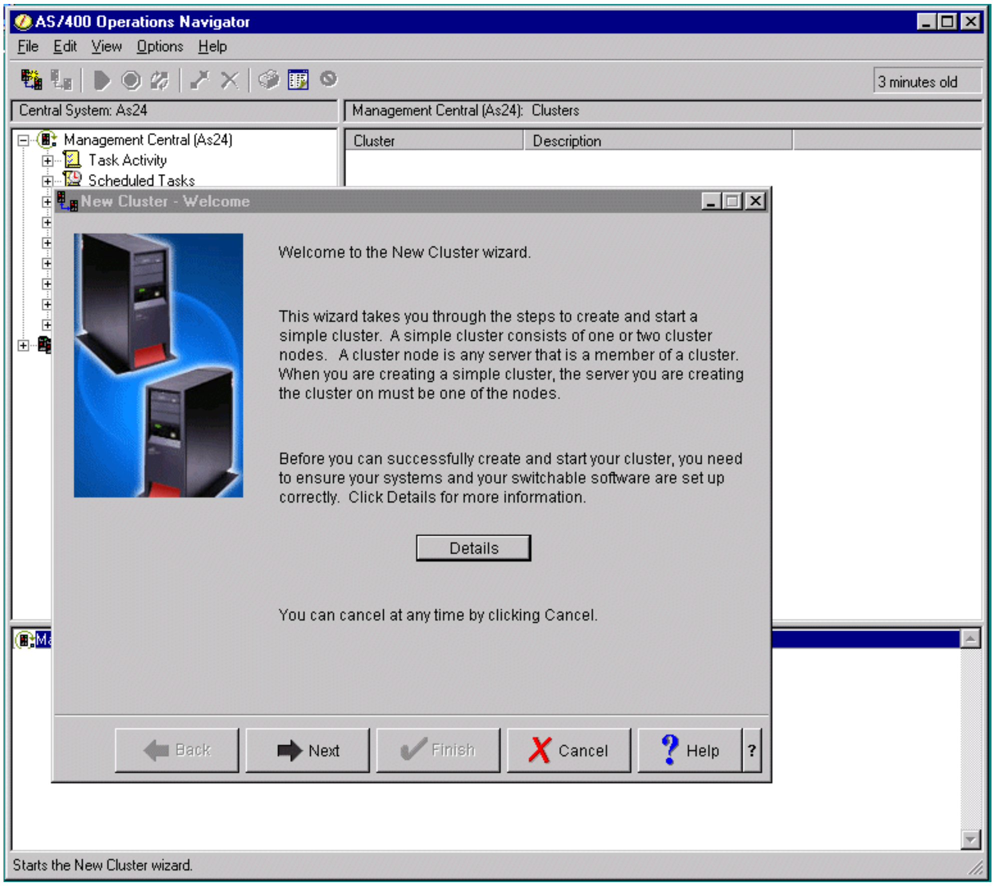
Server:

Description:

● Start the New Cluster wizard

Figure 5-4 Starting the New Cluster wizard

4. The new cluster dialogue box (Figure 5-5) reminds you that there are requirements for the cluster to be created. Click **Details** to see the prerequisites.


AS/400 Operations Navigator
File Edit View Options Help
3 minutes old
Central System: As24
Management Central (As24): Clusters
Management Central (As24)
Task Activity
Scheduled Tasks
Cluster
Description
New Cluster - Welcome
Details
Back
Next
Finish
Cancel
Help
?
Starts the New Cluster wizard.

Welcome to the New Cluster wizard.

This wizard takes you through the steps to create and start a simple cluster. A simple cluster consists of one or two cluster nodes. A cluster node is any server that is a member of a cluster. When you are creating a simple cluster, the server you are creating the cluster on must be one of the nodes.

Before you can successfully create and start your cluster, you need to ensure your systems and your switchable software are set up correctly. Click Details for more information.

You can cancel at any time by clicking Cancel.

Figure 5-5 New Cluster Wizard welcome window

As illustrated in Figure 5-6, hardware and software prerequisites are listed. Do not proceed until these prerequisites are met:

– **Hardware**: It is necessary to have either two V5R1 systems, or a logically partitioned system with a minimum of two partitions, where each partition is at V5R1.

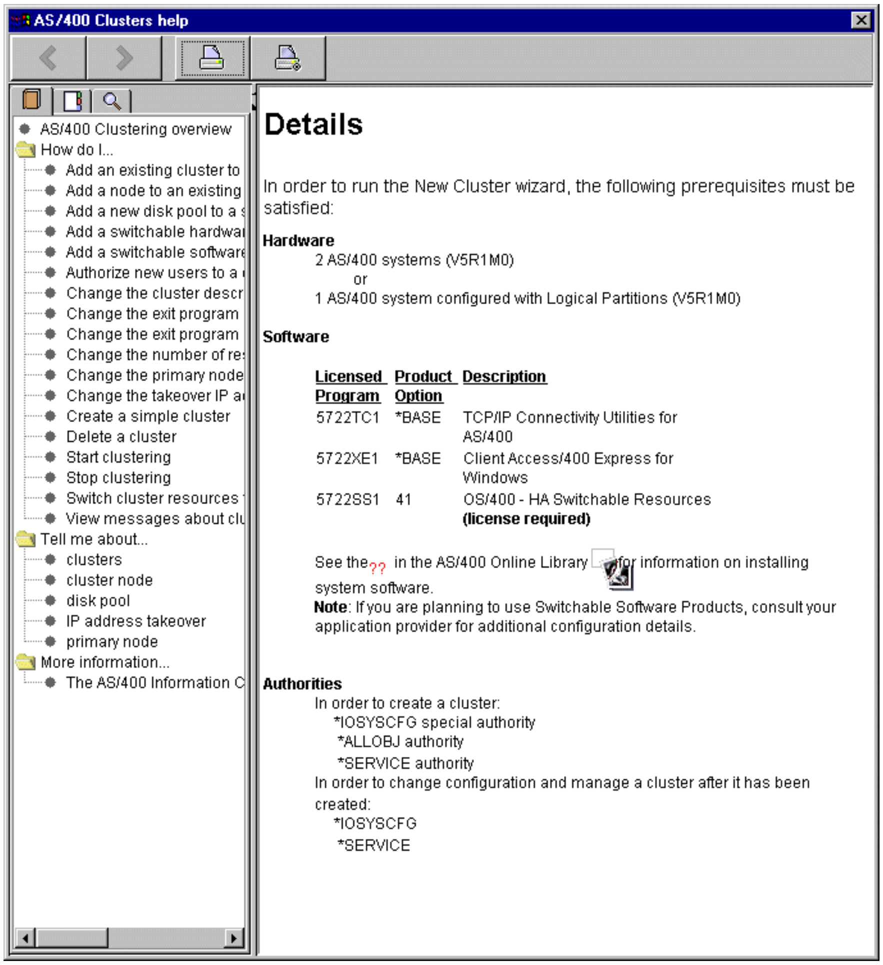
  If you plan to create a Switchable IASP in a standalone disk tower, install an HSL adapter card.

– **Software:** The TCP/IP Connectivity Utilities product (5722-TC1) is required, as is Client Access Express (5722-XE1) and OS/400 HA Switchable Resources (Option 41 of 5722-SS1).

  HSL OptiConnect (Option 23 of 5722-SS1) is not required for clustering or IASPs. It is required for system-to-system communication, for example, Distributed Data Management (DDM).

Prerequisites are identified in a Help panel of Operations Navigator clustering, as illustrated in Figure 5-6.

For a further discussion on prerequisites for clustering, see 4.5, “Prerequisites for iSeries clustering” on page 37.



Details

In order to run the New Cluster wizard, the following prerequisites must be satisfied:

Hardware

2 AS/400 systems (V5R1M0)
or
1 AS/400 system configured with Logical Partitions (V5R1M0)

Software

Licensed Program	**Product Option**	**Description**
5722TC1	*BASE	TCP/IP Connectivity Utilities for AS/400
5722XE1	*BASE	Client Access/400 Express for Windows
5722SS1	41	OS/400 - HA Switchable Resources **(license required)**

See the ?? in the AS/400 Online Library for information on installing system software.

Note: If you are planning to use Switchable Software Products, consult your application provider for additional configuration details.

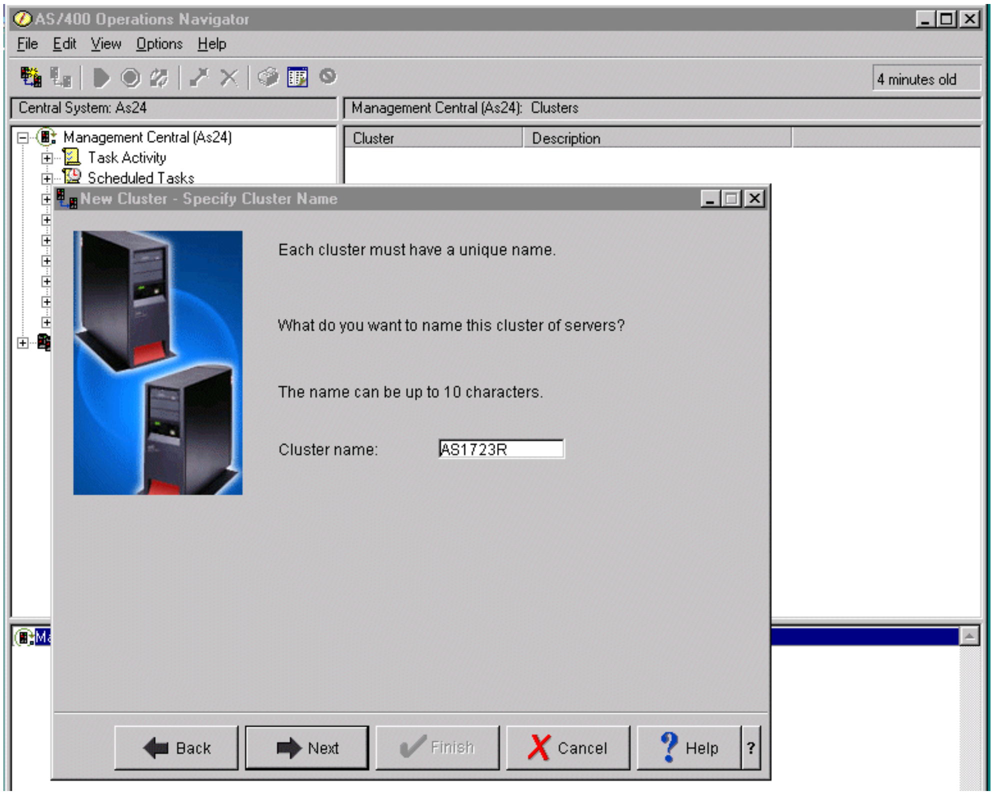
Authorities

In order to create a cluster:
*IOSYSCFG special authority
*ALLOBJ authority
*SERVICE authority

In order to change configuration and manage a cluster after it has been created:
*IOSYSCFG
*SERVICE

Figure 5-6 Cluster prerequisites

5. As shown in Figure 5-7, enter the name of the cluster. Choose a descriptive name; it should be no more than ten characters in length.


AS/400 Operations Navigator
File Edit View Options Help
4 minutes old
Central System: As24
Management Central (As24): Clusters
Management Central (As24)
Task Activity
Scheduled Tasks
Cluster
Description
New Cluster - Specify Cluster Name
Back
Next
Finish
Cancel
Help
?

Each cluster must have a unique name.

What do you want to name this cluster of servers?

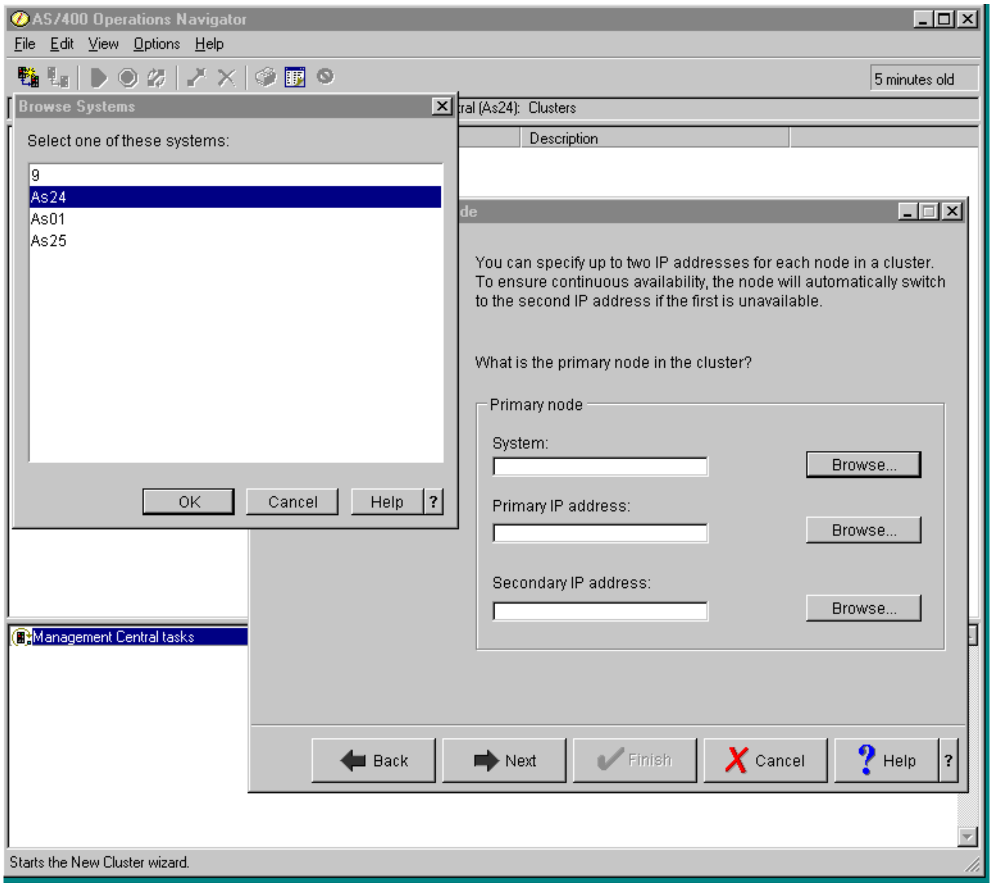
The name can be up to 10 characters.

Cluster name: AS1723R

Figure 5-7 Naming the cluster

6. Specify which system is to be the primary node in the cluster.

   Use the **Browse** button to select a system. The server selected does not have to appear in the select list to be a valid node in the cluster. See Figure 5-8.



AS/400 Operations Navigator
File Edit View Options Help
5 minutes old
tral (As24): Clusters
Description
Management Central tasks
Starts the New Cluster wizard.

Browse Systems

Select one of these systems:

- 9
- As24
- As01
- As25

You can specify up to two IP addresses for each node in a cluster. To ensure continuous availability, the node will automatically switch to the second IP address if the first is unavailable.

What is the primary node in the cluster?

Primary node

System:

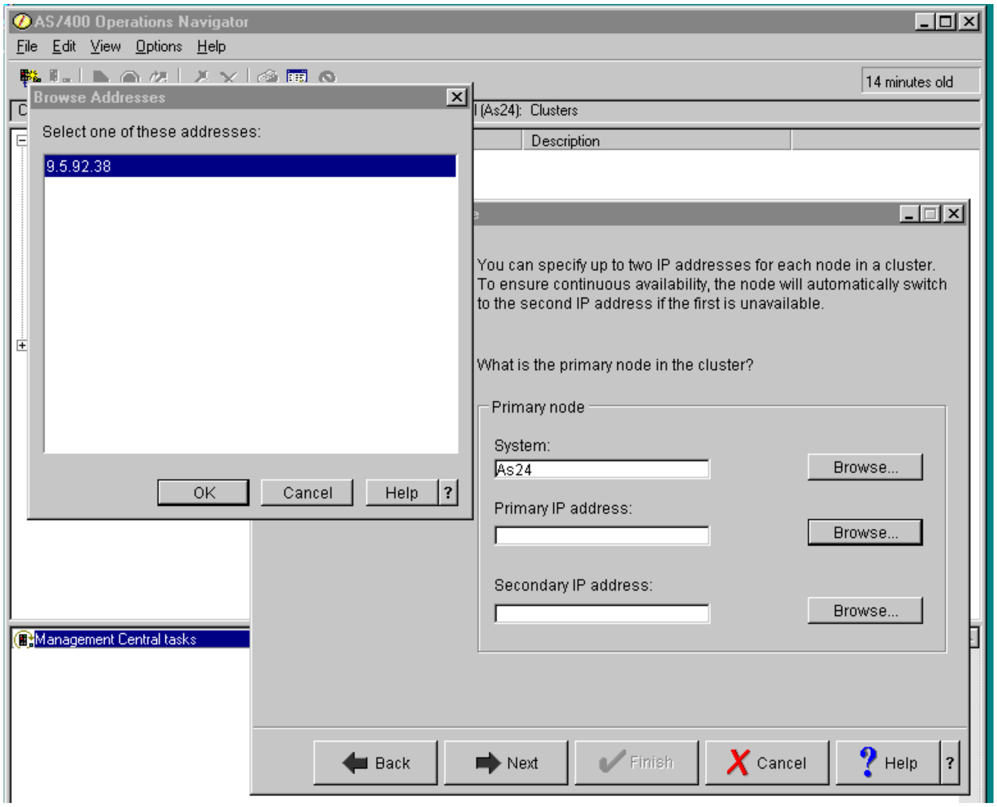
Primary IP address:

Secondary IP address:

Figure 5-8 Specifying a system to be the primary node

7. Specify the IP address of the Primary Node. Enter the IP address directly or select it from the drop-down dialogue box. See Figure 5-9.

> **Hint**: If the IP address does not appear in the dialogue box, there may be a problem with the Host table. When the system name is entered, its address is retrieved from the Domain Name Server (DNS). Enter the IP address as the system name to bypass this check.



Select one of these addresses:

9.5.92.38

You can specify up to two IP addresses for each node in a cluster. To ensure continuous availability, the node will automatically switch to the second IP address if the first is unavailable.

What is the primary node in the cluster?

Primary node

System: As24

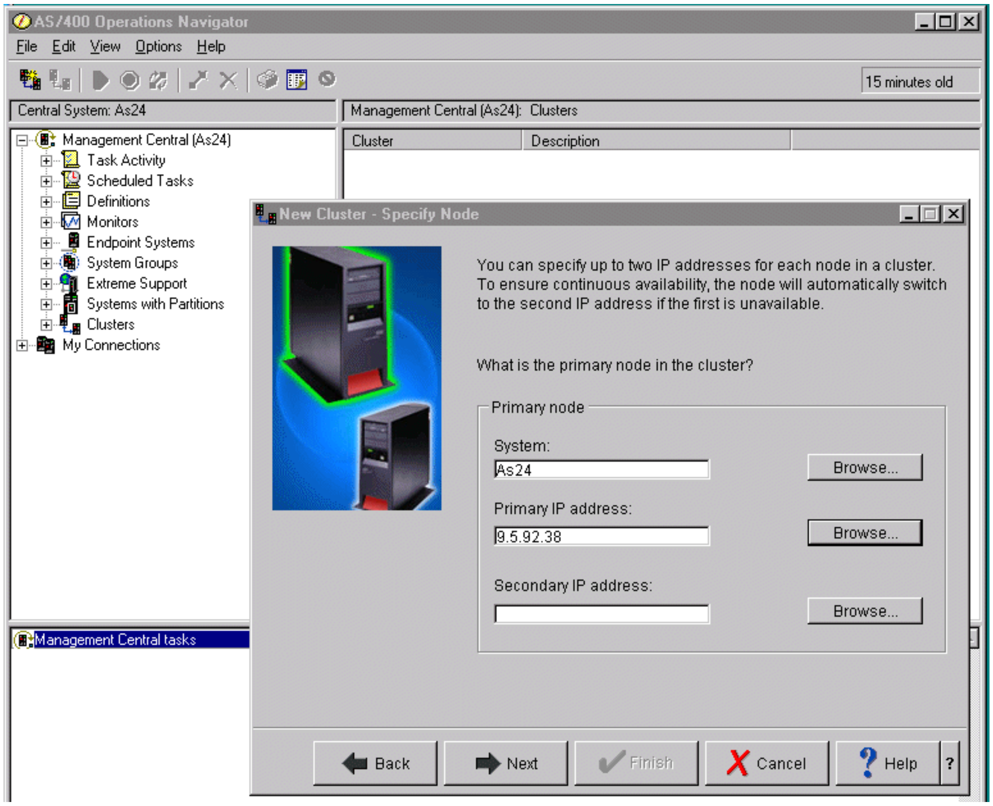
Primary IP address:

Secondary IP address:

Figure 5-9 Specifying the IP address of primary node

Figure 5-10 shows the completed display.

Note: A secondary IP address can be specified for redundancy. This allows a secondary IP path to the system in the event that the primary interface is unavailable.



You can specify up to two IP addresses for each node in a cluster. To ensure continuous availability, the node will automatically switch to the second IP address if the first is unavailable.

What is the primary node in the cluster?

Primary node

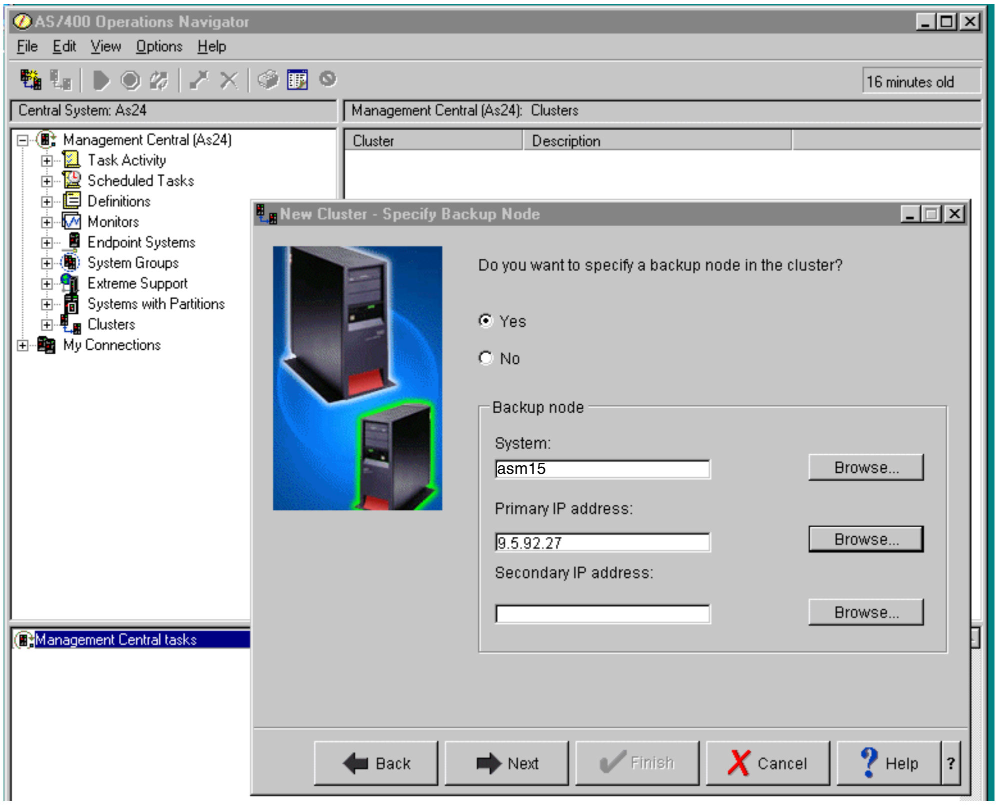
System: As24

Primary IP address: 9.5.92.38

Secondary IP address:

Figure 5-10 Cluster primary node completion display

8. The next window (Figure 5-11) allows you to select the Backup Node and IP address in the same way that the primary node is selected.



Do you want to specify a backup node in the cluster?

● Yes

○ No

Backup node

System: asm15

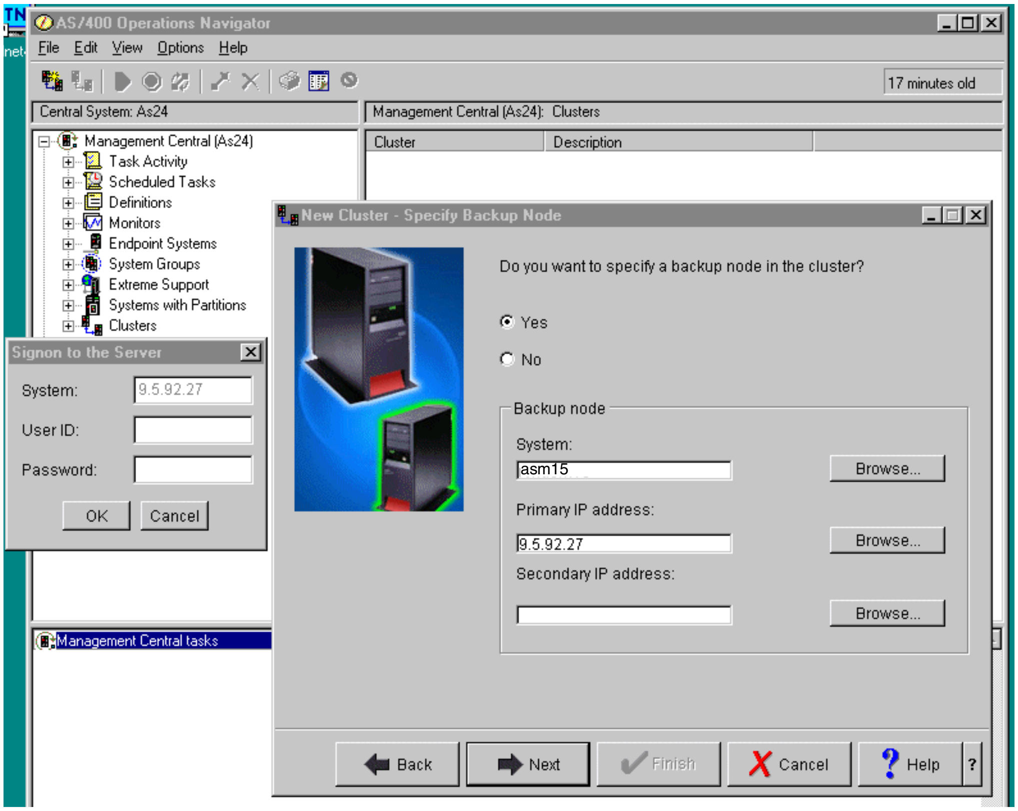
Primary IP address: 9.5.92.27

Secondary IP address:

Figure 5-11 Specifying the name and IP address of backup node

9. You are then prompted to sign on to the Backup Node. See Figure 5-12.

Important: The user profile entered must exist on both nodes in the cluster.



Signon to the Server

System: 9.5.92.27

User ID:

Password:

New Cluster - Specify Backup Node

Do you want to specify a backup node in the cluster?

● Yes

○ No

Backup node

System: asm15

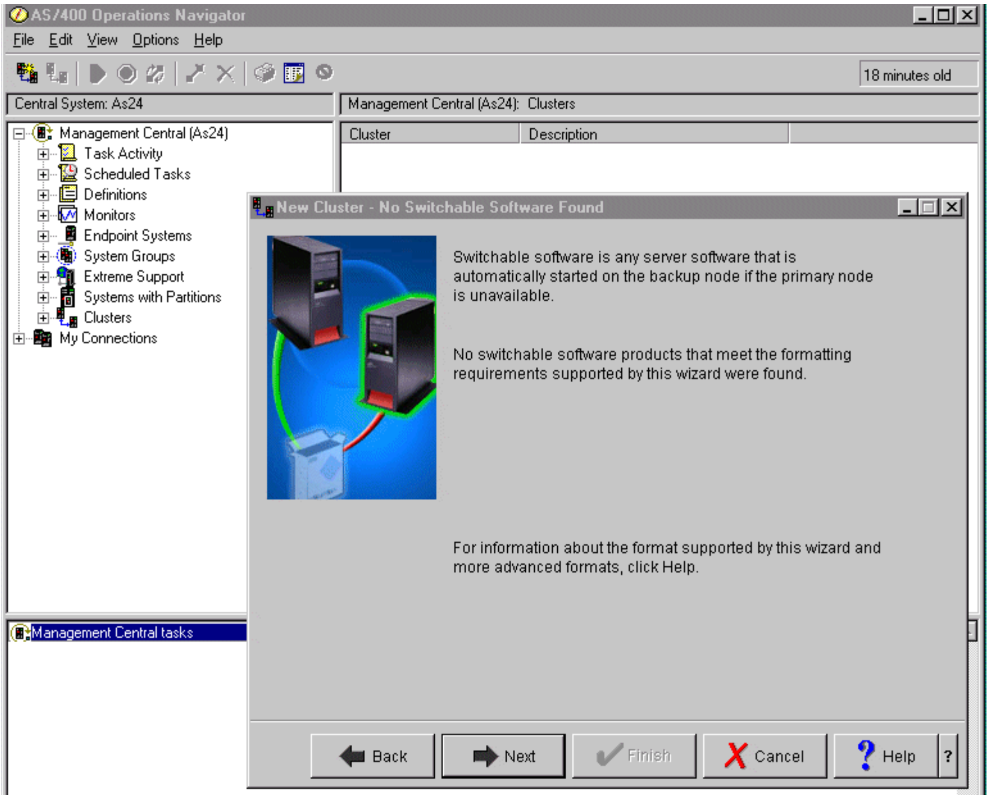
Primary IP address: 9.5.92.27

Secondary IP address:

Figure 5-12 Signing on to the backup node

10. After a search of the systems, you are informed whether “Switchable” software is installed as shown in Figure 5-13. Adding Switchable Software is described starting in Step 16 of this procedure.

> **Note:** Switchable software is any server software that is automatically started on the backup node if the primary node is unavailable. An example of switchable software is Lotus Domino for iSeries, which is ClusterProven and therefore switchable. See Chapter 8, “ClusterProven for iSeries applications” on page 161, for a further description of ClusterProven.



Switchable software is any server software that is automatically started on the backup node if the primary node is unavailable.

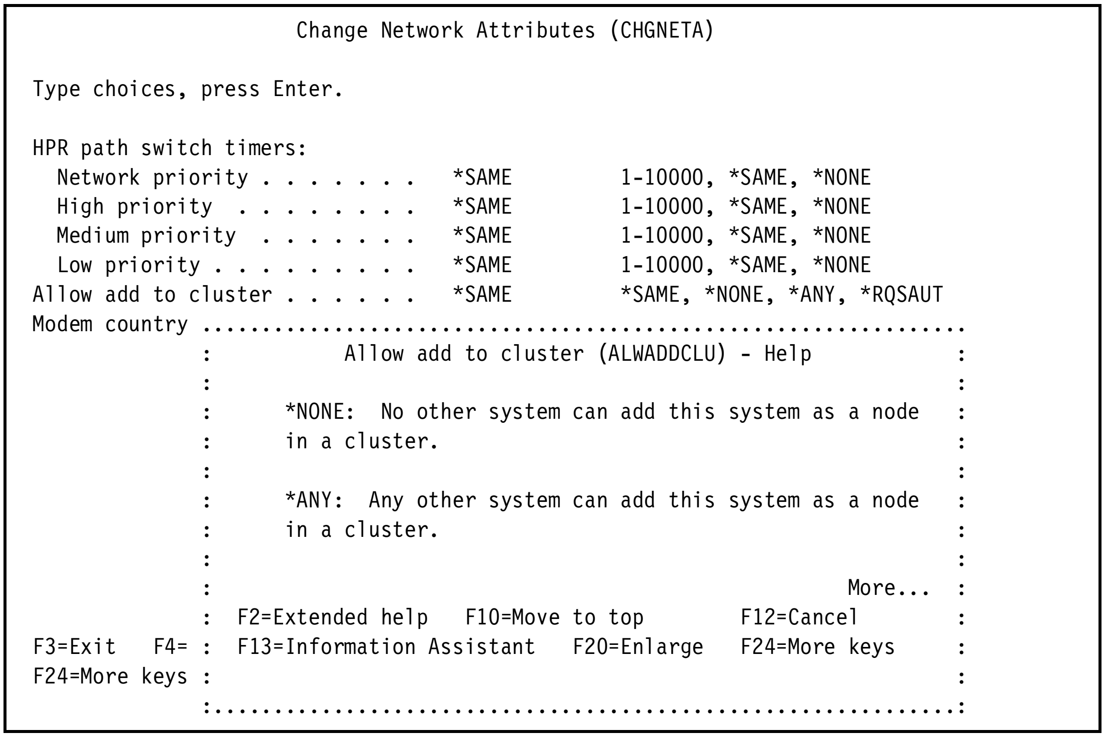
No switchable software products that meet the formatting requirements supported by this wizard were found.

For information about the format supported by this wizard and more advanced formats, click Help.

Figure 5-13 Switchable software

An error message may appear that indicates that there is a problem with the network attributes as shown in Figure 5-14.

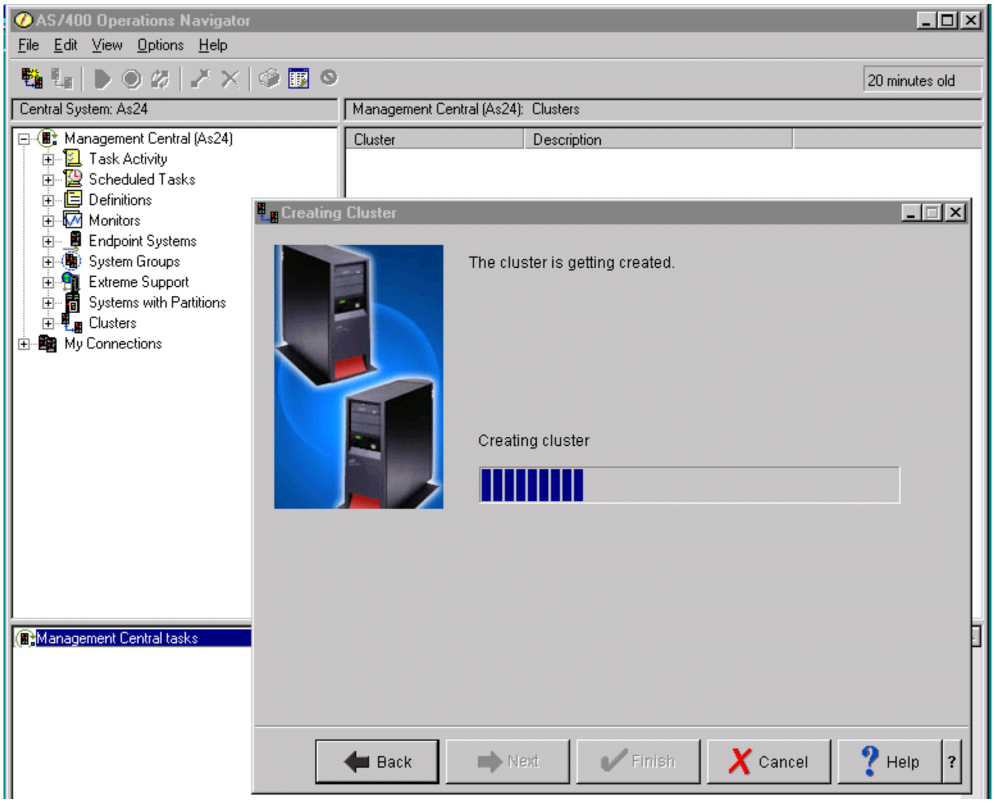
> **Note:** Both systems need to have the Allow Add to Cluster (ALWADDCLU) network attribute set to *ANY or *RQSAUT.



```
                     Change Network Attributes (CHGNETA)

Type choices, press Enter.

HPR path switch timers:
  Network priority . . . . . . .   *SAME         1-10000, *SAME, *NONE
  High priority  . . . . . . . .   *SAME         1-10000, *SAME, *NONE
  Medium priority  . . . . . . .   *SAME         1-10000, *SAME, *NONE
  Low priority . . . . . . . . .   *SAME         1-10000, *SAME, *NONE
Allow add to cluster . . . . . .   *SAME         *SAME, *NONE, *ANY, *RQSAUT
Modem country ..............................................................
              :           Allow add to cluster (ALWADDCLU) - Help           :
              :                                                             :
              :      *NONE:  No other system can add this system as a node  :
              :      in a cluster.                                          :
              :                                                             :
              :      *ANY:  Any other system can add this system as a node  :
              :      in a cluster.                                          :
              :                                                             :
              :                                                   More...  :
              :  F2=Extended help   F10=Move to top       F12=Cancel        :
F3=Exit   F4= :  F13=Information Assistant   F20=Enlarge   F24=More keys     :
F24=More keys :                                                             :
              :.............................................................:
```

Figure 5-14 Network attribute to allow clustering

11. The cluster is created. The status bar appears on the Creating Cluster display (Figure 5-15) to indicate the progress.

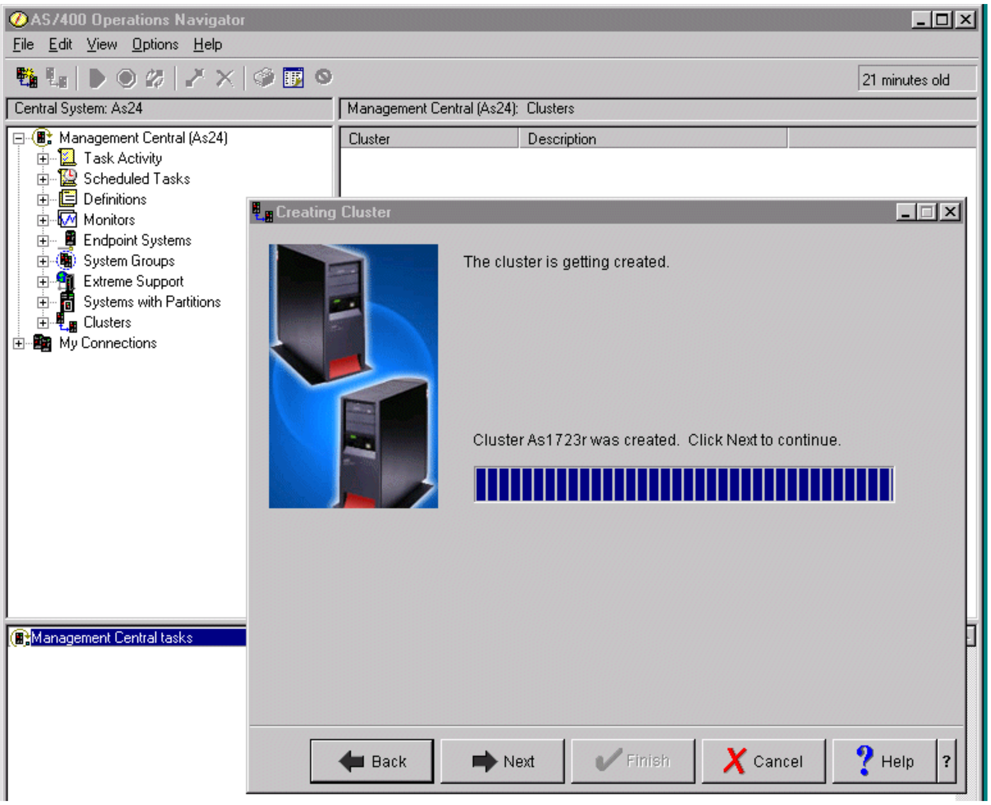


The cluster is getting created.

Creating cluster

Figure 5-15 Creating a cluster

12. When the cluster is created successfully, a completion message is displayed. See Figure 5-16.

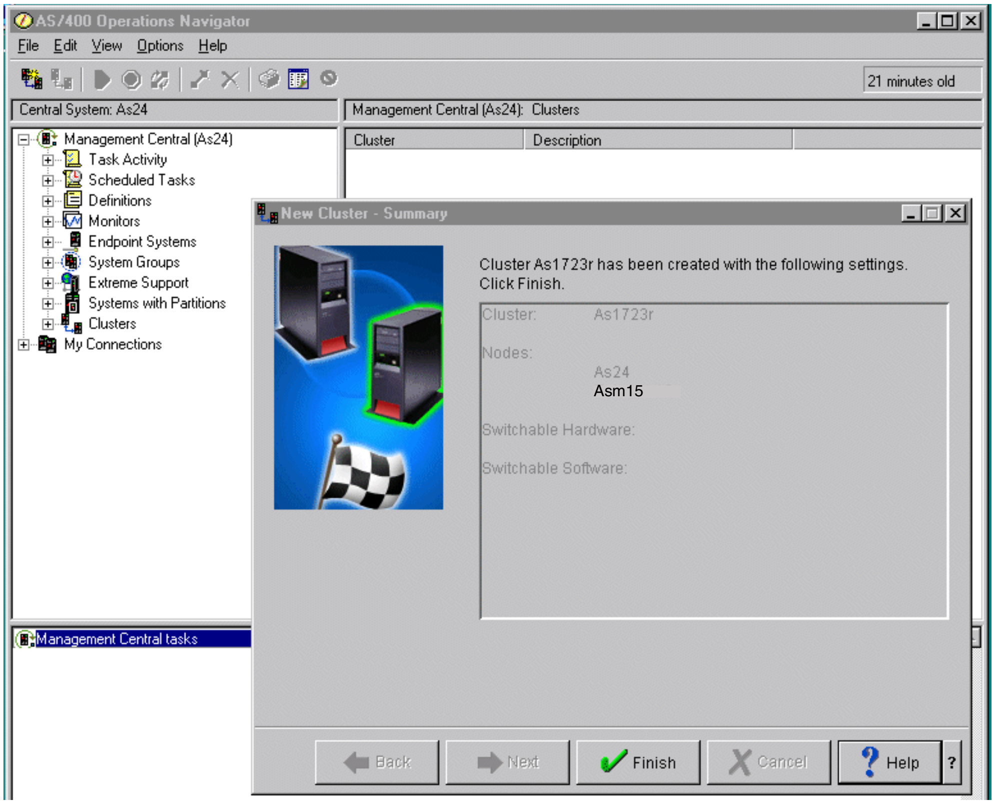


The cluster is getting created.

Cluster As1723r was created. Click Next to continue.

Figure 5-16 Cluster creation complete

13. The New Cluster - Summary display (Figure 5-17) shows a summary of the created cluster. The nodes in the cluster are shown, as well as any switchable hardware or software that may be present.



Cluster As1723r has been created with the following settings. Click Finish.

Cluster: As1723r

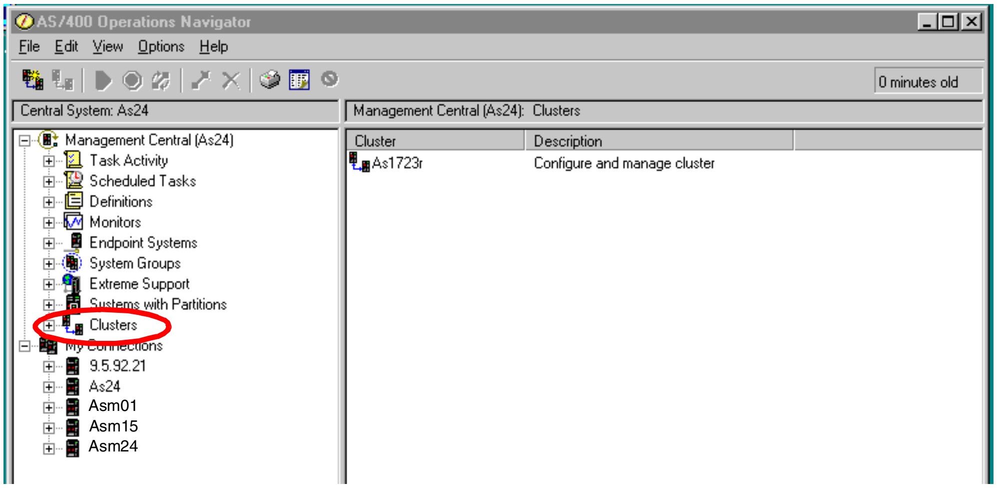
Nodes:
As24
Asm15

Switchable Hardware:

Switchable Software:

Figure 5-17 The cluster is created

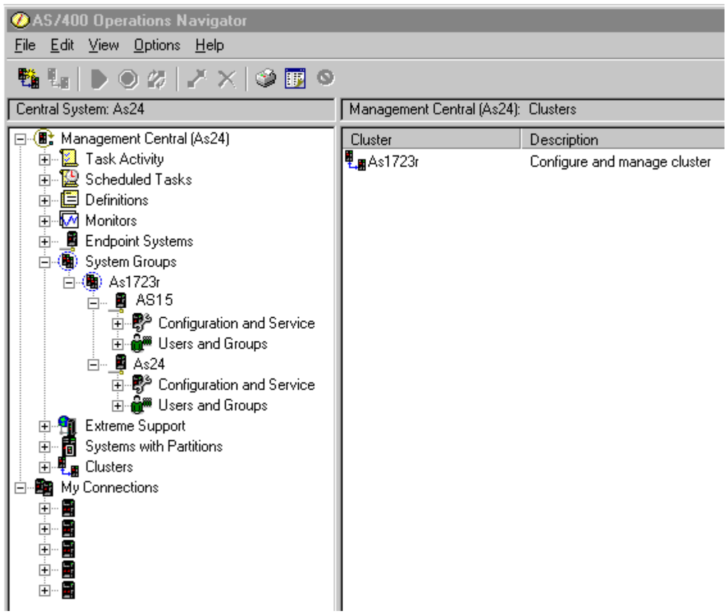
14. Refresh the Operations Navigator screen (F5). You now see the new cluster under the Cluster in Management Central as shown in Figure 5-18.



Cluster	Description
As1723r	Configure and manage cluster

Figure 5-18 New cluster appears in Management Central

Note that two system groups are created by the process – one for each node in the cluster. See Figure 5-19.

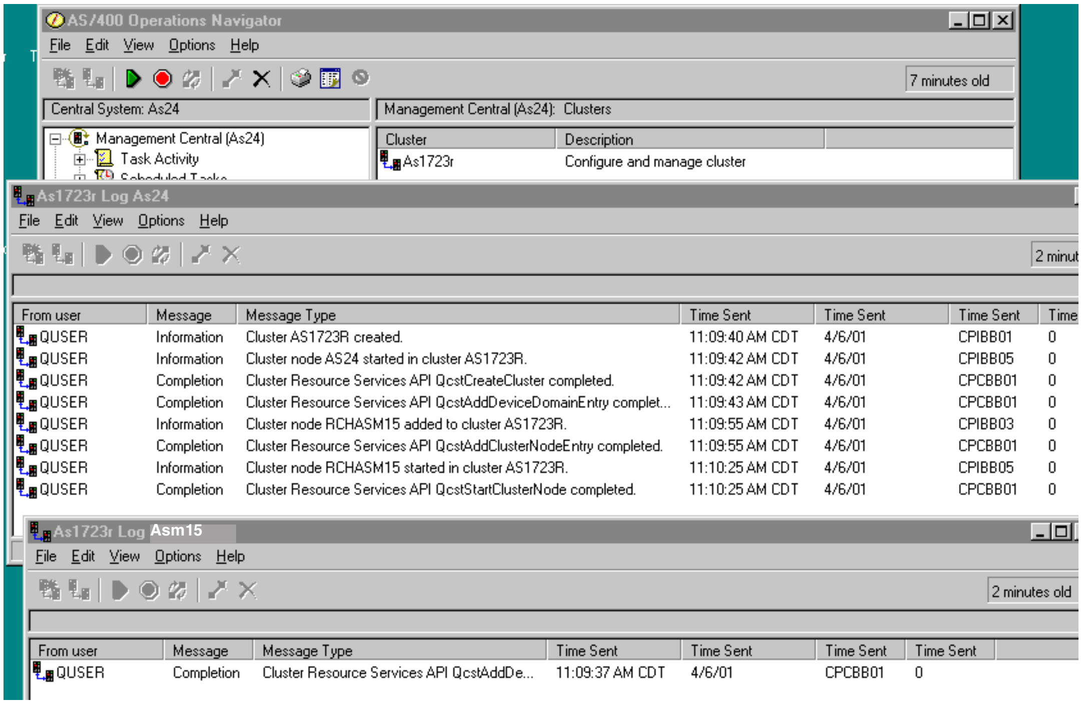


Cluster	Description
As1723r	Configure and manage cluster

Figure 5-19 System groups created

15. Right-click the cluster name and select **Cluster Log**. This allows you to look at the cluster creation log, which is shown in Figure 5-20.

> **Note:** The GUI calls the cluster APIs. The following actions transpire with the GUI function:
>
> - The QcstCreateCluster API is called, which is equivalent to the CRTCLU command.
> - The ADDDEVDMNE command calls the QcstAddDeviceDomainEntry API.
> - The ADDCLUNODE command calls the QcstAddClusterNodeEntry API.
> - The QcstStartClusterNode API starts the cluster node. This is equivalent to using the STRCLUNOD command.


AS/400 Operations Navigator
File Edit View Options Help
7 minutes old
Central System: As24
Management Central (As24): Clusters
Management Central (As24)
Task Activity
Scheduled Tasks
As1723r Log As24
File Edit View Options Help
2 minut
As1723r Log Asm15
File Edit View Options Help
2 minutes old

Cluster	Description
As1723r	Configure and manage cluster

From user	Message	Message Type	Time Sent	Time Sent	Time Sent	Time
QUSER	Information	Cluster AS1723R created.	11:09:40 AM CDT	4/6/01	CPIBB01	0
QUSER	Information	Cluster node AS24 started in cluster AS1723R.	11:09:42 AM CDT	4/6/01	CPIBB05	0
QUSER	Completion	Cluster Resource Services API QcstCreateCluster completed.	11:09:42 AM CDT	4/6/01	CPCBB01	0
QUSER	Completion	Cluster Resource Services API QcstAddDeviceDomainEntry complet...	11:09:43 AM CDT	4/6/01	CPCBB01	0
QUSER	Information	Cluster node RCHASM15 added to cluster AS1723R.	11:09:55 AM CDT	4/6/01	CPIBB03	0
QUSER	Completion	Cluster Resource Services API QcstAddClusterNodeEntry completed.	11:09:55 AM CDT	4/6/01	CPCBB01	0
QUSER	Information	Cluster node RCHASM15 started in cluster AS1723R.	11:10:25 AM CDT	4/6/01	CPIBB05	0
QUSER	Completion	Cluster Resource Services API QcstStartClusterNode completed.	11:10:25 AM CDT	4/6/01	CPCBB01	0

From user	Message	Message Type	Time Sent	Time Sent	Time Sent	Time Sent
QUSER	Completion	Cluster Resource Services API QcstAddDe...	11:09:37 AM CDT	4/6/01	CPCBB01	0

Figure 5-20 The cluster creation log

You can make other changes to the cluster with this drop-down menu, such as to Collection Services, Inventory, or Fixes. See Figure 5-21.

Note: The Cluster starts by default. Use Operations Navigator to stop or start the cluster.

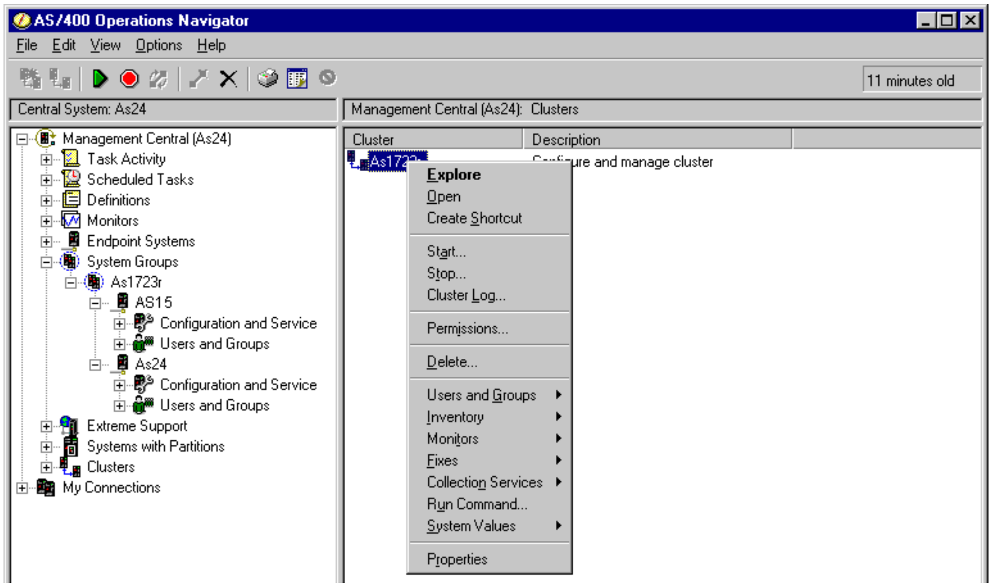


Figure 5-21 Managing a cluster with Operations Navigator

16. The cluster is created. To add Switchable Software to the cluster, click the **+** (plus sign) to expand **Clusters**. Click **Switchable Software**. Then right-click and select **Add Product**. See Figure 5-22.

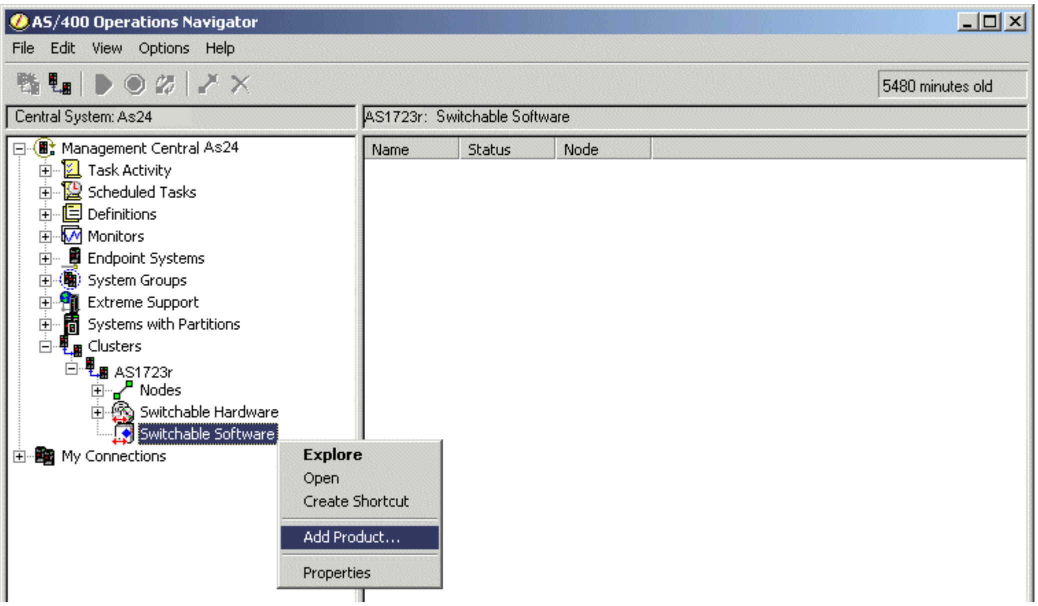
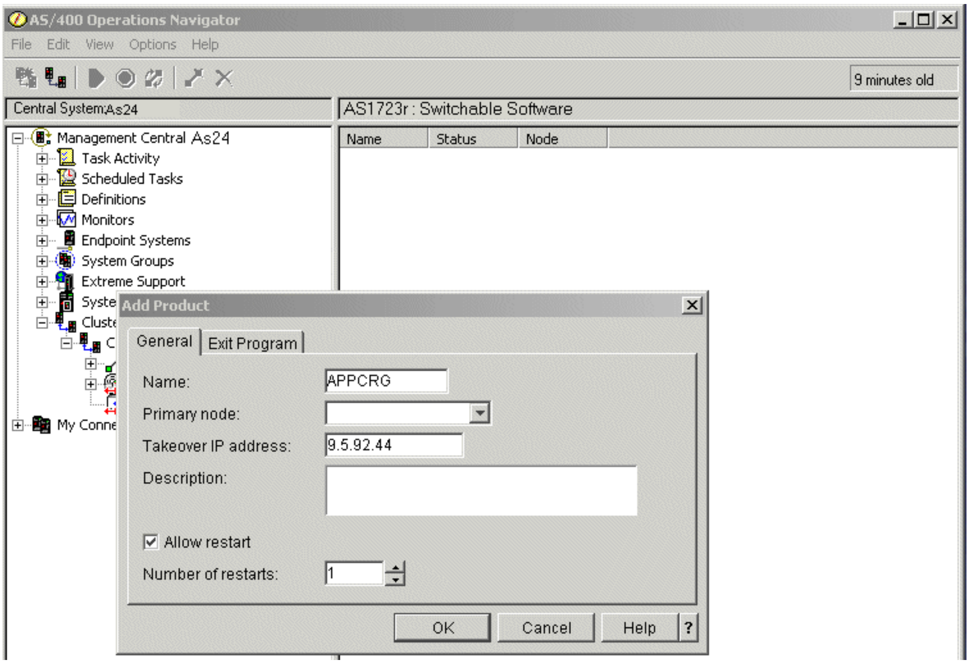


Figure 5-22 Adding switchable software

17. On the Add Product window (Figure 5-23), enter the cluster name, primary node, and the takeover IP address for the application.



Name: APPCRG

Primary node:

Takeover IP address: 9.5.92.44

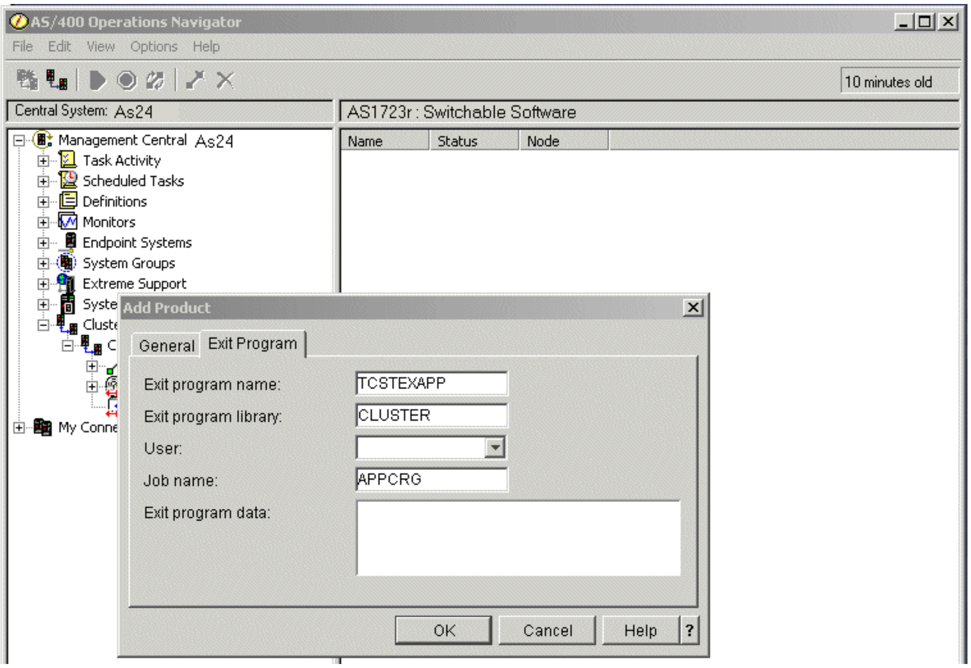
Description:

☑ Allow restart

Number of restarts: 1

Figure 5-23 Adding the application CRG

18. Click the **Exit Program** tab (Figure 5-24). Enter the exit program name and library, the user under which the job will run, and the name of the job. Then, click **OK**.



Exit program name: TCSTEXAPP

Exit program library: CLUSTER

User:

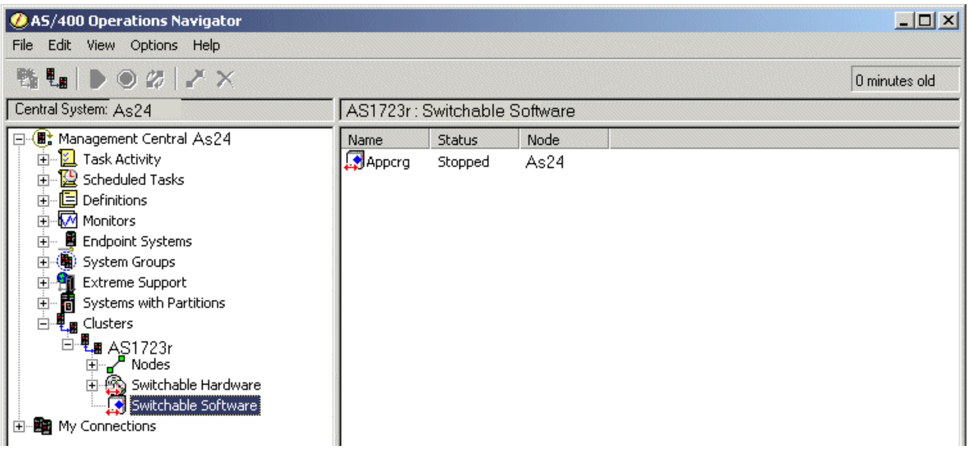
Job name: APPCRG

Exit program data:

Figure 5-24 Entering the name of the exit program

Note: The exit program used here is the example shipped in QUSRTOOL.

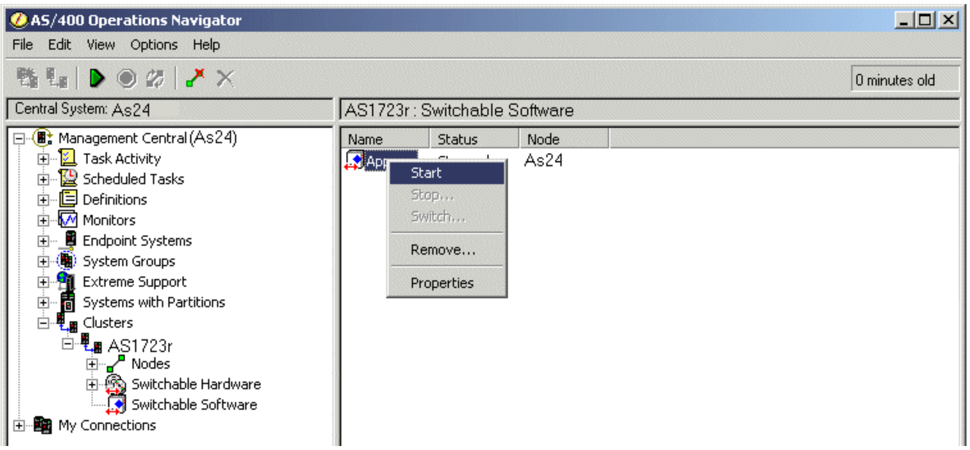
19. Refresh the Operations Navigator window (F5). The new software appears under Switchable Software a shown in Figure 5-25.



Name	Status	Node
Appcrg	Stopped	As24

Figure 5-25 *Application added*

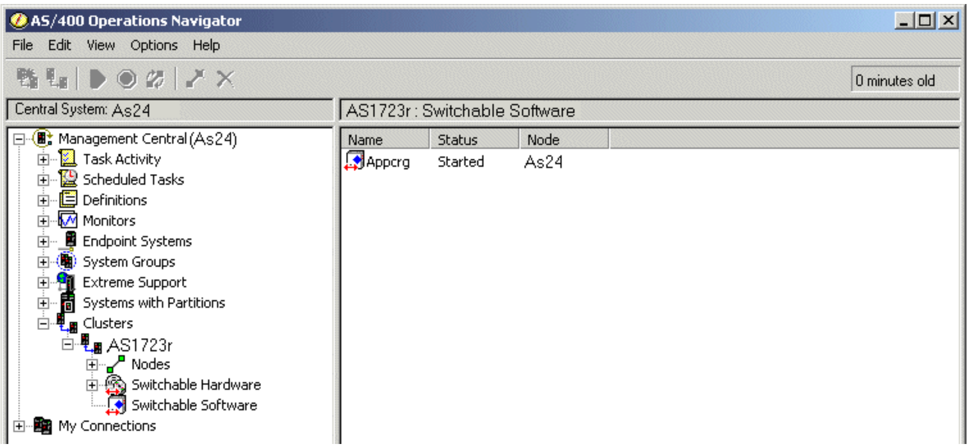
20. The status of the Software is *Stopped*. Right-click the software and select **Start** (Figure 5-26).



Name	Status	Node
App		As24

Figure 5-26 *Starting the application*

21. Refresh the Operations Navigator window (F5). The status changes to *Started* (Figure 5-27).



AS1723r : Switchable Software

Name	Status	Node
Appcrg	Started	As24

Figure 5-27 CRG started

22. To switch the software to the backup system, right-click the software and select **Switch** (Figure 5-28).

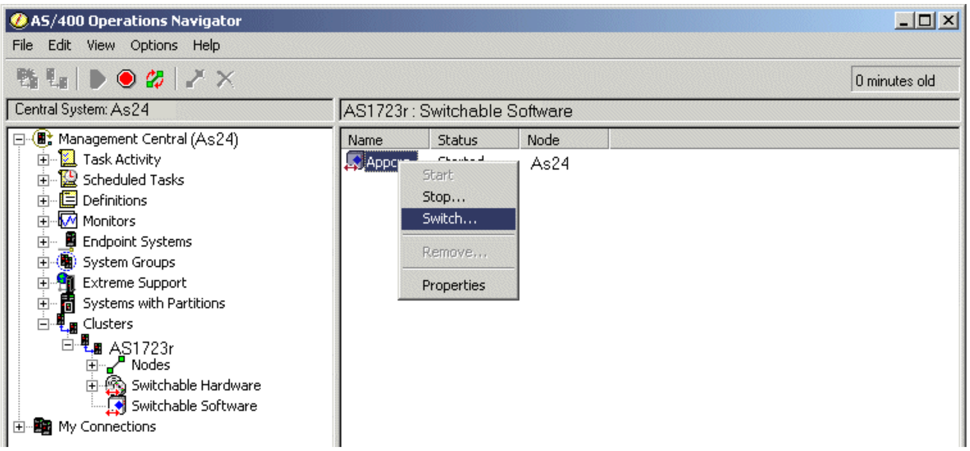
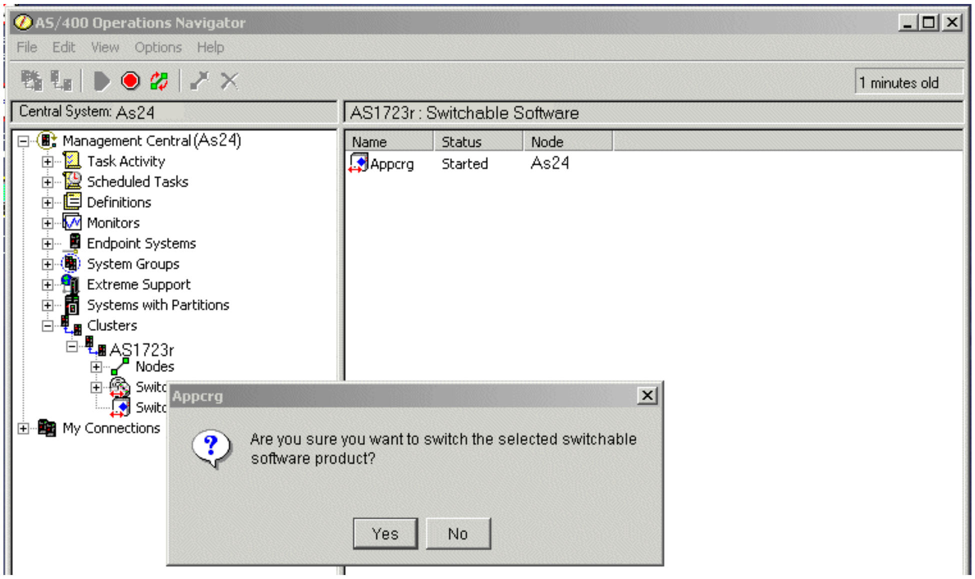


Figure 5-28 Switching the application to a backup system

23. Click **Yes** to confirm the switch (Figure 5-29).



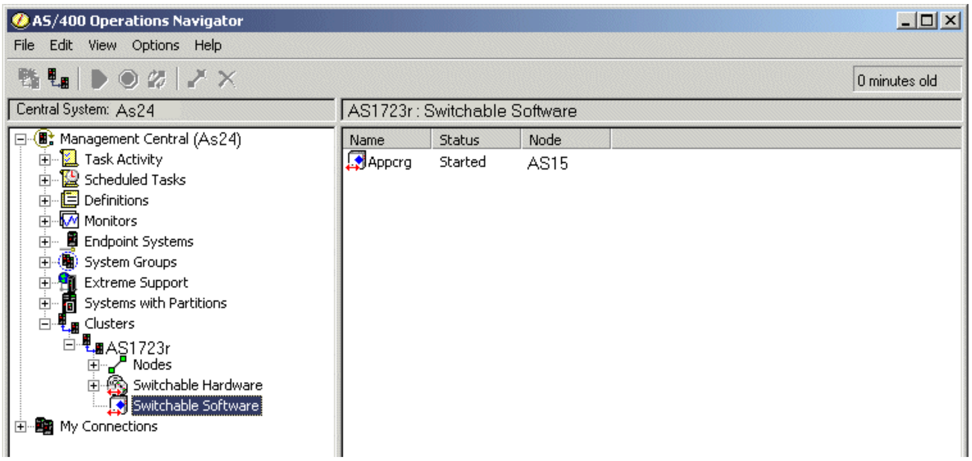
AS1723r : Switchable Software

Name	Status	Node
Appcrg	Started	As24

Are you sure you want to switch the selected switchable software product?

Figure 5-29 Confirming the switch of the application

24. Refresh the Operations Navigator window (F5). The active node has changed to the backup system (Figure 5-30).



AS1723r : Switchable Software

Name	Status	Node
Appcrg	Started	AS15

Figure 5-30 Node changed to backup

Common problems

As with operations of any type, problems can be encountered when using the Operations Navigator GUI to create a simple cluster. Some of the typical errors encountered are listed in this section. Use the command that is identified to correct the problem.

- The Allow Additional Cluster (ALWADDCLU) network attribute is not set to *ANY or *RQSAUT. Use either of the following commands:

```
CHGNETA ALWADDCLU(*ANY)
CHGNETA ALWADDCLU(*RQSAUT)
```

- Host Servers are not started:

```
STRHOSTSVR *ALL
```

- The user profile does not have correct authority:

```
CHGUSRPRF user_profile SPCAUTH(*IOSYSCFG *ALLOBJ *SERVICE)
```

- TCPIP Internet Daemon is not started:

  ```
  STRTCPSVR *INETD
  ```

- The user profile of the user creating the cluster should be the same on all nodes in the cluster

5.3 Using QUSRTOOL CL commands and OS/400 APIs to implement an iSeries cluster

The pairing of OS/400 APIs and CL commands is one of two supported interfaces when full clustering is not required. The other supported interface for simple clustering is described in 5.2, “Using the Operations Navigator GUI to implement and manage iSeries clusters” on page 61.

Notes:

- The commands described in this section (and chapter) are part of the QUSRTOOL library that is shipped with OS/400 V5R1. OS/400 native commands do not support iSeries clustering at this time. You can find a list of the commands and APIs available in the QUSRTOOL library in B.3, “Cluster APIs and related QUSRTOOL commands” on page 286.
- In the following steps, both of the commands offered in QUSRTOOL and base OS/400 commands are illustrated. You may choose to use one or the other.

To use CL commands to create a non-complex iSeries cluster, follow these seven easy steps:

1. Perform these functions on each node in the cluster:
   a. Change the network attributes to allow a cluster:

      ```
      CHGNETA ALWADDCLU(*ANY)
      ```

   b. Activate TCP using:

      ```
      STRTCP
      ```

      This command also starts the INETD server job (QTOGINTD).
   c. Verify the connection to the system:

      ```
      PING system-name
      ```

      Record the TCP/IP address.
   d. Make sure the controlling subsystem can handle more than one batch job:

      ```
      DSPSYSVAL QCTLSBSD
      WRKSBSD ctl-sbs
      ```

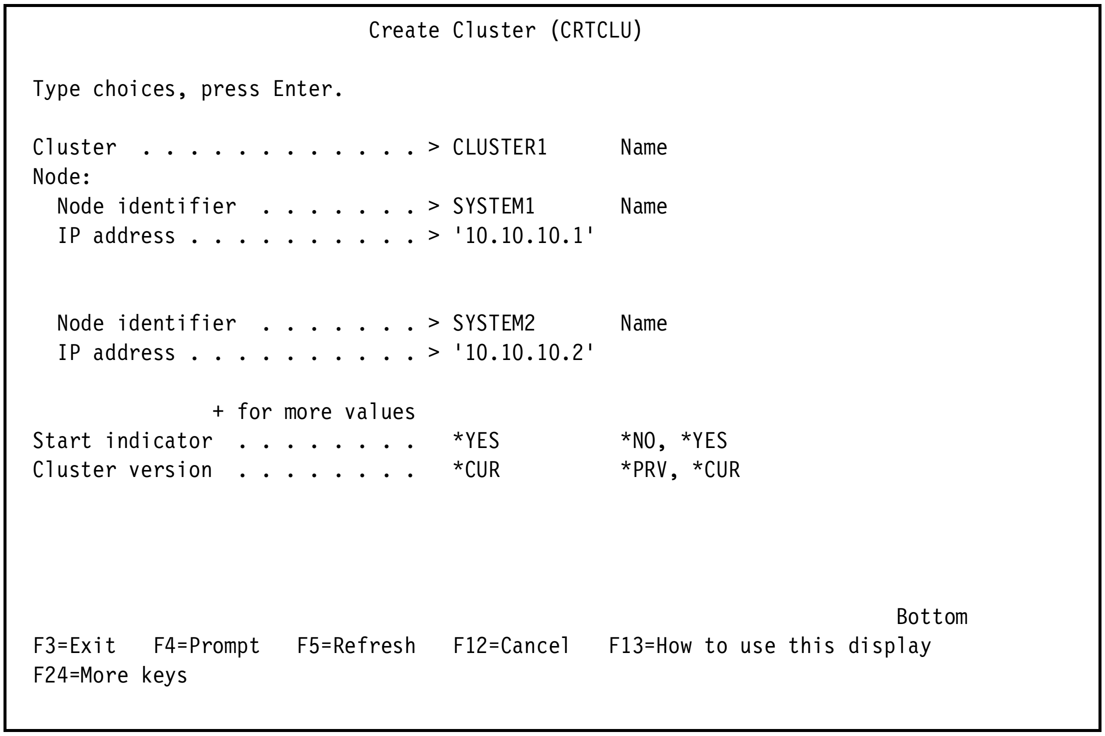
      i. Select option **6** (Job queue entries). Note the value for *Max Active* for the job queue listed. If it is not *NOMAX or a value greater than 1, a cluster can appear to be hung as batch jobs execute.
      ii. If job queue is not *NOMAX or a value greater than 1, enter:

      ```
      CHGJOBQE SBSD(lib-name/subsystem-name) JOBQ(lib-name/job-queue-name)
      MAXACT(*NOMAX)
      ```

2. Create the cluster, using either of the following commands:

   ```
   CRTCLU (QUSRTOOL)

   CRTCLU CLUSTER(cluster-name) NODE('ip-address'/system-name)
   ```

See Figure 5-31 for an illustration.



```
                            Create Cluster (CRTCLU)

Type choices, press Enter.

Cluster  . . . . . . . . . . . . > CLUSTER1      Name
Node:
  Node identifier  . . . . . . . > SYSTEM1       Name
  IP address . . . . . . . . . . > '10.10.10.1'


  Node identifier  . . . . . . . > SYSTEM2       Name
  IP address . . . . . . . . . . > '10.10.10.2'

               + for more values
Start indicator  . . . . . . . .   *YES          *NO, *YES
Cluster version  . . . . . . . .   *CUR          *PRV, *CUR





                                                                     Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 5-31 Create Cluster QUSRTOOL command

> **Note:** The Cluster version parameter in the Create Cluster command allows the user to specify whether this cluster is to be a type 2 cluster (V5R1) or a type 1 cluster (V4R4 and V4R5). *CUR indicates type 2, and *PRV indicates type 1. This fact is important if the type 1 nodes are to be added to the cluster.

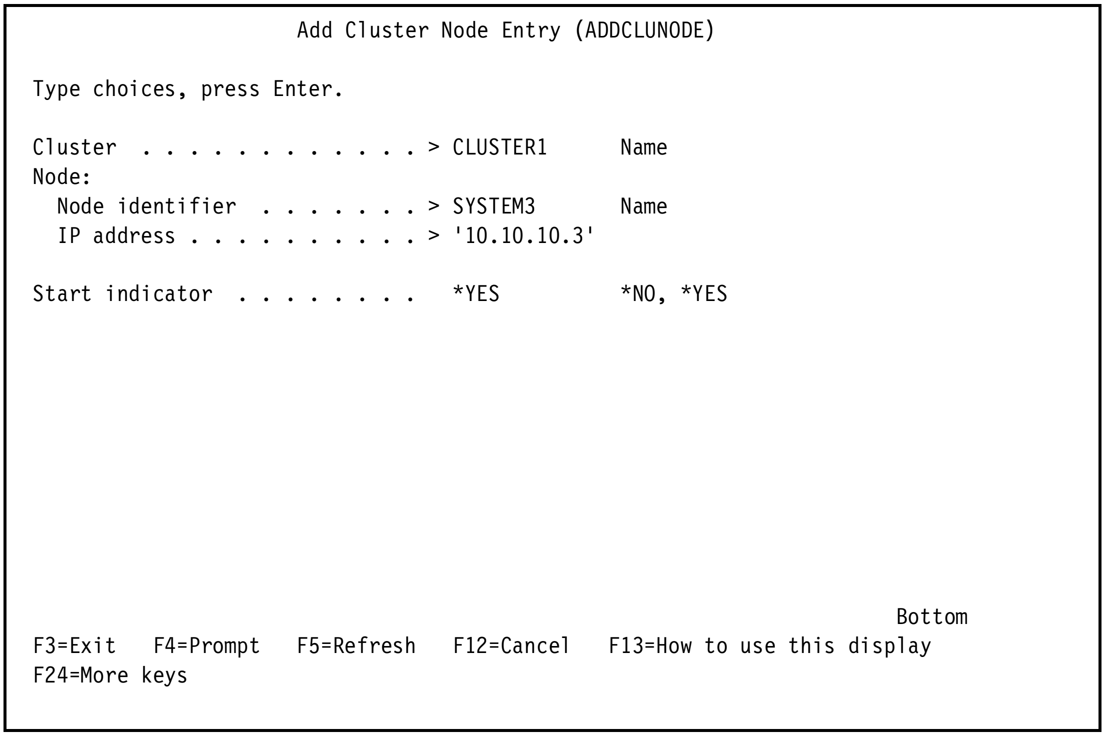
3. Add another node to the cluster from the active cluster node:

   ```
   ADDCLUNOD (QUSRTOOL)
   ```

   Use the IP address from the ping operation completed in step 1, for example:

   ```
   ADDCLUNOD CLUSTER(cluster-name) NODE('ip-address'/system-name)
   ```

   See Figure 5-32 for an illustration.



```
                      Add Cluster Node Entry (ADDCLUNODE)

 Type choices, press Enter.

 Cluster  . . . . . . . . . . . . > CLUSTER1      Name
 Node:
   Node identifier  . . . . . . . > SYSTEM3       Name
   IP address . . . . . . . . . . > '10.10.10.3'

 Start indicator  . . . . . . . .   *YES          *NO, *YES










                                                                      Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

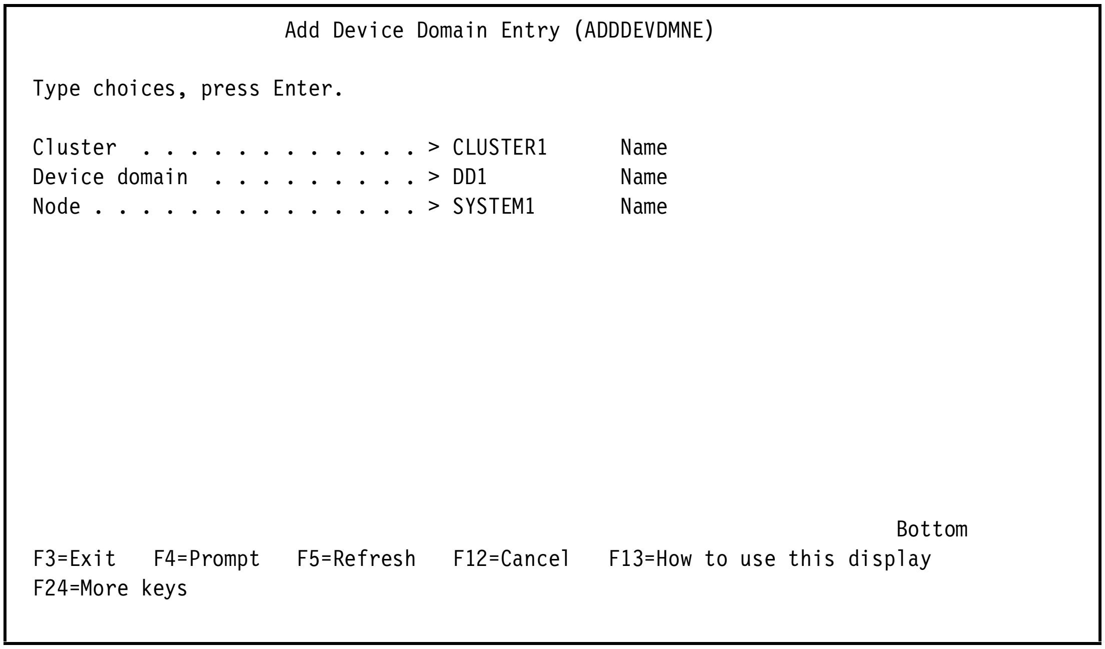
Figure 5-32 Add Cluster Node Entry QUSRTOOL command

4. Define the device domains.

   When you are using APIs or QUSRTOOL commands, device domains must be created manually. (The Simple Cluster Management GUI conceals this step.) Use either of the following commands:

```
ADDDEVDMNE(QUSRTOOL)

ADDDEVDMNE CLUSTER(cluster-name) DEVDMN(choose-a-name-for-the-device-domain)
NODE(system-name)
```

   See Figure 5-33 for an illustration.



```
                     Add Device Domain Entry (ADDDEVDMNE)

 Type choices, press Enter.

 Cluster  . . . . . . . . . . . . > CLUSTER1      Name
 Device domain  . . . . . . . . . > DD1           Name
 Node . . . . . . . . . . . . . . > SYSTEM1       Name















                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 5-33 Add Device Domain Entry QUSRTOOL command

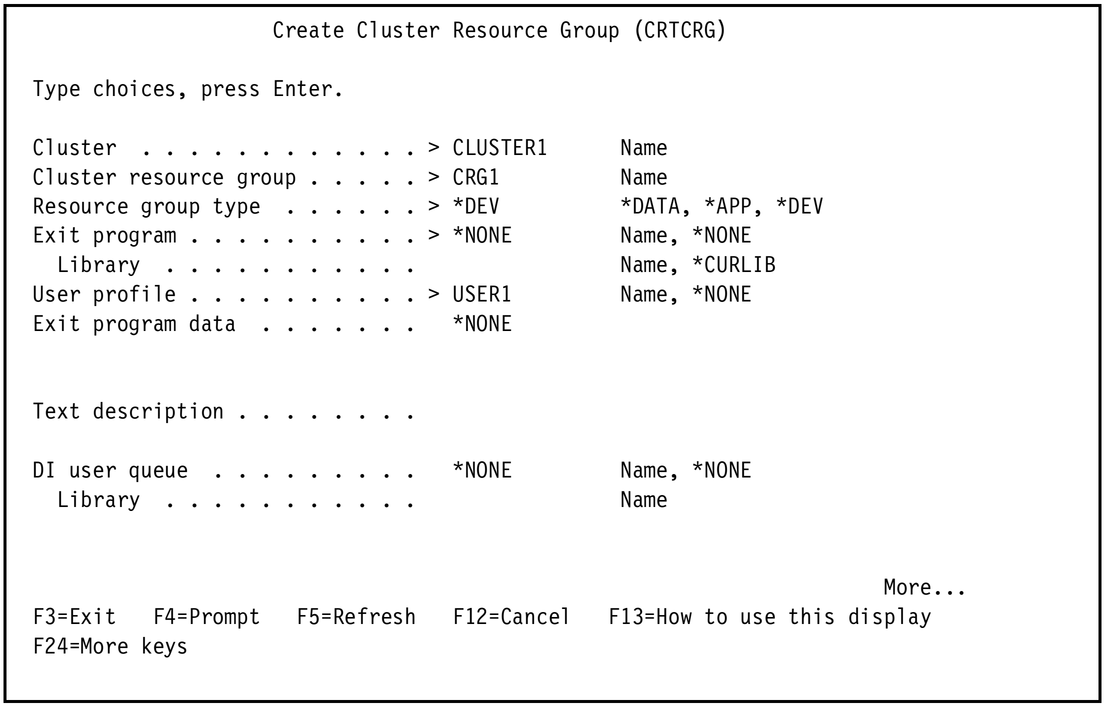
5. Create the Cluster Resource Group, using the command:

   ```
   CRTCRG
   ```

   For example, to create a Device CRG, enter:

   ```
   CRTCRG CLUSTER(cluster-name) CRG(crg-name) CRGTYPE(*dev) TEXT(TEXT) EXITPGMDTA(*NONE)
   USERPRF(*NONE) RCYDMN((system-name *prim) (system-name *back1))
   CFGOBJL((iasp-devd-name))
   ```

   See Figure 5-34 for an illustration.



```
                    Create Cluster Resource Group (CRTCRG)

Type choices, press Enter.

Cluster  . . . . . . . . . . . . > CLUSTER1      Name
Cluster resource group . . . . . > CRG1          Name
Resource group type  . . . . . . > *DEV          *DATA, *APP, *DEV
Exit program . . . . . . . . . . > *NONE         Name, *NONE
  Library  . . . . . . . . . . .                 Name, *CURLIB
User profile . . . . . . . . . . > USER1         Name, *NONE
Exit program data  . . . . . . .   *NONE


Text description . . . . . . . .

DI user queue  . . . . . . . . .   *NONE         Name, *NONE
  Library  . . . . . . . . . . .                 Name



                                                                   More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 5-34 Create CRG QUSRTOOL command

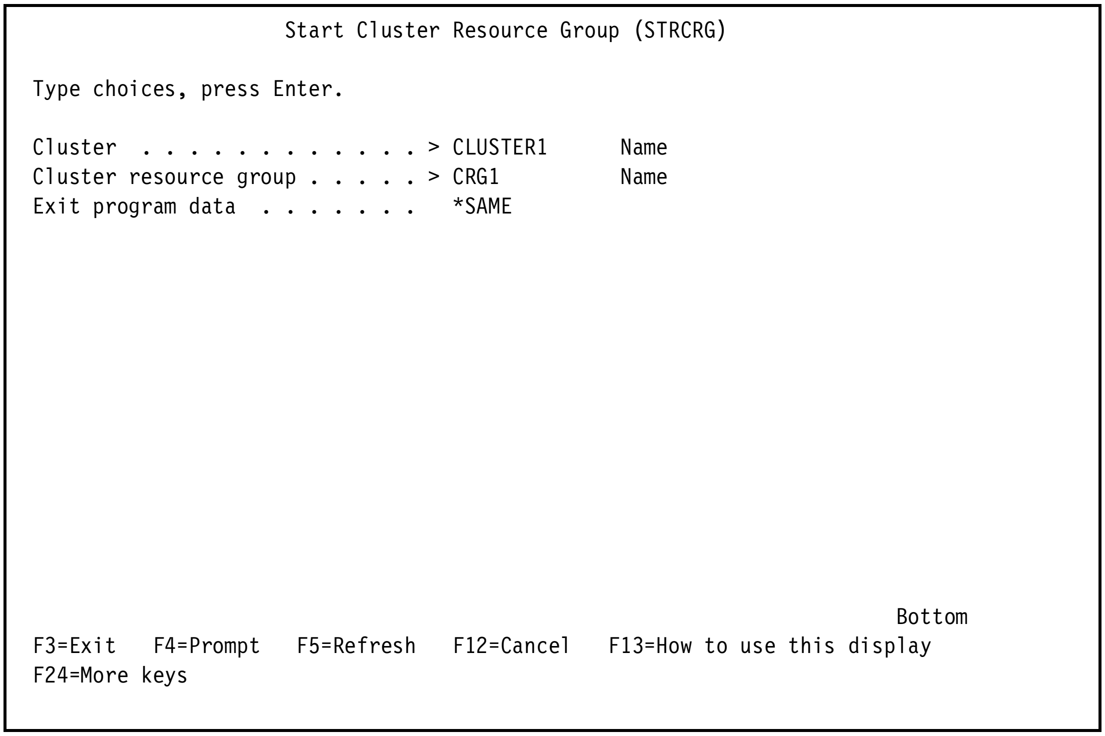
6. Start the CRG using the command:

   `STRTCRG`

   Consider this example:

   STRCRG CLUSTER(*cluster-name*) CRG(*crg-name*)

   See Figure 5-35 for an example.



```
                    Start Cluster Resource Group (STRCRG)

 Type choices, press Enter.

 Cluster  . . . . . . . . . . . . > CLUSTER1      Name
 Cluster resource group . . . . . > CRG1          Name
 Exit program data  . . . . . . .   *SAME


















                                                                        Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 5-35 Start Cluster Resource Group command

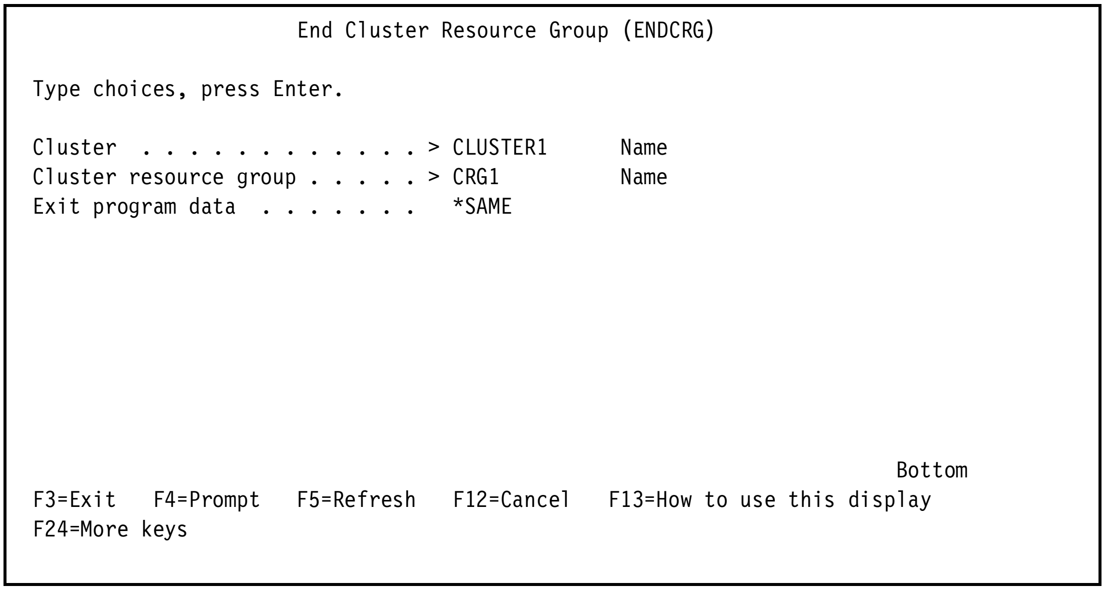
7. End the CRG using the command:

   ```
   ENDCRG
   ```

   Consider this example:

   ```
   ENDCRG CLUSTER(cluster-name) CRG(crg-name)
   ```

   See Figure 5-36 for an illustration.



```
                    End Cluster Resource Group (ENDCRG)

Type choices, press Enter.

Cluster  . . . . . . . . . . . . > CLUSTER1      Name
Cluster resource group . . . . . > CRG1          Name
Exit program data  . . . . . . .   *SAME









                                                                     Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 5-36 End Cluster Resource Group QUSRTOOL command

Additional CL commands are available to further enhance the cluster by adding nodes, devices, remove domains and nodes, and print information. You can find a list of QUSRTOOL commands in B.3, “Cluster APIs and related QUSRTOOL commands” on page 286.

5.3.1 Using OS/400 cluster management APIs to implement a cluster

OS/400 APIs are part of one of the two supported interfaces when full iSeries clustering support is not required. The APIs provided with OS/400 support developing code for the management and creation of a simple cluster environment.

The other supported interface is described in 5.2, “Using the Operations Navigator GUI to implement and manage iSeries clusters” on page 61.

> **Note:** Since APIs are designed to be used within a custom application, example coding is not provided in this redbook.

To configure a cluster, you need to understand the attributes for each node of the cluster:

- What nodes are in the cluster (what are the iSeries servers that make up the cluster)
- What the current cluster version is
- What the cluster resources (the resilient objects and resilient applications) are
- What the policies related to failover or switchover for the cluster policies are
- What the required CRGs are
- Obtain IP addresses for each application CRG; the IP addresses must not be used by other applications
- Decide the level required level for cluster messaging

You can find a list of APIs that support clustering in B.3, “Cluster APIs and related QUSRTOOL commands” on page 286.

5.3.2 Definitions revisited

Using the APIs and commands requires a good knowledge of cluster definitions and how they relate to one another. Before we look at cluster creation using the cluster APIs, it is important that you understand the terminology that is related to clusters.

A definition and brief description of the terms that are important to understand iSeries clustering are provided in this section. Each term is described in more detail in the remainder of this redbook.

- **Cluster node**: Cluster node is any iSeries server that is a member of a cluster. Any name can be used. However, it can be simplest if the node name is the same name as is used for either the host name or the system name.

  The cluster node name is associated with one or more Internet Protocol (IP) addresses that represent an iSeries server. Cluster communications makes use of the TCP/IP protocol suite to provide the communications paths between cluster services on each node in the cluster. The set of cluster nodes that are configured as part of the cluster is referred to as the *cluster membership list*.

- **Cluster Resource Group**: A Cluster Resource Group is an OS/400 system object that is a set or grouping of cluster resources.

  The group describes a recovery domain and supplies the name of the CRG exit program that manages cluster-related events for that group. One such event is moving an access point from one node to another node.

  CRG objects are defined as data resilient, application resilient, or device resilient. Data resiliency enables multiple copies of data to be maintained on more than one node in a cluster and enables the point of access to be changed to a backup node.

  Application resiliency enables an application (program) to be restarted on either the same node or a different node in the cluster. Device resiliency enables a device resource to be moved (switched) to a backup node. Every Data and Application CRG has a CRG exit program associated with it. The exit program is optional for resilient Device CRGs.

- **Replication**: Replication involves copying objects from one node in a cluster to one or more other nodes in the cluster.

  Replication makes a copy of objects in real time. Replication makes and keeps the objects on the clustered servers identical. A change to an object on one node in a cluster is replicated to other nodes in the cluster. Replication can be done through the use of journaling.

  For more information on how journaling works, see *iSeries Backup and Recovery*, SC41-5304.

- **Resilient resource**: Resilient resources are highly available with clustering implemented on the systems of the cluster. These system resources can be resilient:

  - Cluster node
  - CRG
  - Replicated data
  - Resilient applications
  - Switchable Internet Protocol (IP) address
  - Resilient device

If the primary access point for a particular set of resilient resources in a cluster node incurs an outage, another cluster node that is defined as the backup for that set of resources becomes the access point.

The definition of the relationship between the nodes associated with a set of resilient resources is found in the CRG object. CRGs are replicated and coordinated across the nodes in the cluster through Cluster Resource Services.

- **Recovery domain**: A recovery domain is a subset of nodes in the cluster that are grouped together in a CRG for a common purpose, such as performing a recovery action.

  A domain represents those nodes of the cluster from which a cluster resource can be accessed. The subset of cluster nodes that is assigned to a particular CRG supports either the primary point of access, secondary (backup) point of access, or the replicate. Each node in the recovery domain has a role with respect to the current operational environment of the cluster. This is known as its current role in the recovery domain. As the cluster goes through operational changes, such as when nodes end, start, or fail, the node's current role is changed accordingly.

  Each node in the recovery domain has a role with respect to the preferred or ideal cluster environment. This is known as its preferred role in the recovery domain. The preferred role is a static definition that is initially set when the CRG is created. As the cluster environment changes, the preferred role is not changed.

- **Device domain**: A device domain is a collection of cluster nodes that share resources.

  More specifically, nodes in a device domain can participate in a switching action for a collection of resilient device resources. Device domains are identified and managed through a set interface that allows the user to add a node to a device domain or remove a node from a device domain.

  Device domains are used to manage certain global information necessary to switch a resilient device from one system to another. All systems in the device domain use this information to ensure that conflicts do not occur when devices are switched. For example, for a collection of switchable IASPs, the IASP identification, disk unit assignments, and virtual address assignments must be unique across the entire device domain.

  A cluster node can belong to at most one device domain. A node must first be as a member of a device domain before a node can be added to a recovery domain for a device CRG. All nodes that are to be in the recovery domain for a resilient device CRG must be in the same device domain.

  Install Option 41 of OS/400, HA Switchable Resources, on the system to create and manage device domains. A valid license key must exist for this option.

  For more details on managing device domains, refer to the Add Device Domain Entry (QcstAddDeviceDomainEntry) and Remove Device Domain Entry (QcstRemoveDeviceDomainEntry) APIs.

- **Failover and switchover**: A failover is when the system automatically switches over to one or more backup systems in the event of a system failure. A switchover involves a manual switch to switch the access from one system to another.

  A switchover is done, for example, to perform system maintenance, such as applying program temporary fixes (PTFs), installing a new release, or upgrading the system.

  When multiple CRGs are involved in a failover action, the system processes the device CRGs first, the data CRGs second, and the application CRGs next. For an administrative switchover of multiple CRGs, consider the relationships between the CRGs when specifying their order. For example, if an application CRG depends on data associated with a device CRG, the steps of an ordered switchover are:

a. Stop the application on the old primary (to quiesce changes to the data)
b. Switch the device CRG to the new primary
c. Switch the application CRG to the new primary
d. Restart the application on the new primary

- **Cluster version**: A cluster version represents the level of clustering function available on the system. Functions are improved and added in each release of OS/400.

  Versioning is a technique that allows the cluster to contain systems at multiple release levels. The systems fully interoperate by determining the communications protocol level to be used.

Refer to the “Glossary” on page 295 for a list of additional definitions of terms that relate to clustering on the iSeries server.

5.3.3 Cluster versions

There are two cluster version levels:

- **Potential cluster version** represents the more advanced level of cluster function that is available for a given node. With this level, the node is capable of communicating with the level of each of the other cluster nodes.
- **Current cluster version** represents the version actually used for all cluster operations. This level facilitates the nodes to communicate within the cluster.

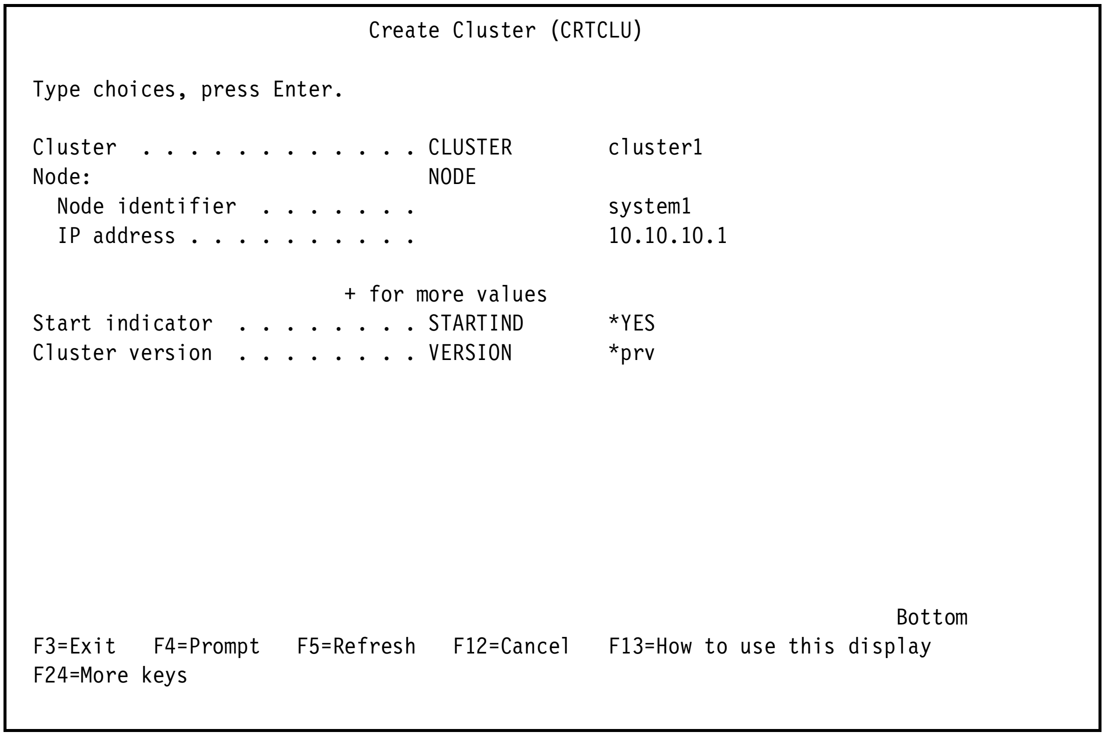
The potential cluster version increments with each OS/400 release that has significant function not available in earlier cluster versions. If the current cluster version is less than the potential cluster version, then that function cannot be used since some nodes cannot recognize or process the request. To take advantage of such new function, each system in the cluster must be at the same potential cluster version. Set the cluster version to that level.

When a node attempts to join a cluster, its potential cluster version is compared against the current cluster version. If the value of the potential cluster version is not the same as current version (N) or not equal to the next version level (N+1), then the node is not allowed to join the cluster.

The current cluster version is initially set by the first node defined in the cluster, using the value specified on the create cluster function. See the Create Cluster API (QcstCreateCluster) for more information.

For example, to have OS/400 V4R4 and V4R5 nodes exist with V5R1 nodes, you need to perform one of the following steps:

- Create the cluster on a V4R4 or V4R5 system and add in the V5R1 node.
- Create the cluster on a V5R1 system. Specify “`Allow previous nodes to be added`” to the cluster. Then add V4R4 and V4R5 systems to the cluster. See Figure 5-37.



```
                              Create Cluster (CRTCLU)

Type choices, press Enter.

Cluster  . . . . . . . . . . . . CLUSTER       cluster1
Node:                            NODE
  Node identifier  . . . . . . .               system1
  IP address . . . . . . . . . .               10.10.10.1

                          + for more values
Start indicator  . . . . . . . . STARTIND      *YES
Cluster version  . . . . . . . . VERSION       *prv








                                                                        Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 5-37 Create Cluster specifying previous cluster version

In a mixed release cluster, cluster protocols are run at the lowest node release level (N). The protocol is defined when the cluster is first created. N can be set either to the potential node version running on the node that originated the create cluster request, or one cluster version previous to the originators potential node version. Nodes in the cluster can differ by at most one cluster version level.

Once all systems in the cluster are upgraded to the next release, upgrade the cluster version so that new functions are available. Use the adjust cluster version function for this purpose.

See the brief description of the Adjust Cluster Version (QcstAdjustClusterVersion) API Table 5-1 on page 61 to relate this API with other Cluster APIs.

6

Independent ASPs explained

An independent ASP (IASP) is a configuration of multiple disk storage units into one group. The group can include all the disk units in one I/O tower, a subset of the disk units in an I/O tower, or disk units from two or more I/O towers.

This chapter presents information on the concepts, objectives, architecture, and design of independent ASPs. This information is useful not only to understand IASPs, but to market and strategically plan for an IASP implementation.

6.1 What independent ASPs are

Auxiliary storage pools (ASPs) have been part of the iSeries architecture since the announcement of the AS/400 in 1988, and of the System/38 before it. System ASPs and user ASPs enable a division of the total disk storage on the system into logical groups. One or more applications or data objects can then be isolated onto one or more ASPs, to support improvements for backup and recovery, performance, and more.

What are independent ASPs (IASPs)? IASPs are like user ASPs (described in 6.3, "OS/400 IASP support and terminology" on page 102). The "I" in the acronym IASP stands for "independent", which makes a significant difference.

Independent ASPs can be used on a single system, or switched between multiple systems or LPAR partitions. When used on a single system (as a standalone IASP), the IASP can be dynamically turned on or off. The IASP, and therefore, its contents – application and data – are dynamically made available or unavailable to the system.

When used across multiple systems, the IASP, and therefore, its contents – application and data – can be switched between those systems. The ability to be turned off or on, or to be switched between systems, is what differentiates IASPs from regular ASPs and provides IASPs the independent characteristics. The characteristic for an IASPs to be varied off or on, or attached and detached, can be done without performing a system IPL. This increases the flexibility offered by IASPs.

Clustering support is required to switch IASPs across multiple systems. Cluster Management is used to switch the IASP across systems in the cluster. At any one time, the IASP can be used from one of the clustered systems. The IASP cannot be used simultaneously from multiple systems.

The ability for one iSeries server to pickup or inherit the disks of another iSeries server with all its contents, without a system IPL, is a revolutionary concept. As a result, it is expected that IASPs will open up many useful application possibilities for iSeries customers.

6.2 Uses for IASPs

A predictable question for any new function in OS/400 is: What do I use it for? This section identifies some of the uses for IASPs.

Independent ASP supports objects in the Integrated File System (User Defined File System (UDFS)). Therefore, most of the uses in OS/400 V5R1 relate to applications that use the IFS.

Some of the more apparent uses of independent ASPs at V5R1 are:

- To use independent ASPs to store anything that resides in the Integrated File System (IFS), for example:
  - Web pages
  - Domino data directories

    Domino's use of independent ASPs for its data directory is described in 6.7.1, "Domino and independent ASPs" on page 112.
  - Linux file system
- Storage for Integrated xSeries Server for iSeries (Integrated Netfinity Server)
- Tivoli Storage Manager storage

- Applications that use the Integrated File System, such as:
  - MQSeries
  - Applications ported from other systems which utilize the IFS for storage, for example UNIX or Windows NT
  - Data mining and business intelligence data if the application uses the Integrated File System
- To store separate applications and their associated data on separate IASPs

  For example, store Domino data on one IASP, and store HTTP Server data on another IASP.
- To store separate instances of data related to a specific application on separate IASPs

  For example, store Domino data for one customer on one IASP, and store Domino data for a second customer on a different IASP.
- System upgrades

  It is believed that the IASP can be used for system hardware and software upgrades. That is, switch the IASP to the backup system and upgrade the primary system while the backup system keeps production operation. After the upgrade, switch back to the primary system. Note that at the time of writing this redbook, this use had not been tested and therefore is not supported by IBM.

Although the more “glamorous” or high profile use of an IASP is as an ASP that can be switched between systems, there are other valid uses for a standalone IASP, for example:

- In a single system environment, an independent ASP can be used to store certain data offline except for the periods when it is actually needed.

  The isolation provided by storing data offline means that there is less work necessary for certain functions such as IPL, reclaim storage, and some save operations.
- A single system IASP can be a stepping stone for the more complex usage of the switchable IASP.

Note: An Enterprise Storage Server (commonly known as Shark) that is attached to a switchable IASP tower switches when the IASP switches. However, individual Logical Unit Number (LUN) sets of the Shark cannot be switched. The Shark must switch as an entity.

Some of these uses are discussed within this chapter. Other uses for an IASP are expected to evolve over time. It is anticipated that applications will look to store journal receivers on an IASP once that support is available.

Attention: IBM has released a statement of direction that other objects, such as libraries, database files, and journals and journal receivers, will be supported on the IASP in a later release. The statement says:

In a future release, IBM intends to enhance OS/400 support for switching the ownership of objects between primary and secondary servers through the use of independent ASPs and switchable disk technologies by extending the list of possible switched objects beyond the IFS files. Additions would include objects associated with the DB2 UDB for iSeries and other OS/400 library-based objects.

Look for the redbook *Moving Applications to Switchable Independent ASP*s, SG24-6802, which is scheduled for publication later in the second half of 2002.

6.3 OS/400 IASP support and terminology

The types and quantity of ASPs supported on the iSeries servers depend upon the release of OS/400 installed on the system. Table 6-1 summarizes these characteristics.

Table 6-1 Type and quantity of ASP support by OS/400 release

Type of ASP	OS/400 release supported	ASP number	Maximum quantity supported on the system
System ASP	All	ASP 1	1
User ASP*	V5R1	ASP 02 - ASP 32	31
User ASP*	V4R5 and earlier	ASP 02 - ASP 16	15
Independent ASP	V5R1	ASP 33 - ASP 99	67
* Also known as Basic ASP			

Other characteristics of ASPs include:

- **System ASP**

  The system ASP contains SLIC and OS/400 code. It can contain objects of any object type.

- **User ASP**

  User ASPs are any ASP other than the system ASP.

  The additional 16 user ASPs supported with V5R1 have the same characteristics as the traditional ASP numbers 2 through 16 originally available with the AS/400 system.

- **Independent ASPs**

  Independent ASPs are user ASPs that can be varied offline or varied online independent of the rest of the server storage. IASPs can be switched between iSeries servers (and partitions).

  Independent ASPs are known to the system by both names and numbers (33 through 99). The preferred user interface is by name. In many cases, the user interface does not support a reference by IASP number.

  The iSeries Machine Interface (MI) recognizes both names and numbers. A machine instruction associates an IASP name with its number.

Figure 6-1 further illustrates the V5R1 supported characteristics of ASPs.

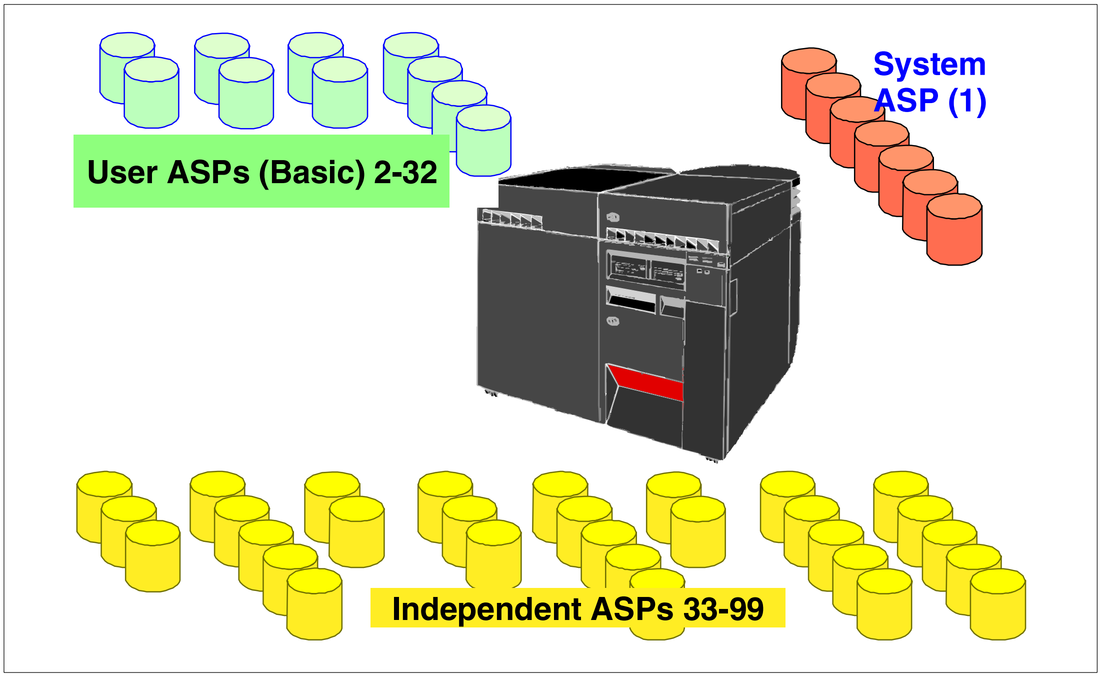


Figure 6-1 ASPs at V5R1

With V5R1 the user interface for the iSeries is graphical, using Operations Navigator. Within the GUI tools, the term *disk pool* is used for ASPs. The terms *disk pool* and *auxiliary storage pool* (ASP) are synonymous.

The term *basic disk pool* also refers to the traditional user ASP. In pre-V5R1 terms, *basic* refers to pools 2 through 15. The scope of this definition is now expanded to include ASPs 2 through 32.

Figure 6-2 illustrates the relationship of ASPs using this disk pool terminology.

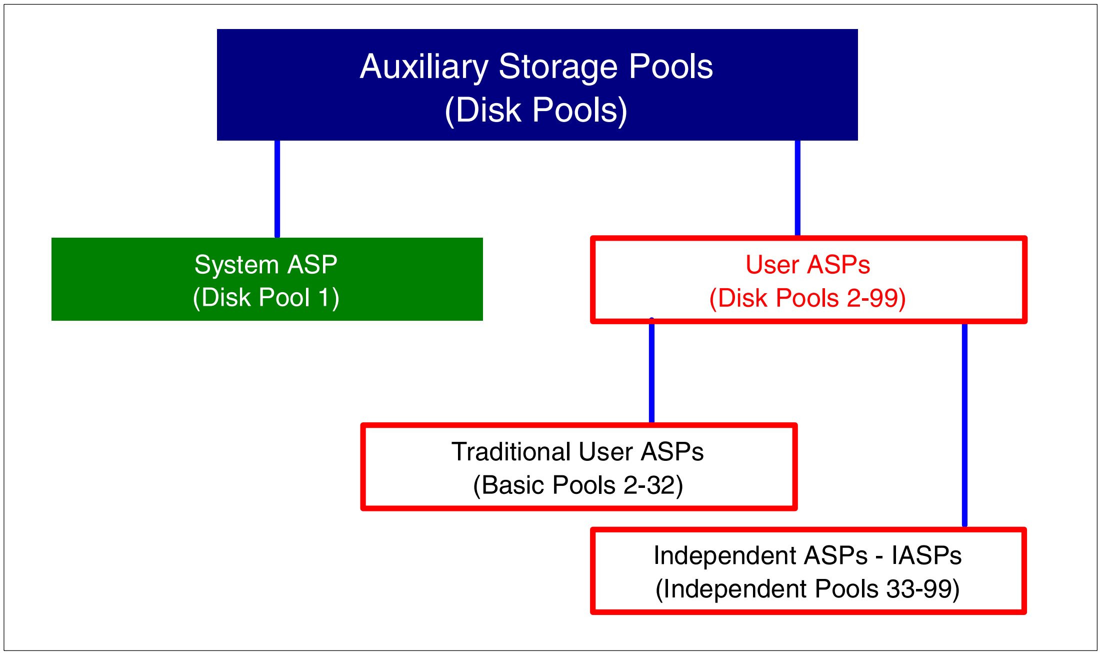


Figure 6-2 Additional ASP terminology

6.3.1 OS/400 software options

Option 41 of the OS/400 is the High Availability Switchable Resources feature. This feature is required to enable the Cluster Management wizard in Operations Navigator and to enable switching of resilient devices, such as a switchable IASP.

Note: OS/400 Option 41 is not required to configure a standalone IASP.

There can be confusion regarding the requirement of OptiConnect and the use of IASPs.

Attention: To clarify the relationship of Option 23 and switched disks, remember OptiConnect over HSL (OS/400 Option 23) is not required to use a switchable or standalone IASP.

To configure and use a switchable IASP, HSL hardware connectivity supported with V5R1 is required. OptiConnect over HSL is a viable option for high speed system-to-system I/O. It is a useful option for applications, such as replication.

Enable OptiConnect by installing Option 23 of OS/400. It is a separately purchased feature.

6.3.2 High Speed Link OptiConnect

This section covers the HSL requirements and some of the clustering constructs that are necessary to create an IASP.

HSL technology is introduced with OS/400 V4R5 as the means to attach I/O towers to the base system unit. HSL fabric is useful for high-speed server-to-server interconnect, since it can run ten times faster than SPD OptiConnect at V4R5.

HSL is a pre-requisite for creating and using switchable IASPs. HSL is not required for a standalone IASP.

HSL OptiConnect replaces the SPD connection. This support does not require additional hardware.

Note: IBM plans to bring the Power4 microprocessor technology to the iSeries in the second half of 2002. IBM does not intend for the planned Power 4 iSeries servers to support non-PCI (SPD-based) I/O controllers and adapters.

Each iSeries model announced in April 2001 (Models 270 through 840) is shipped with the appropriate HSL port that can be used for HSL OptiConnect and for the new switching IASP support. The V4R5 iSeries 830 and 840 models (available before April 2001) use the older HSL port. However, these models can upgrade to the new HSL hardware.

The V4R5 iSeries 270s and 820s cannot have the newer technology port installed on their system, because they use the same boards to drive the processor or processors and HSL. A customer with a pre- April 2001 270 or 820 must order or upgrade to a new April 2001 270 or 820 processor to use the new HSL hardware.

HSL limitations are fully described in *AS/400e to IBM @server iSeries Migration: A Guide to System Upgrades at V4R5 and V5R1*, SG24-6055.

6.3.3 Configuration source and the IASP

Since an IASP can switch between systems, certain configuration information of the IASP is stored on the IASP itself to make it self contained.

The configuration information of the disks on which the IASP is contained in the configuration source (sometimes referred to as *load stone*) on the IASP itself. When the IASP switches, the IASP configuration information switches with it.

In general, the load source of the system keeps information on all ASPs on the system, including the independent ASPs. Detailed information for the basic ASPs is kept in the load source. Limited information for the independent ASPs is kept on the load source.

Detailed configuration information of the IASP is contained in the configuration source of the IASP. This is a key concept to remember to understand switching disks.

6.4 Considerations of switched disk and IASP use

When planning an iSeries availability solution, consider the characteristics of IASPs, as well as their advantages and disadvantages. For example, consider these statements regarding switched disk or IASPs when determining their value in an availability solution:

- The time to vary on an IASP during the switching process depends on the number of objects on the IASP, and not the size of the objects. If possible, keep the number of objects small.
- For a quicker vary on or off, keep the User-ID Number (UID) and Group-ID Number (GID) of user profiles that own objects on the IASP the same between nodes of the cluster. Having different UIDs lengthens the vary on time.
- All the disk units within a tower are switched as an entity.

  When a tower containing the IASP is switched, all other I/O devices on that tower (if any), such as tape drive, CD ROM, printers, etc., are also switched to the other node.
- The number of devices in a tower affects the switchover time.

  The devices in a switchover are reset as part of the switchover process.
- The devices within a tower that is switched can autoconfigure. This can result in different resource names on the switched-to node.

  Manually configure the devices on the node that the tower is switched to so that the resource names match on both systems.
- In an LPAR environment, the IOP controlling the disks is switched between the partitions.

  Make sure that the console is not on that IOP. Otherwise the console is switched with the IOP. Without the console, the primary partition is inaccessible.
- Objects in an IASP cannot be journaled.

  Since replication uses journals, objects in an IASP cannot be replicated to a backup system at V5R1.
- Library-based objects are not supported in the IASP at V5R1.

The rules regarding HSL differ dependent upon the model of the iSeries server. For example, the following information varies by system model:

- How many entities can be on an HSL loop
- How many of these entities can be switched

Refer to *AS/400e to IBM @server iSeries Migration: A Guide to System Upgrades at V4R5 and V5R1*, SG24-6055, to understand these requirements.

Example configurations to illustrate some of these implementation rules are outlined in 6.6, “IASP configuration examples” on page 110, and 6.7.2, “Integrated xSeries Server and independent ASPs” on page 113.

6.4.1 Advantages of using IASPs

Improvements to iSeries availability are enabled through the use of IASPs, beyond the capabilities of replication solutions. This section discusses the advantages of IASP use:

- For disk drives in the IASP, device parity protection can be stopped or started from within OS/400.

  For regular ASPs, stopping and starting device parity protection is a Dedicated Service Tools (DST) function.
- For disk drives in the IASP, mirroring can be turned on and off from within OS/400.

  For regular ASPs, stopping and starting mirroring is a DST function.
- IASPs enable a higher level of availability without the need to buy a duplicate set of disks for the backup system. In a sense, IASPs are the poor man's option for higher availability.
- The contents of a switchable IASP can be made available to the backup system without any kind of replication.
- It is not necessary to maintain multiple copies of data, programs, and other objects.

  Multiple copies of objects is a function of replication.
- There is minimal additional system overhead with IASP.

  Replication requires more CPU cycles when replicating to a backup system.
- There is no network traffic associated with IASP.

  Replication across a LAN or WAN involves network traffic.
- There is less work for system functions such as IPL, reclaim storage, and some save operations.

  In a single system environment, an independent ASP can be used to store certain data offline except for the periods when it is actually needed. The isolation provided by storing data offline means that there is less work necessary for system functions.
- Objects are not “in flight” in the event of a failure.

  With replication, it is possible that journal entries become “trapped” on the source system at the time of failure and do not arrive at the target machine.

6.4.2 Disadvantages of using IASP

There can be disadvantages to the system if IASPs are utilized, for example:

- IASPs represent a single point of failure in the system.

  If the disks in the IASP are permanently damaged and the data is unrecoverable, data is available only up to the last backup copy. IASPs protect the iSeries server against system failures, not against disk failure.
- Because of loop limitations with HSL, the systems must be within 250 meters using optical HSL cables or 15 meters with copper cables.

  The production and backup systems can be several thousand kilometers apart when replication is used. IASPs are, therefore, not useful as a disaster recovery solution.

- If the IASP configuration involves an HSL loop, a V5R1 supported HSL port card is required (such as HSL port features #2754, #2755, #2758, #2759, #2774, and #2777).

  HSL port cards available prior to V5R1 do not work with IASPs. However, systems with original HSL hardware can be replaced by newer HSL port cards.

- The IASP works with only one system at any one time. The IASP cannot be used for balancing workload.

  Typically customers use their backup system for read-only types of activities, such as creating or printing reports, running batch applications, and running queries. This helps spread workload to the backup system and thereby helps performance of the production system.

6.5 Objects defining an IASP configuration

A cluster, device domain, device CRG, and device description are configuration objects used to implement independent ASPs or clusters. This section introduces these structures. Implementation details are offered in Chapter 7, “Operations Navigator independent ASP GUI” on page 125.

6.5.1 Relationship of IASP configuration objects

Several object types are required to set up a *switchable IASP*:

- Cluster
- Device domain
- Device CRG
- Device description

The inter-relationship of each IASP and cluster configuration object is illustrated in Figure 6-3.

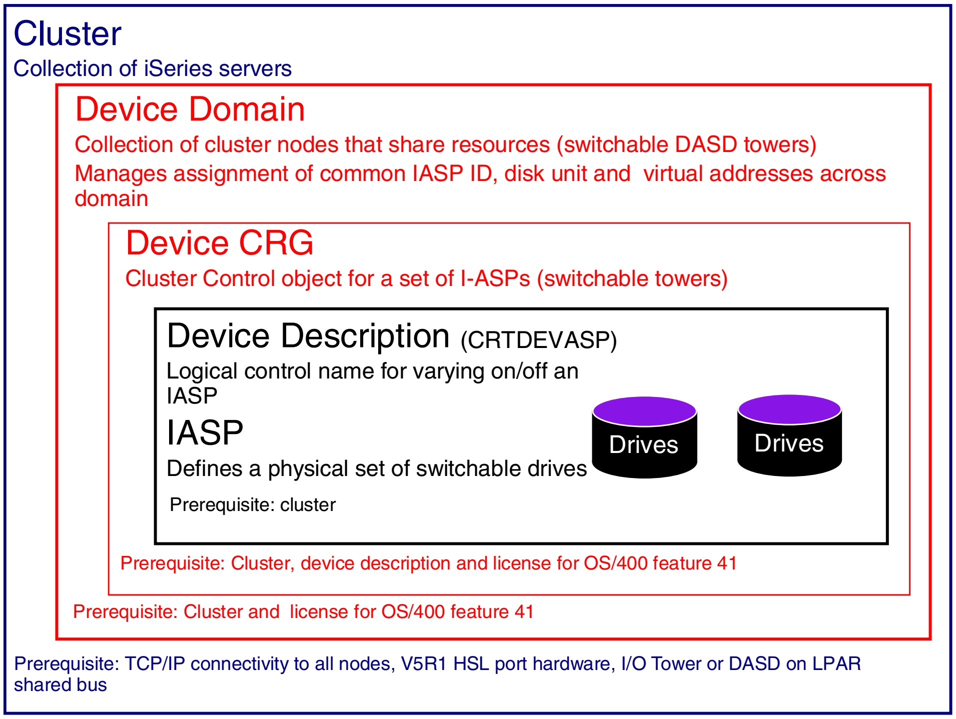

Cluster
Collection of iSeries servers
Device Domain
Collection of cluster nodes that share resources (switchable DASD towers)
Manages assignment of common IASP ID, disk unit and virtual addresses across domain
Device CRG
Cluster Control object for a set of I-ASPs (switchable towers)
Device Description (CRTDEVASP)
Logical control name for varying on/off an IASP
IASP
Defines a physical set of switchable drives
Drives
Drives
Prerequisite: cluster
Prerequisite: Cluster, device description and license for OS/400 feature 41
Prerequisite: Cluster and license for OS/400 feature 41
Prerequisite: TCP/IP connectivity to all nodes, V5R1 HSL port hardware, I/O Tower or DASD on LPAR shared bus

Figure 6-3 Switchable IASP object relationship

There is an enforced relationship between the resilient device CRG object and the switchable IASP physical resources. This relationship is enabled through OS/400 Option 41 - HA Switchable Resources. Option 41 is an additional cost, licensed option of OS/400.

A standalone IASP does not require a cluster definition. To set up a standalone IASP, use the GUI tool or the green screen interface to:

- Create a device description (using the Create Device ASP (CRTDEVASP) command)
- Assign the disks to the IASP
- Populate the IASP with data

Note: Switchable IASPs can only be created using the IBM Operations Navigator GUI tool.

6.5.2 Device domain object

A device domain is the first of the cluster constructs to be defined when creating a switchable IASP. It is a logical construct within Cluster Resource Services that is used to ensure that there are no configuration conflicts that prevent a switchover or failover.

The device domain is a subset of cluster nodes.

The set of configuration resources associated with a collection of resilient devices can be switched across the nodes in the device domain. Resource assignments are negotiated to ensure that no conflicts exist. The configuration resources assigned to the device domain must be unique within the entire device domain.Therefore, even though only one node can use a resilient device at any given time, that device can be switched to another node and brought online.

These cluster resources are negotiated across a device domain to ensure there are no conflicts:

- IASP number assignments

  IASPs are automatically assigned a number to correlate the name of the IASP. The user chooses the resource name. The system manages the assigned IASP numbers, which may not be in numerical order. The order depends on a number of factors, including the creation date and the creation of IASPs on other nodes in the device domain.

- DASD unit number assignments

  To keep from conflicting with the permanently attached disk units of each node, all IASP unit numbers begin with a four. IASP disk unit numbers start with the number 4001.

  Section 6.8.2, “Numbering new pools” on page 114, discusses DASD unit numbers in more detail.

- Virtual address assignments

  The cluster configuration determines the virtual address space required for the IASP. Virtual address assignments (the cluster configuration) are ensured not to conflict across all nodes in the device domain.

Note: With the Operations Navigator GUI wizard, the device domain, and the device CRG are created automatically by the wizard.

6.5.3 Device CRG object

A device CRG is the second clustering construct used to define and create a switchable IASP. Device CRGs or CRGs define the recovery domain for the switchable IASP. They provide the control for switching the IASP and for managing other aspects of the IASP.

Switched disks enable a storage tower that is connected via High Speed Link (HSL) OptiConnect to two local iSeries servers, to be switched to the server used for recovery in event of a failure of the primary server.

As entries in the device list, device CRGs support objects that are IASP devices only. Integrated File System objects only are allowed within the IASP at V5R1. Releases to follow OS/400 V5R1 will support database objects.

In V5R1, resilient device CRGs support these features:

- Are comprised of one or more IASPs
- Can be manually or automatically switched to a secondary node.

Cluster interfaces to support a resilient device CRG include functions to:

- Manage a device CRG
- Add or remove a resilient device list entry
- Call a CRG exit program

Note: A CRG exit program is optional. For simple clustering support offered in V5R1, the exit program is not required. The support for a CRG exit program is primarily to support high availability business partners.

There can be one or more IASPs in a device list for a specific device CRG.

6.5.4 IASPs, device domains, and device CRGs

Figure 6-4 illustrates the relationship between the device domain and device CRG when IASPs are implemented in a clustered environment.

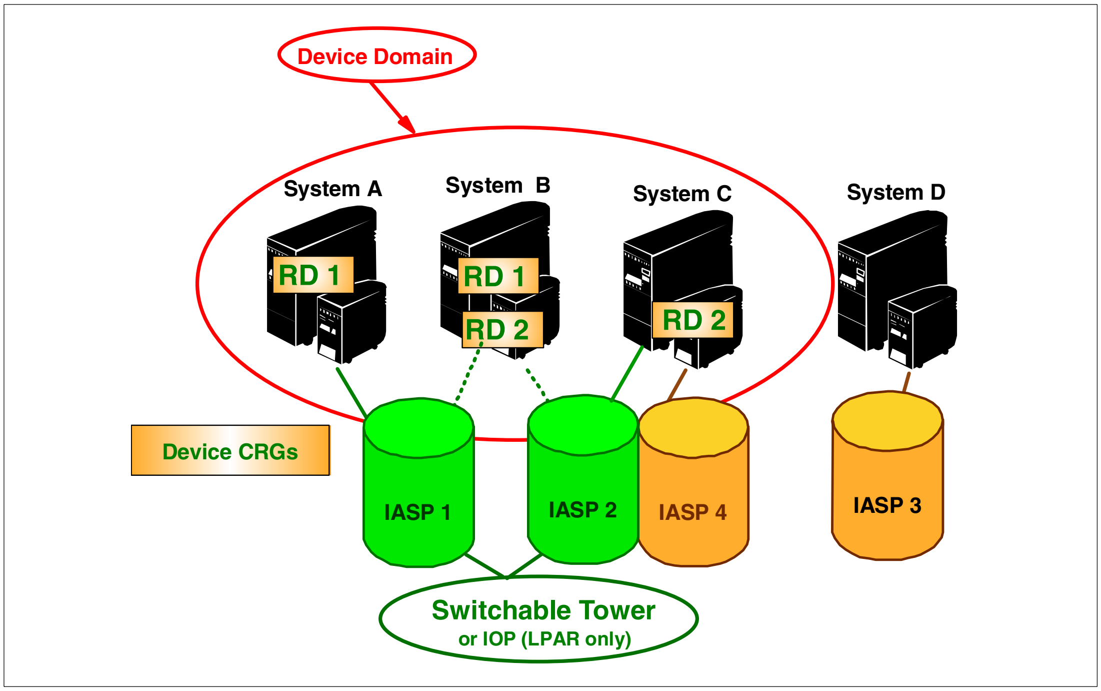


Figure 6-4 IASPs, device domains, and device CRGs

For this example, there are four systems: A, B, C, and D.

An independent ASP, named IASP4, is defined for System C. Another IASP, called IASP3, is defined for System D. Neither of these IASPs are intended to be switched to other systems. Consequently, there is no need for any system other than the owning system to be aware of their existence.

IASP1 is defined as switchable between Systems A and B. IASP2 is defined as switchable between Systems B and C. Therefore, Systems A and B contain configuration information about IASP1, and Systems B and C contain configuration information about IASP2.

The definition of the resilient devices named IASP1 and IASP2 results in an interrelationship between Systems A, B, and C. These systems form part of a device domain.

Actually there are two device domains in this example: one for IASP1 that includes Systems A and B, and the other for IASP2 that includes Systems B and C.

6.6 IASP configuration examples

An independent ASP is a method of configuring multiple disk storage units into one group. The group can include all the disk units in one I/O tower, a subset of the disk units in an I/O tower, or disk units from two or more I/O towers. To illustrate this concept, this section describes four of the possible IASP configurations.

The first example is an IASP configuration that is made up of all the disk units in a tower. This is shown in the diagram on the left side of Figure 6-5 (labeled as IASP33).

The second example illustrates an IASP configuration with two IASPs in the same tower. This is shown in the diagram on the right side of Figure 6-5. Independent IASPs numbered 38 and 39 each define a subset of the disks of the tower. Note that when the tower is switched, both IASPs are switched to the second system.

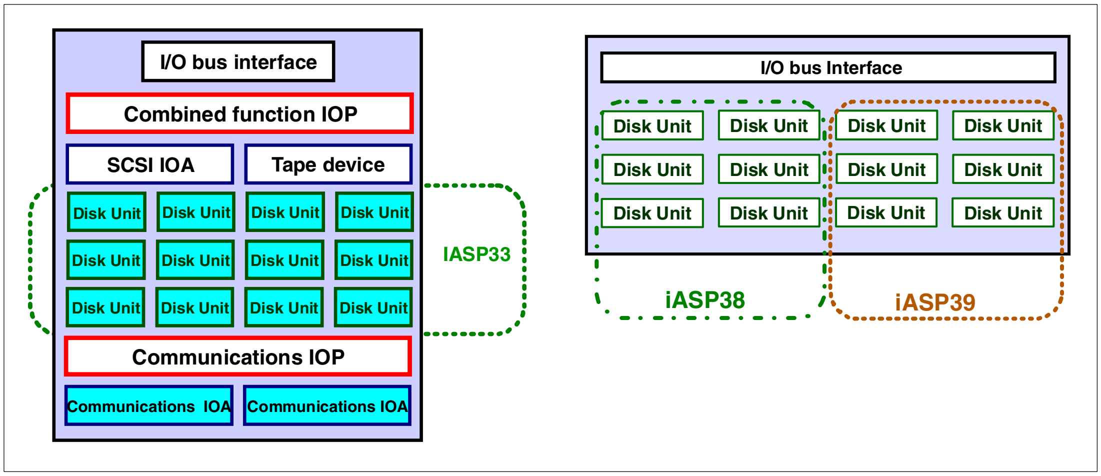


Figure 6-5 IASP configuration: Examples 1 and 2

The third example illustrates a three IASP configuration that uses two towers. See the diagram on the left side of Figure 6-6 for an illustration. Independent ASPs numbered IASP51 and IASP53 are made up of a subset of disks of their individual towers. IASP52 spans the two towers. It is made up of disks from both of the towers.

Note that, practically speaking, IASP52 does not represent an optimum configuration. This is because when IASP52 is switched, IASP51 or IASP53 is forced to switch with it. A better configuration is for IASP52 to be fully contained within one of the towers. Then IASP51 and IASP53 can share the second tower.

The fourth example illustrates that an IASP configuration (IASP45) can span two towers. See the diagram on the right side of Figure 6-6.

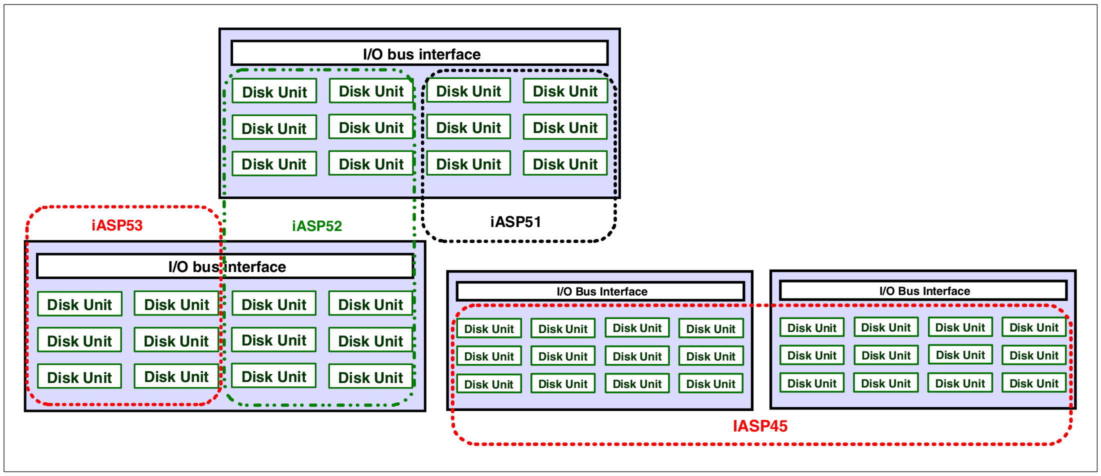


Figure 6-6 IASP configuration: Examples 3 and 4

While all devices packaged in a single I/O tower are switched from one iSeries server to another, clustering support allows for configuration and management of the IASPs using the disk units in the tower only. The non-ASP devices, such as, printers, tape drives, CD-ROM drive, etc., are usable by the iSeries server that now owns the tower. The user is responsible for performing the necessary device configuration, for example, varying on of the non-disk devices.

Specifically, the #5074 I/O tower is an enclosure commonly used to house the components of disk units, and therefore an IASP. Although the #5074 I/O tower can house devices other than disk units, only IASPs (switched disks) are supported by clustering in V5R1.

The #5079 tower is a hardware package that behaves the same as a #5074 tower, but contains twice as many disk units. In terms of clustering, the #5079 tower can be thought of as two #5074s.

6.7 IASP application examples

iSeries applications can readily benefit from the availability and recovery advantages offered by independent ASPs. This section briefly describes the use of IASPs in two application settings: Domino and the Integrated xSeries Server for iSeries.

6.7.1 Domino and independent ASPs

A Domino server can be defined on more than one system. Its data directory can reside on a switchable IASP. This allows a Domino server's data to be switched from one iSeries server to another. The same Domino server can be started on another iSeries server and access the data directory on the IASP. See Figure 6-7 for an illustration.

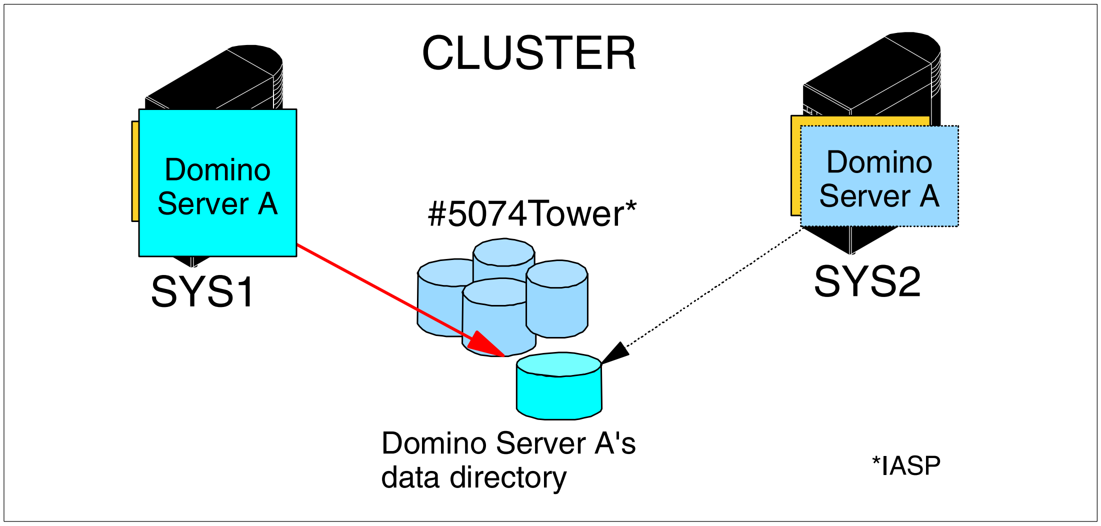


Figure 6-7 Domino using IASP

For a detailed discussion on complete Domino for iSeries clustering capabilities, refer to:
http://www.ibm.com/eserver/iseries/domino

6.7.2 Integrated xSeries Server and independent ASPs

Windows disks support an independent ASP. For example, if a server named Server A runs with the Integrated xSeries Server for iSeries, or a direct attached xSeries Server, the steps to switch the disk are:

1. Take iSeries Server A offline.

   The disk tower switches to iSeries Server B.

2. Manually link network server descriptions (NWSDs) to iSeries Server B resource names.
3. Reboot the Windows servers.

   Windows servers are back online on iSeries Server B.

See Figure 6-8 for an illustration.

Note: The xSeries servers must have the same configuration.

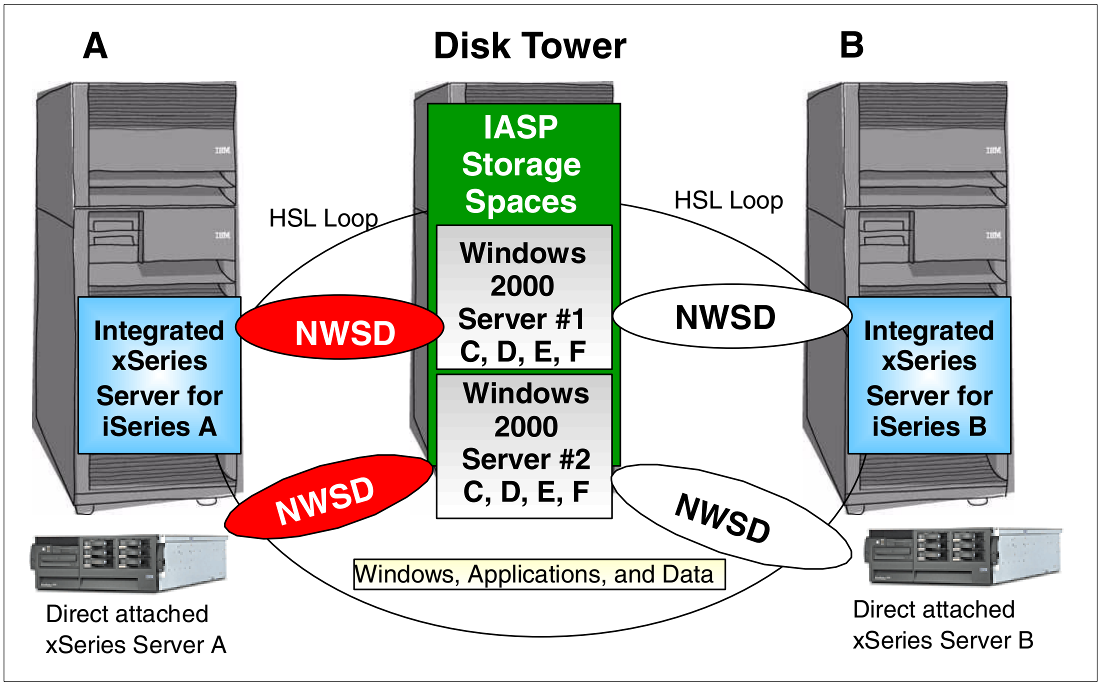

A
Disk Tower
B
IASP Storage Spaces
HSL Loop
HSL Loop
Windows 2000 Server #1 C, D, E, F
Windows 2000 Server #2 C, D, E, F
NWSD
NWSD
NWSD
NWSD
Integrated xSeries Server for iSeries A
Integrated xSeries Server for iSeries B
Windows, Applications, and Data
Direct attached xSeries Server A
Direct attached xSeries Server B

Figure 6-8 Integrated xSeries Server or Direct Attached xSeries Server using IASP

6.8 Configuring and managing IASPs

The simplest method to configure and manage an IASP is with the Operations Navigator interface. A “green-screen” DST interface can be used to configure ASPs. However, with V5R1, if a function is supported with Operations Navigator, a GUI is the recommended interface.

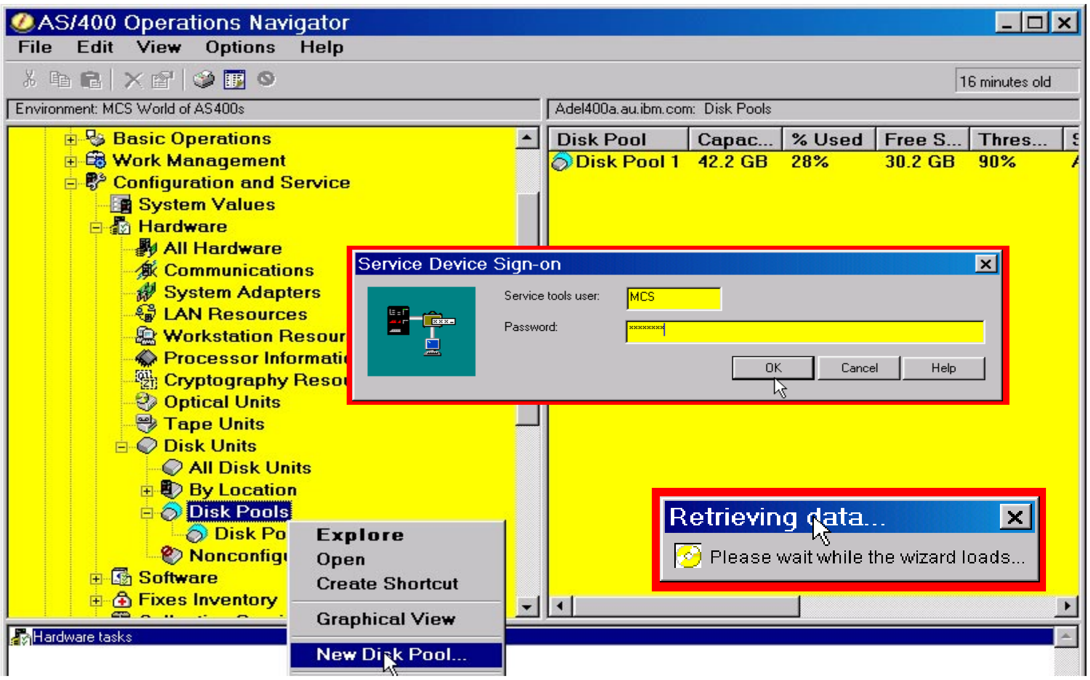
This section highlights the aspects of creating an IASP using the Operations Navigator. Detailed step-by-step instructions for creating the IASPs using the Operations Navigator are covered in Chapter 7, “Operations Navigator independent ASP GUI” on page 125.

6.8.1 Creating a new disk pool

One of the disk functions available in the hardware path of the Configuration and Service function of Operations Navigator is to create a disk pool. This section explains Operations Navigator initial steps to create a disk pool.

Access to the disk functions is controlled via the Service Tools User Profiles, a function of SST introduced at V5R1. Once signed into the Service Tools functions, select the option for disk pools (auxiliary storage pool).

In our example, the user selects the “New Disk Pool” option. A “retrieving data” message is displayed to notify the user that the request is being processed. From this starting point, follow the Disk Pool Wizard to create the disk pool. See Figure 6-9 for an illustration.



Disk Pool	Capac...	% Used	Free S...	Thres...
Disk Pool 1	42.2 GB	28%	30.2 GB	90%

Service tools user: MCS

Password: ********

Retrieving data...

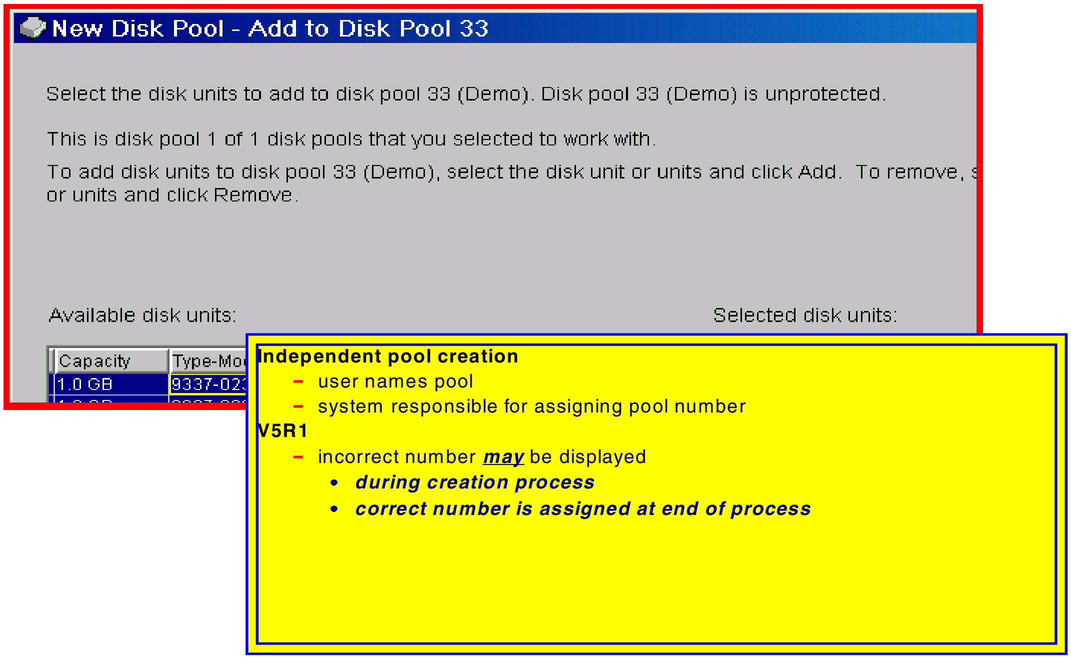
Please wait while the wizard loads...

Figure 6-9 Creating a disk pool using the GUI interface

Note: Only one Disk Pool Creation task can be run at any one time.

6.8.2 Numbering new pools

When creating independent disk pools, the user has the opportunity to name the disk pool. The system assigns the pool number. The user is not given an option to select a number for the pool. Figure 6-10 illustrates the cautions displayed to the user regarding IASP pool number assignment.



New Disk Pool - Add to Disk Pool 33

Select the disk units to add to disk pool 33 (Demo). Disk pool 33 (Demo) is unprotected.

This is disk pool 1 of 1 disk pools that you selected to work with.

To add disk units to disk pool 33 (Demo), select the disk unit or units and click Add. To remove, s or units and click Remove.

Available disk units:

Selected disk units:

Independent pool creation

- user names pool
- system responsible for assigning pool number

V5R1

- incorrect number ***may*** be displayed
  - ***during creation process***
  - ***correct number is assigned at end of process***

Figure 6-10 IASP pool number is assigned by the system

Important: In V5R1, there are cases when the GUI does not display the correct number when creating the pool.

There is no method for the system to unmistakably know the number until the pool is actually created. The present implementation is to not show any number for the independent pool until it is fully created. This is a known restriction that is addressed in a later release.

6.8.3 Selecting the disk units to add to a pool

When creating the IASP using the Operations Navigator interface, the GUI wizard looks at all the disks in the system and assigns them a *suitability rank* for use with an IASP. The suitability rank indicates how appropriate the disk unit is to add to the specific disk pool. The lower the number, the more appropriate the match.

A number 0-99 indicates the ranking of the most suitable disk, as illustrated in Table 6-2. Select the disk units for inclusion in the IASP based on the rankings.

Table 6-2 Independent ASP rankings

Rank	Description
2	Disk pool and disk are nonswitchable.
53	Disk pool and disk are switchable.
100-199	Valid disks, but not most suitable.
102	Disk pool and disk are nonswitchable, but disk can be made switchable.
151	Disk pool and disk are switchable. The disk's switchable entity is in the same Cluster Resource Group.

Rank	Description
153	Disk pool is switchable. Disk is switchable, but its switchable entity has no disks in any disk pool in the disk pool's CRG.
200-299	Unlikely disks, and should not be used unless absolutely necessary.
202	Disk pool is nonswitchable. Disk is switchable and its switchable entity already has disks in a switchable disk pool.
300-399	Invalid disks. Cannot be used in the Disk Pool.

Find a full list of rankings in the help text associated with the Operations Navigator function.

6.8.4 ASP unit numbering

The system selects the ASP number for IASPs. The numbers range from 33 through 99. Use ASP numbers range from 2 through 32.

Across an IASP device domain, numbers assigned to ASPs are unique and conform to these standards:

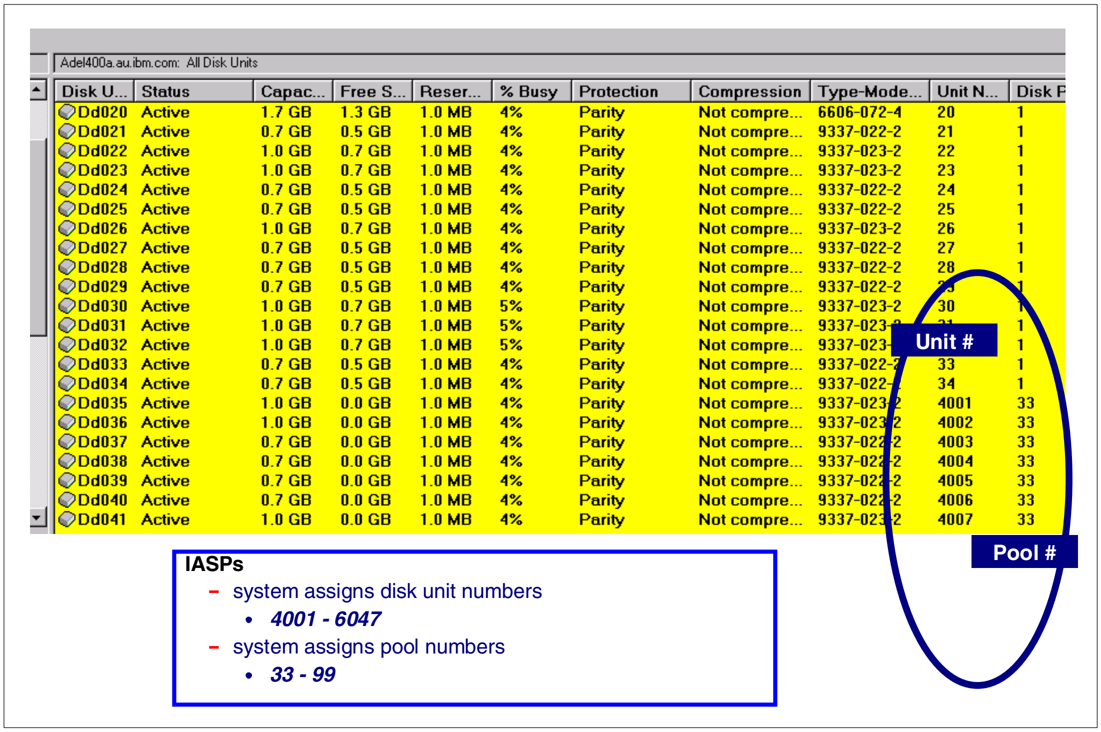
- User ASP numbers range from two through 2047.
- System ASP is number 1.
- IASPs are assigned numbers in the range of 4001 through 6047.

The separate ranges provide additional disk unit numbers with these limits:

- Each node in the IASP device domain can have up to 2047 units in its system and user ASPs.
- The entire IASP device can have up to 2047 units in IASPs.

The ASP numbers are in separate ranges to avoid a conflict when user ASPs are added to an IASP device domain within a node.

Figure 6-11 shows the IASP pool number and disk unit numbers assigned by the system to an IASP. The disk units in this example are in the range 4001 through 4007. The IASP number is 33.



Adel400a.au.ibm.com: All Disk Units

Disk U...	Status	Capac...	Free S...	Reser...	% Busy	Protection	Compression	Type-Mode...	Unit N...	Disk P
Dd020	Active	1.7 GB	1.3 GB	1.0 MB	4%	Parity	Not compre...	6606-072-4	20	1
Dd021	Active	0.7 GB	0.5 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	21	1
Dd022	Active	1.0 GB	0.7 GB	1.0 MB	4%	Parity	Not compre...	9337-023-2	22	1
Dd023	Active	1.0 GB	0.7 GB	1.0 MB	4%	Parity	Not compre...	9337-023-2	23	1
Dd024	Active	0.7 GB	0.5 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	24	1
Dd025	Active	0.7 GB	0.5 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	25	1
Dd026	Active	1.0 GB	0.7 GB	1.0 MB	4%	Parity	Not compre...	9337-023-2	26	1
Dd027	Active	0.7 GB	0.5 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	27	1
Dd028	Active	0.7 GB	0.5 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	28	1
Dd029	Active	0.7 GB	0.5 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	29	1
Dd030	Active	1.0 GB	0.7 GB	1.0 MB	5%	Parity	Not compre...	9337-023-2	30	1
Dd031	Active	1.0 GB	0.7 GB	1.0 MB	5%	Parity	Not compre...	9337-023-2	31	1
Dd032	Active	1.0 GB	0.7 GB	1.0 MB	5%	Parity	Not compre...	9337-023-	[illegible]	1
Dd033	Active	0.7 GB	0.5 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	33	1
Dd034	Active	0.7 GB	0.5 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	34	1
Dd035	Active	1.0 GB	0.0 GB	1.0 MB	4%	Parity	Not compre...	9337-023-2	4001	33
Dd036	Active	1.0 GB	0.0 GB	1.0 MB	4%	Parity	Not compre...	9337-023-2	4002	33
Dd037	Active	0.7 GB	0.0 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	4003	33
Dd038	Active	0.7 GB	0.0 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	4004	33
Dd039	Active	0.7 GB	0.0 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	4005	33
Dd040	Active	0.7 GB	0.0 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	4006	33
Dd041	Active	1.0 GB	0.0 GB	1.0 MB	4%	Parity	Not compre...	9337-023-2	4007	33

Unit #

Pool #

IASPs

- system assigns disk unit numbers
  - ***4001 - 6047***
- system assigns pool numbers
  - ***33 - 99***

Figure 6-11 Unit numbering for IASPs and disks in the IASP

6.8.5 Selecting All Disk Units to add to a pool

An alternate path available to create IASPs on a system is to select the *All Disk Units* path.

The *Select* path described in 6.8.1, “Creating a new disk pool” on page 114, is the path most users take to create an independent pool. In this path, there is no option to create more than one disk pool.

For a new system or new environment, select the **All Disk Units** path to work with more than one pool in the same operation. Follow this alternative GUI path to create disk pools.

6.9 IASP and security

This section discusses the characteristics of user profiles and authorization lists as they relate to the security of an IASP.

6.9.1 User profiles and IASPs

User profile information is stored in the system ASP. Each user profile object is an object type of *USRPRF.

Copies of *USRPRF objects are not in any independent pool. However, some user profile information must be maintained on the IASP itself.

Each object in an IASP requires this user profile information:

- The owner of the object
- The primary group of the object
- The private authority entries for the object

See Figure 6-12 for an illustration.

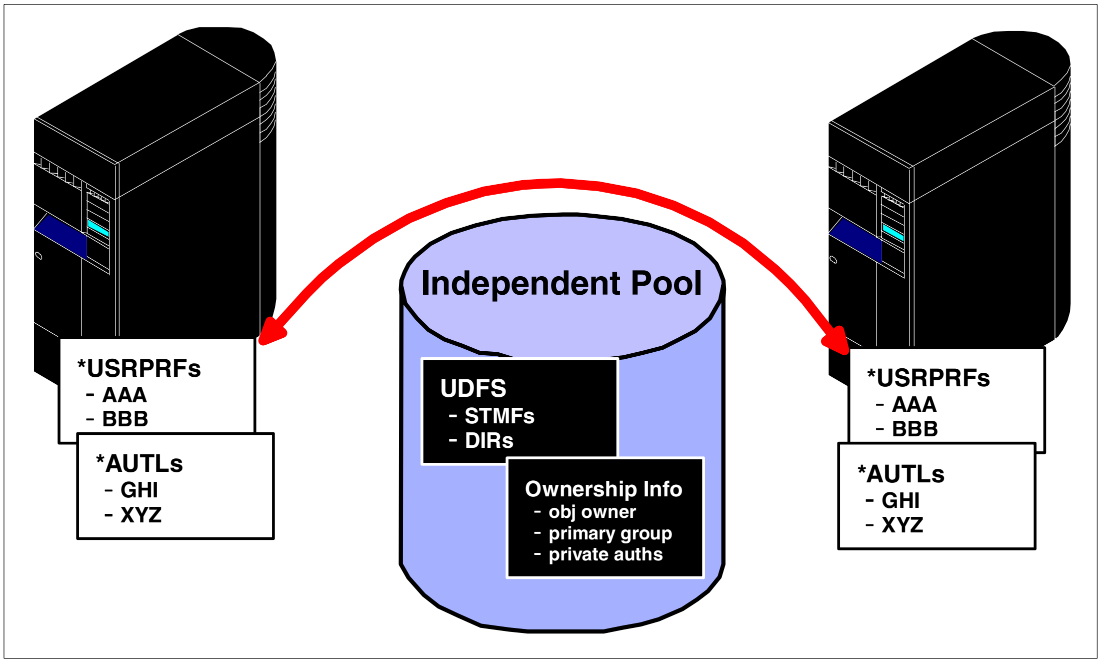


Figure 6-12 IASPs and user profiles

Additional storage (above that consumed by objects) is required for these system security structures. This is necessary to make the independent ASP self-contained. These structures consume a maximum of 10% of the disk space within the IASP. The percentage varies and depends on the size of the objects referenced by these structures.

SLIC is responsible for storing, retrieving, changing, or deleting the authority information stored on an IASP. OS/400 security interfaces accommodate the SLIC changes.

When creating independent disk pools in a multiple (clustered) system environment, it is assumed the content of any user profiles is synchronized across the cluster in which the user profile exists.

6.9.2 Accounting for space owned

The total storage attributed to an individual user profile is the sum of the storage used:

- In each online IASP
- By the user in the local system user ASPs

Use the Display User Profile (DSPUSRPRF) CL command to display the total value of this storage.

If an IASP is brought online, some user profiles can have the value of their total storage increase dramatically. In some cases, the total amount of storage to which they are authorized is exceeded.

To prevent disruptions to a system, the limit for the maximum amount of auxiliary storage that can be allocated by a user profile is not enforced when bringing an IASP online. That is, if bringing an IASP online causes a user profile to exceed its storage limit, the storage limit is allowed to exceed the specified amount. Subsequent requests for additional storage cause a “User Profile Storage Limit Exceeded” exception.

Consider this example to help illustrate this concept. An IASP starts in an offline status. User AAA has a maximum storage (MAXSTG) value of 1000. 500 objects are in the local system and user pools. Similarly, user BBB has a MAXSTG of 1500 with 1200 in use.

The IASP contains objects HJK and RST, which represent 950 units of total storage.

Figure 6-13 illustrates this situation.

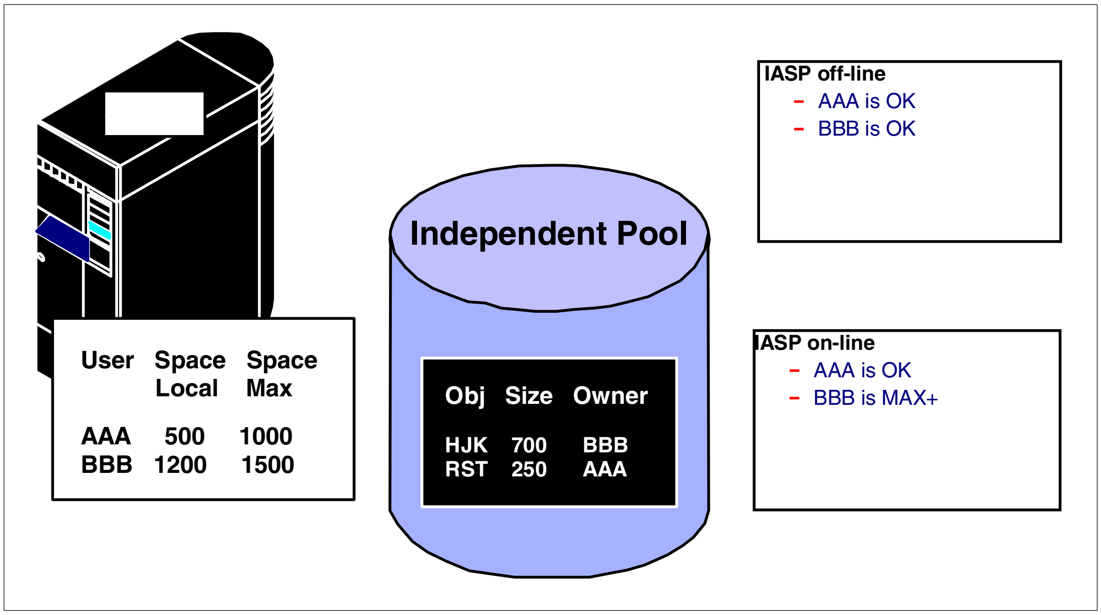


Figure 6-13 Accounting for space used

Then the IASP is brought online.

User profile AAA is updated to a value of 750, which is still below its MAXSTG value of 1000. However, BBB now exceeds its MAXSTG value of 1500. That is, the addition of HJK brings the value owned to 1900.

User profile BBB is not allowed to own any additional objects.

6.9.3 IASP and authorization lists

Authorization lists are used to secure access to objects in a system, irrespective of ASP boundaries.

An authorization list (similar to a user profile) is implemented as a distributed object. That is, it can secure objects within the system ASP, user ASPs, and any IASPs.

There remains one copy of each authorization list (*AUTL) object in the system ASP. Copies of *AUTL objects are not kept on IASPs. However, when an object on an IASP is secured by an authorization list, the name of that list is kept on the IASP.

This concept is illustrated in Figure 6-14. Authorization lists AUTL111 and AUTL222 are stored in the IASP, because the objects HJK and RST are secured by these lists. There is no reference on the IASP itself to the authorization list AUTL333 because the object LHR is not stored on the IASP.

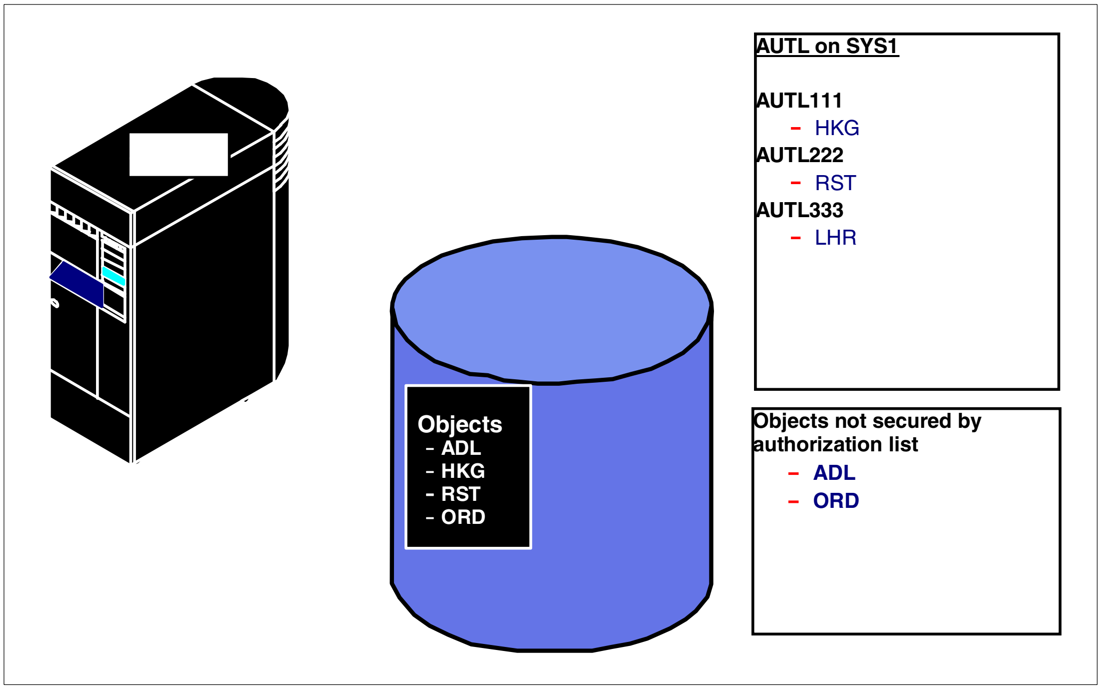


Figure 6-14 IASPs and authorization lists

As with user profiles, the Licensed Internal Code (LIC) is responsible for handling this function. The user interface does not change for storing, retrieving, changing, or deleting the authorization list information on an IASP.

Switching IASPs and *AUTLs

The effect of switching an IASP when at *AUTL is implemented as described in this section. For this description, these are the assumptions, activities, and results:

- **Assumptions**
  - An IASP has been switched from one system to another.
  - A user attempts to access an object in the IASP.
  - The object in the IASP is secured by an authorization list.
- **Activity**
  - LIC accesses the IASP and retrieves the name of the authorization list. LIC attempts to connect to the authorization list on the newly attached system.
  - If an authorization list with that name is found on this system, authority checking proceeds as normal.
  - If an authorization list with that name is not found on this system, the attempted access to the object is handled the same as if the authorization list is damaged.
- **Results**
  - Any authority check that is satisfied before the authority list is tested works as expected.
  - When an authority check tries to access the missing authorization list, the result is an “object damaged” failure.
  - The Display Authority (DSPAUT) command displays blanks for the authorization list name.

- The user can change the authorization list name using the Change Authority (CHGAUT) command.

6.10 System operations with IASPs

An IASP can be brought online and made active at any time during IPL or after the system is IPLed. The IASP is brought online by the system during IPL if the device description of the IASP specifies *YES for the Online at IPL parameter (ONLINE(*YES)). Once the IASP is online and active, the objects within the IASP are accessible and usable.

> **Important:** An IASP can go offline and yet the remainder of the system remains functional. However, once an IASP is offline, the objects that it contains are no longer visible to, accessible from, or usable by the system where it was previously online and active.

An IASP that is online on one system can be varied offline by that system. This can be done explicitly by user request or it can be done implicitly by system termination.

6.10.1 IASP overflow

There is a difference between ASPs and independent ASPs as it relates to the action when an ASP overflows. That is:

- An overflow of a basic ASP occurs when the ASP fills. The excess data spills into the system ASP.
- IASPs are designed so that they cannot overflow; otherwise, they would not be switchable.

When there is insufficient space available for a requested allocation of storage in an IASP, Auxiliary Storage Management either signals a new exception (ASP Storage Limit Exceeded) or returns an error code.

6.10.2 Switching IASPs between nodes

The process of switching IASPs between nodes within a domain is performed by cluster software. If the IASP is to be switched between LPAR partitions, then the disk units can be either internal or external components.

If the IASPs are to be switched between machines (that is, separate CECs), then the disk units must be external, such as those in a switchable tower or a Storage Area Network (SAN).

A logical SPCN related question to ask is: When the SPCN node of a switchable tower has a failure, what happens to the tower? The answer is that Cluster Management handles this. Even though the SPCN node of the switchable tower fails, Cluster Management instructs the tower to remain on. The tower does not power down. Cluster Management then switches the tower to the other system and the other system then has access to the tower.

6.10.3 Reclaim Storage and IASPs

With the introduction of IASPs comes the capability to run Reclaim Storage (RCLSTG) on an IASP while the rest of the system keeps running. This implies that multiple IASP RCLSTG processes can execute simultaneously.

V5R1 functional changes to the RCLSTG command added to support IASPs are:

- *SYSBAS values

  If the *SYSBAS value is specified for the ASP device, the Reclaim Storage operation runs as it does on systems prior to V5R1. The reclaim operation is performed on the system and on traditional user-defined ASPs.

  If the value specified is an ASP device name, then that ASP is reclaimed.

- Reclaim Storage for an ASP device (that is, an IASP) can be run without the system being in restricted state. Multiple jobs can be submitted, each performing RCLSTG on a different ASP device. Multiple ASP devices can be reclaimed in parallel.

Note: Reclaiming an auxiliary storage pool device requires that there are no active users of the ASP device that is the subject of the reclaim.

6.11 Positioning cluster middleware solutions with IASPs

With the introduction of IASPs on the iSeries server, the questions arise:

- Are cluster middleware solutions needed?
- Can an IASP solution support high availability for the business?

To understand the fit of IASPs, it is important to understand the requirements of the business. Use the information in Table 6-3 to compare the needs of the business to the functions available with a replication solution and that offered by an IASP switched disk solution.

Table 6-3 Comparison of cluster middleware replication and IASP switched disk

	Cluster middleware replication solution	**IASP switched disk solution**
Number of systems in cluster	Tens of systems	Two systems
Single Point of Failure	None	Disk subsystem
Cost factors	Additional disk capacity. Replication software.	Switchable I/O expansion tower
Performance factors	Replication overhead	Negligible
Typical failover time	hours	Around 15 minutes
Typical switchover time	Around 30 minutes	Around 5 minutes
Real time coverage	Objects journaled	Objects in IASP
Objects supported	A very large set	IFS only (except QSYS.LIB)
Geographic dispersion	Unlimited	Limited distance to attach (250 meters maximum)
Disaster Recovery Protection	Yes	No
Concurrent backup	Yes	No
Setup	Replication environment. What to replicate.	Requires Clustering support on the two systems.

While it is true that IASPs provide a new means of high availability on the iSeries server, it does not replace business partner solutions. Independent ASPs co-exists with these solutions.

Consider these characteristics of IASP and replication solutions:

- Replication solutions provide geographical dispersal of the data.

  The production and backup systems can be several thousand kilometers apart. This is an important factor for effective disaster recovery.
- With an IASP solution, the systems must be within 250 meters of each other because of the limitations of the HSL Loop. With some V5R1 configurations, the distance is limited to 15 meters.
- Customers who simply want high availability at a lower cost can use independent ASPs without adding disks for the backup system.
- A replication solution provides switchover capability between two systems.

  The level of monitoring between systems can be primitive. However, the approach for switchover and failover is complex. Monitoring is performed at a high level.

  In comparison, using switchable IASP with clustering provides a means to handle a complex requirement in a relatively simple way. The heartbeat monitoring that is implemented with IBM clustering is very sophisticated. Once properly setup, the switchover or failover to the backup system can be nearly seamless.

6.12 iSeries IASP and other platform solutions

Non-iSeries platforms (such as UNIX and Windows NT systems) have offered switched disk solutions for high availability for several years. Given earlier limitations and the exposures of a switched disk solution, these platforms have now added replication technology to their suite of high availability offerings.

For example, the high availability product on the IBM @server pSeries (RS/6000) servers is High Availability Cluster Multi-Processing for AIX (HACMP). This product has been around for many years.

HACMP provides the switched disk solution for high availability. A newer product called *HACMP GSO* is now available to provide replication technology for the pSeries customer.

iSeries enabled these two functions in the reverse order. The iSeries offerings started with a replication technology solution and have now added switched disk technology with V5R1.

The important point to note is that both replication and switched disk solutions are needed. Indeed, they must co-exist. This is evident for non-iSeries as well as iSeries platforms.

6.13 Reference material

You can find useful sources of further information on independent ASPs and related topics at:

- InfoCenter: `http://publib.boulder.ibm.com/html/as400/infocenter.html`
- High Availability: `http://www.iseries.ibm.com/ha`
- *iSeries Backup and Recovery*, SC41-5304

Operations Navigator independent ASP GUI

Configuring a cluster, configuring a hardware group (a resilient device CRG), and performing a switch are done from the primary node of a cluster (the central server). The central server is used to activate the cluster objects, but the disk pool (the IASP) is managed from the specific system.

There are two types of IASPs: standalone and switchable. This chapter describes the steps necessary to create each type of independent auxiliary storage pool (IASP) using the GUI that is part of Operations Navigator in OS/400 V5R1. This method creates a system group that is coupled tightly with the cluster.

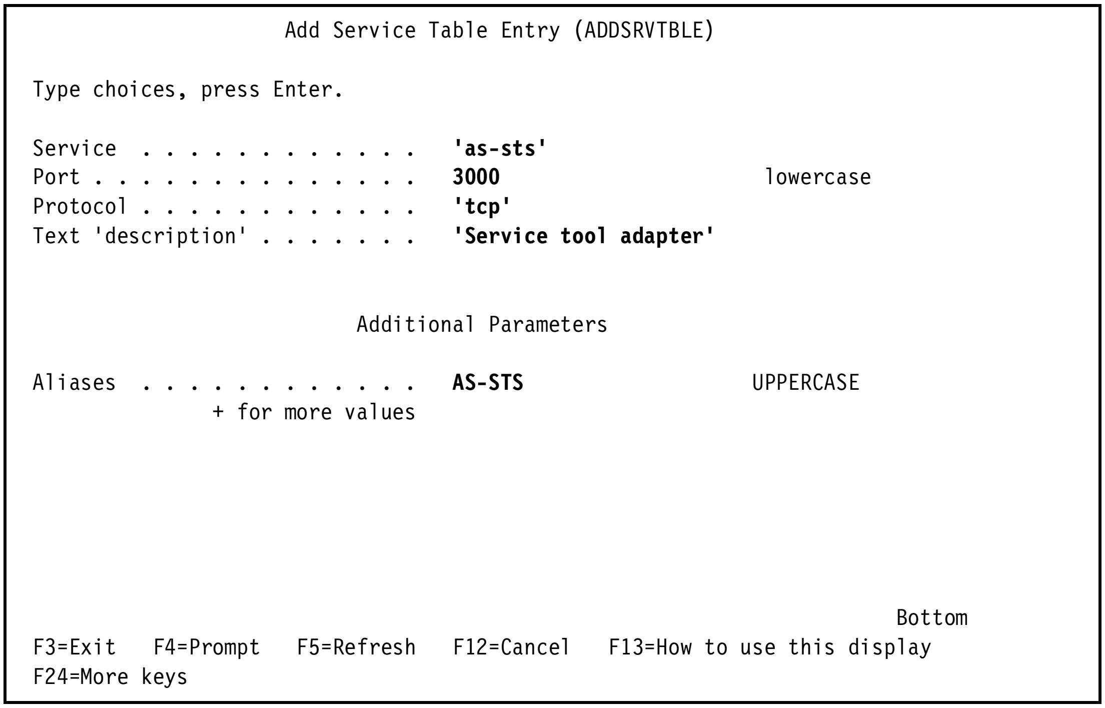
7.1 Prerequisite steps to create an IASP

Before you can create any IASPs, there are some prerequisite steps within Operations Navigator that you must complete. These steps have to do with configuring the adapter used by Service Tools on the iSeries server and setting up Management Central so that disk devices can be managed.

7.1.1 Configuring the Service Tools adapter on the iSeries server

A service table entry is required to use the disk functions of Operations Navigator. The steps to add a service table entry and submit a job to reclaim TCP on the iSeries servers that is to be used with the IASP are illustrated in this section.

Use the Add Service Table Entry (ADDSRVTBE) command and fill in the parameters as shown in Figure 7-1 to add a service table entry named *'as-sts'*.



```
                      Add Service Table Entry (ADDSRVTBLE)

 Type choices, press Enter.

 Service  . . . . . . . . . . .   'as-sts'
 Port . . . . . . . . . . . . .   3000                     lowercase
 Protocol . . . . . . . . . . .   'tcp'
 Text 'description' . . . . . .   'Service tool adapter'


                            Additional Parameters

 Aliases  . . . . . . . . . . .   AS-STS                  UPPERCASE
               + for more values






                                                                    Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 7-1 Adding a service table entry on the iSeries server

Press Enter to continue.

Tip: Enter the value for the Service and Protocol parameters:

- In lowercase letters
- Enclosed in single quotes

If the entries do not follow these rules, and Additional Parameters is selected, the Protocol value becomes uppercase letters, which causes an error to occur.

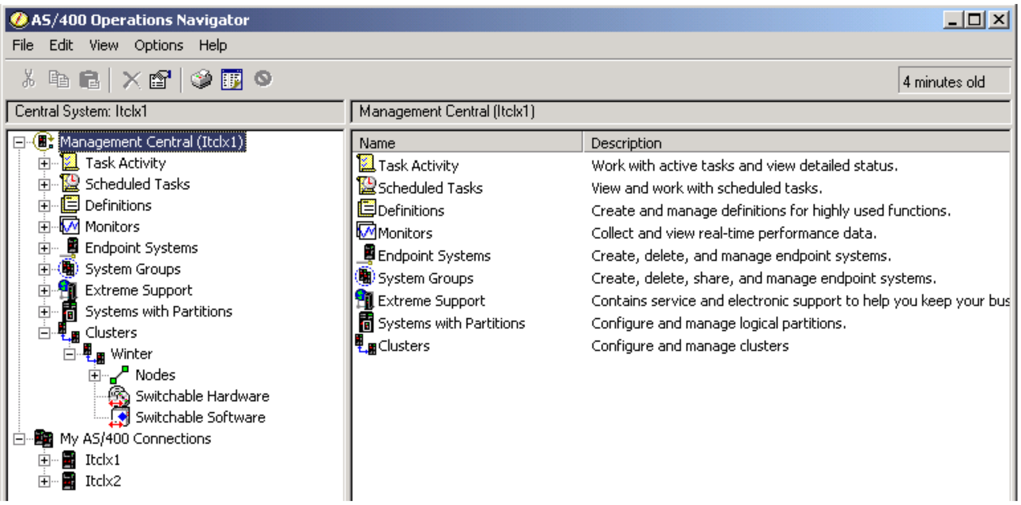
The service table entry added in this step does not become effective until TCP is ended and started again. Use the ENDTCP and STRTCP commands to end and start TCP.

> **Tip:** The connection to the system is broken when ENDTCP is issued from a TCP-connected terminal. Also, commands (including STRTCP) cannot be entered from any TCP-connected terminal until connectivity with the system is re-established.
>
> Enter the Start TCP and End TCP commands from the system console or from a green-screen terminal.

7.1.2 Setting up Management Central

Configure the Management Central function in Operations Navigator. The necessary setup steps are illustrated in this section:

1. Start with Operations Navigator as shown in Figure 7-2.

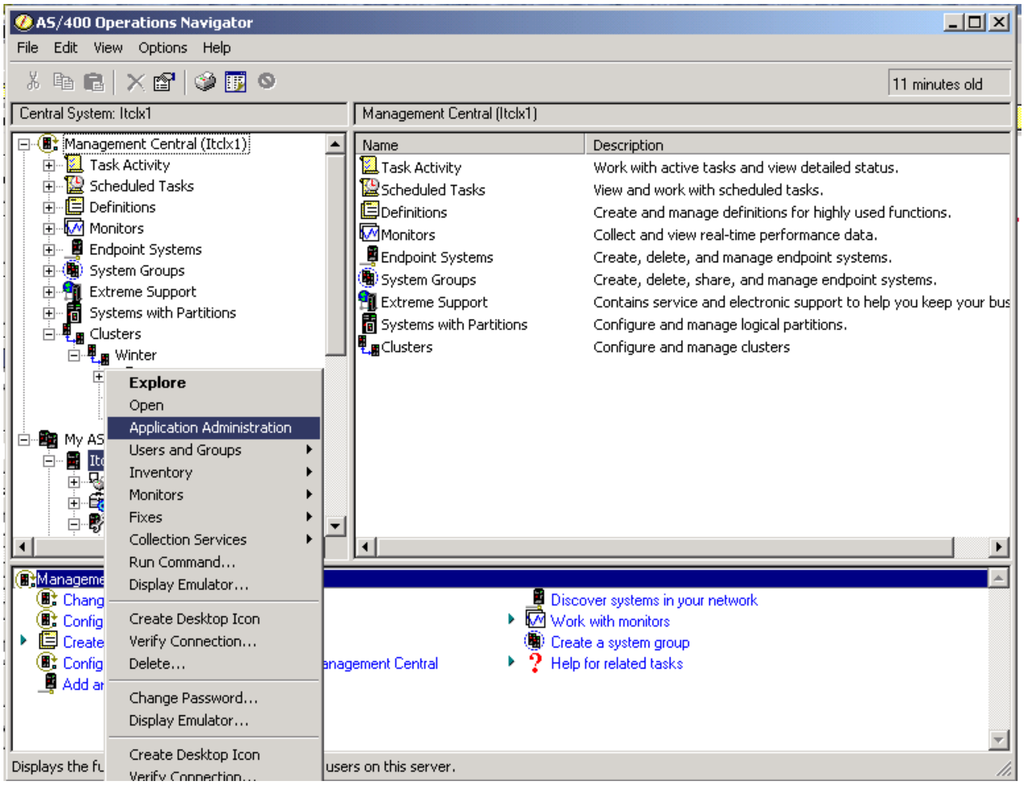


- Management Central (Itclx1)
  - Task Activity
  - Scheduled Tasks
  - Definitions
  - Monitors
  - Endpoint Systems
  - System Groups
  - Extreme Support
  - Systems with Partitions
  - Clusters
    - Winter
      - Nodes
      - Switchable Hardware
      - Switchable Software
- My AS/400 Connections
  - Itclx1
  - Itclx2

Name	Description
Task Activity	Work with active tasks and view detailed status.
Scheduled Tasks	View and work with scheduled tasks.
Definitions	Create and manage definitions for highly used functions.
Monitors	Collect and view real-time performance data.
Endpoint Systems	Create, delete, and manage endpoint systems.
System Groups	Create, delete, share, and manage endpoint systems.
Extreme Support	Contains service and electronic support to help you keep your bus
Systems with Partitions	Configure and manage logical partitions.
Clusters	Configure and manage clusters

Figure 7-2 View of Operations Navigator

2. Right-click the system name. Select **Application Administration** from the drop-down list as illustrated in Figure 7-3.



Name	Description
Task Activity	Work with active tasks and view detailed status.
Scheduled Tasks	View and work with scheduled tasks.
Definitions	Create and manage definitions for highly used functions.
Monitors	Collect and view real-time performance data.
Endpoint Systems	Create, delete, and manage endpoint systems.
System Groups	Create, delete, share, and manage endpoint systems.
Extreme Support	Contains service and electronic support to help you keep your bus
Systems with Partitions	Configure and manage logical partitions.
Clusters	Configure and manage clusters

Figure 7-3 Access Application Administration in Operations Navigator

3. If a window is displayed, as illustrated in Figure 7-4, click **OK** to continue.

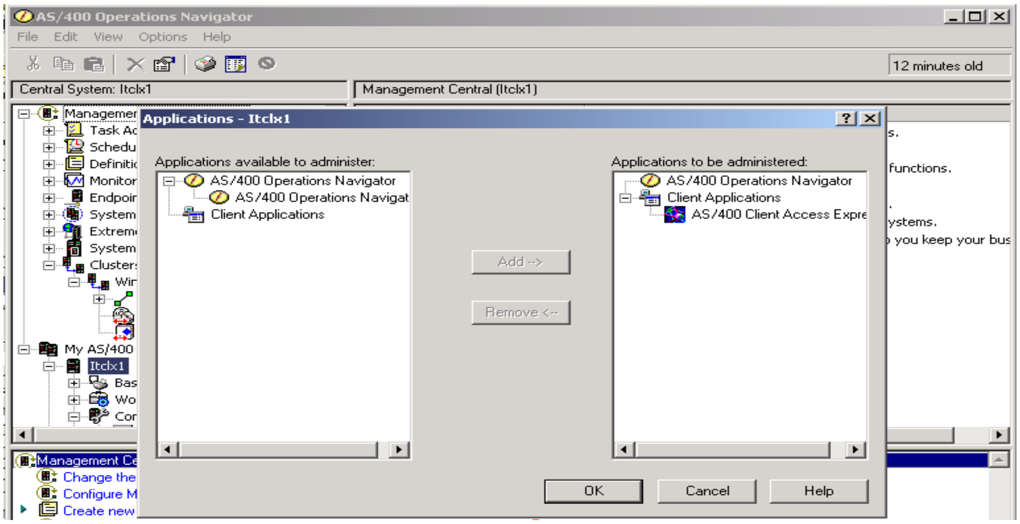
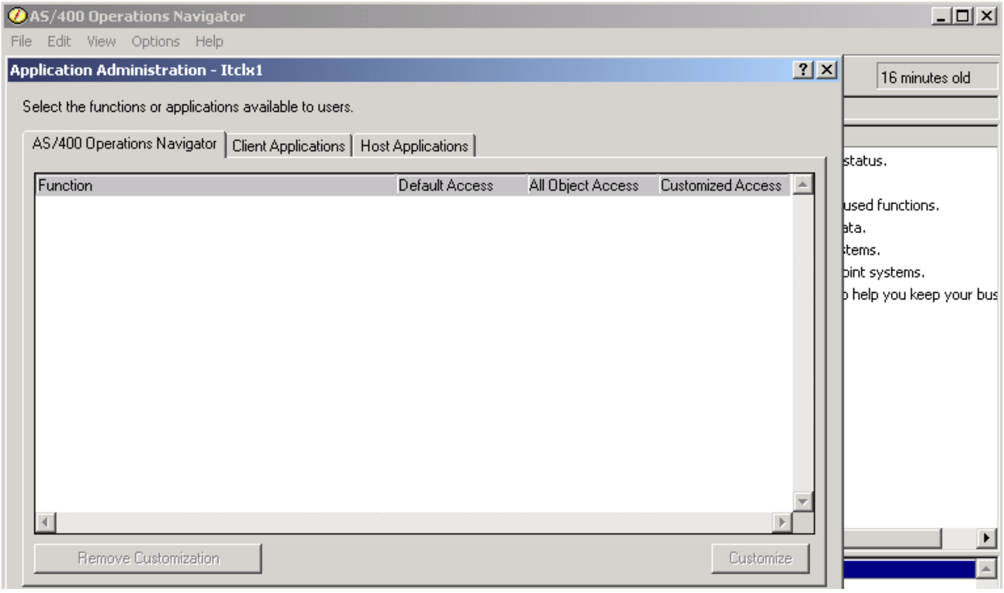


Figure 7-4 Application Administration window

4. Select the **Host Applications** tab on this window, as shown in Figure 7-5.

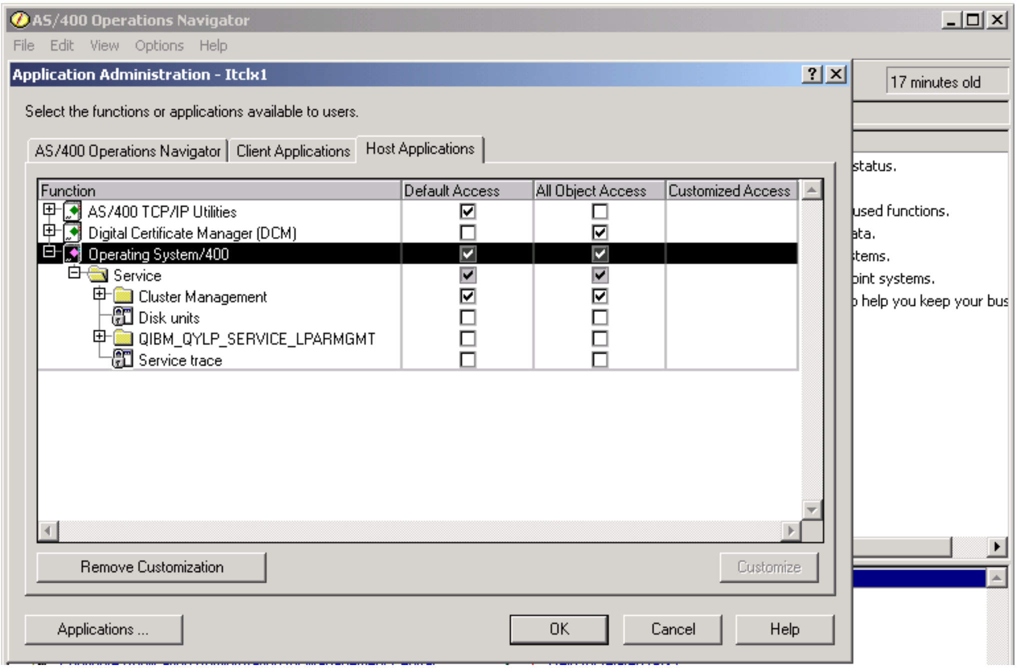


Select the functions or applications available to users.

Function	Default Access	All Object Access	Customized Access

Figure 7-5 Host Applications tab

5. On the Host Applications page (Figure 7-6), expand the **Operating System/400** and **Service** folders until Disk units is displayed. Select the **Default Access** and the **All Object Access** squares next to the Disk units. Click **OK**.



Select the functions or applications available to users.

Function	Default Access	All Object Access	Customized Access
AS/400 TCP/IP Utilities	☑	□	
Digital Certificate Manager (DCM)	□	☑	
Operating System/400	☑	☑	
Service	☑	☑	
Cluster Management	☑	☑	
Disk units	□	□	
QIBM_QYLP_SERVICE_LPARMGMT	□	□	
Service trace	□	□	

Figure 7-6 Enabling disk unit access

Now the Disk Units function can be used from within Operations Navigator.

7.2 The independent auxiliary storage pool GUI

IASPs can be standalone or switchable. The steps to create an IASP depend on the type of IASP to be created. The steps are described in this section.

7.2.1 Creating a standalone independent auxiliary storage pool

Figure 7-7 illustrates a non-switchable independent ASP called IASP1. It is created on a system named AS24.

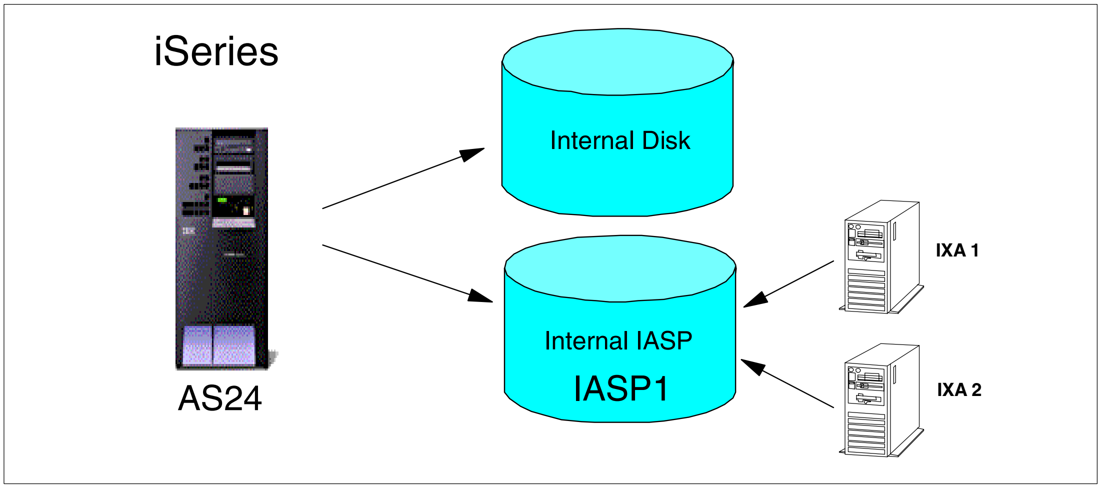


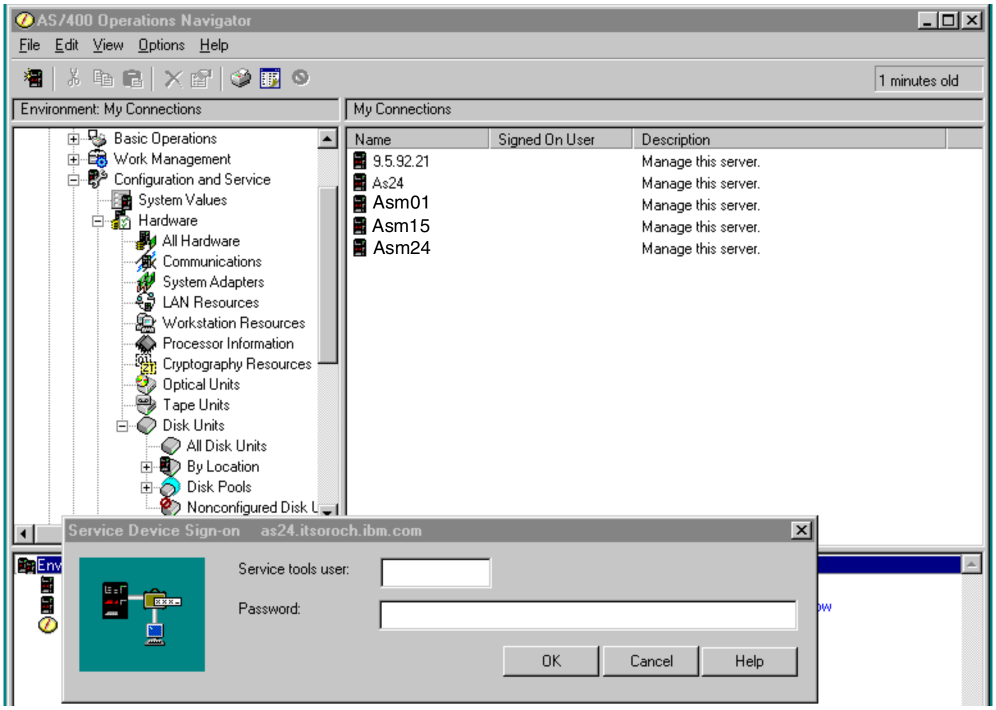
Figure 7-7 Non-switchable independent ASP with Integrated xSeries Adapters (IXA) attached

IASPs are useful to segregate parts of the Integrated File System. They also can be used to store data for two Integrated xSeries Adapters. If one of the Integrated xSeries Adapters fails, the other Integrated xSeries Adapter can be configured to use the data on the IASP.

A standalone IASP is created using the Disk Pools function of Operations Navigator. To arrive at the Disk Pools section within Operations Navigator, click **My Connection-> System Name-> Configuration and Service-> Hardware-> Disk Units-> Disk Pools**.

The steps to create a standalone IASP are:

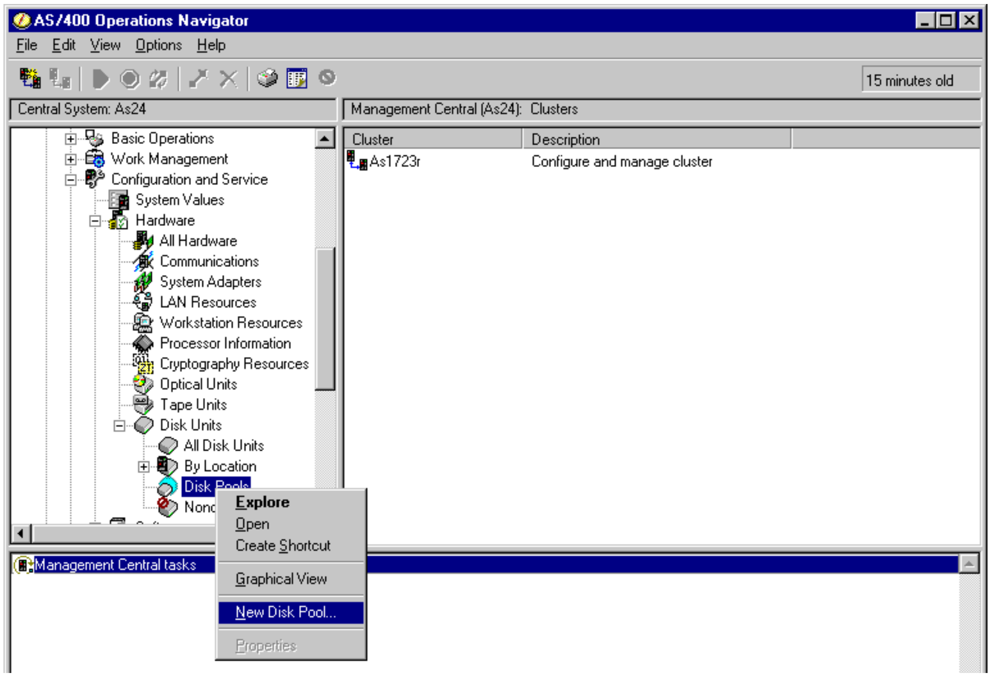
1. Sign on to DST using the Service Tools user ID and password, as illustrated in Figure 7-8.



Name	Signed On User	Description
9.5.92.21		Manage this server.
As24		Manage this server.
Asm01		Manage this server.
Asm15		Manage this server.
Asm24		Manage this server.

Figure 7-8 Signing on to DST

2. Right-click **Disk Pools** and select **New Disk Pool** to invoke the New Disk Pools wizard, as illustrated in Figure 7-9.



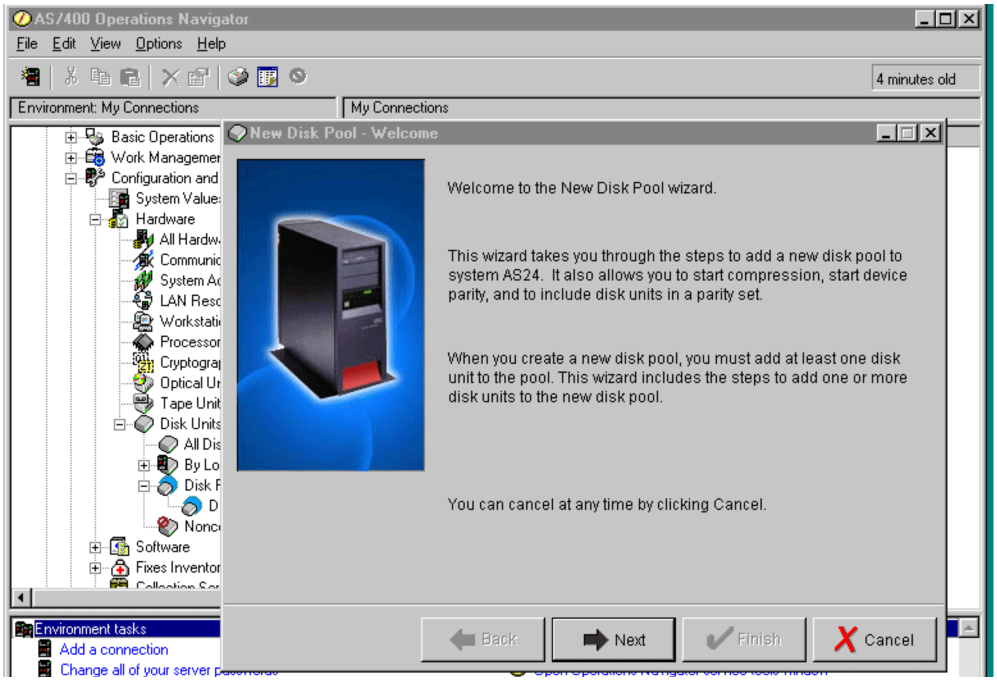
Cluster	Description
As1723r	Configure and manage cluster

Figure 7-9 New Disk Pool

3. Once the IASP is created, vary it on.

   This example next copies some PC files to the IASP to illustrate that they are now usable from the IASP.

The display in Figure 7-10 illustrates the New Disk Pool wizard's Welcome panel.



Welcome to the New Disk Pool wizard.

This wizard takes you through the steps to add a new disk pool to system AS24. It also allows you to start compression, start device parity, and to include disk units in a parity set.

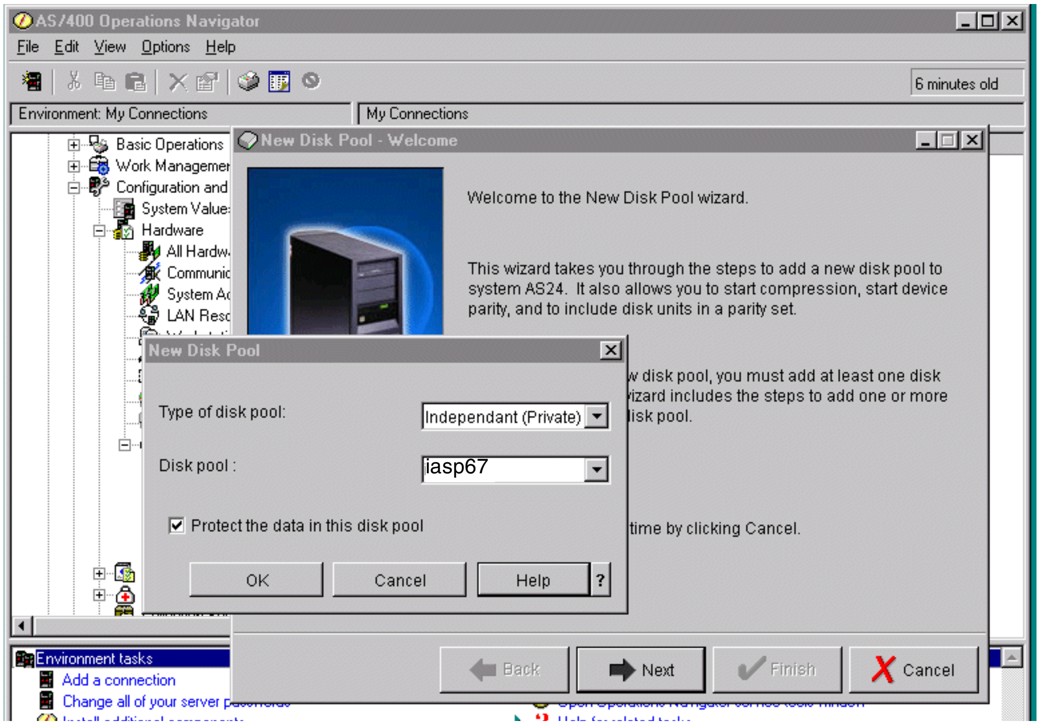
When you create a new disk pool, you must add at least one disk unit to the pool. This wizard includes the steps to add one or more disk units to the new disk pool.

You can cancel at any time by clicking Cancel.

Figure 7-10 Welcome panel of the New Disk Pool Wizard

4. When the New Disk Pool wizard starts, click **Next**. A Specify Disk Pool Type panel is presented, as illustrated in Figure 7-11.

5. Use the pull-down menu and select the type of disk pool. Enter a name for the pool. If any disk units are to be included in a device parity set, select the **Protect the data in this disk pool** check box.



Welcome to the New Disk Pool wizard.

This wizard takes you through the steps to add a new disk pool to system AS24. It also allows you to start compression, start device parity, and to include disk units in a parity set.

w disk pool, you must add at least one disk izard includes the steps to add one or more isk pool.

time by clicking Cancel.

Type of disk pool: Independant (Private)

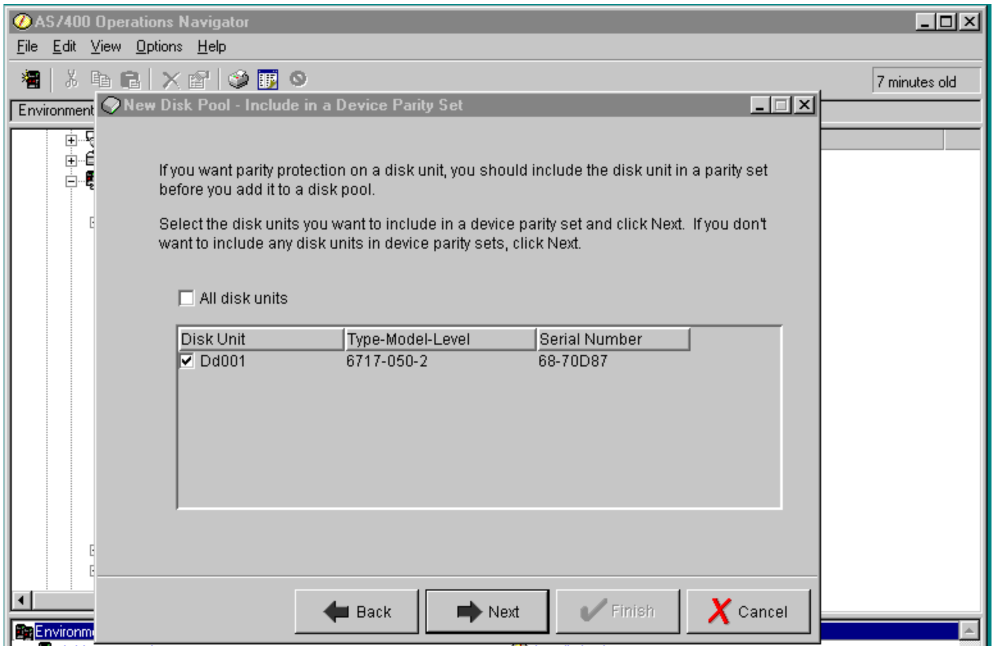
Disk pool : iasp67

☑ Protect the data in this disk pool

Figure 7-11 Specifying the disk pool type

In this example, we select **Independent** for disk pool type, **iasp67** as the pool name, and select **Protect the data in this disk pool**. The pool name is used to distinguish it from other ASPs that are configured on the system.

6. Select the disk units to be protected, as illustrated in Figure 7-12, or select **All disk units**.



AS/400 Operations Navigator
File Edit View Options Help
7 minutes old
Environment
New Disk Pool - Include in a Device Parity Set

If you want parity protection on a disk unit, you should include the disk unit in a parity set before you add it to a disk pool.

Select the disk units you want to include in a device parity set and click Next. If you don't want to include any disk units in device parity sets, click Next.

□ All disk units

Disk Unit	Type-Model-Level	Serial Number
☑ Dd001	6717-050-2	68-70D87

Figure 7-12 Protecting a disk pool

If the disk is not added at this time to a parity set, it can be added later using Operations Navigator.

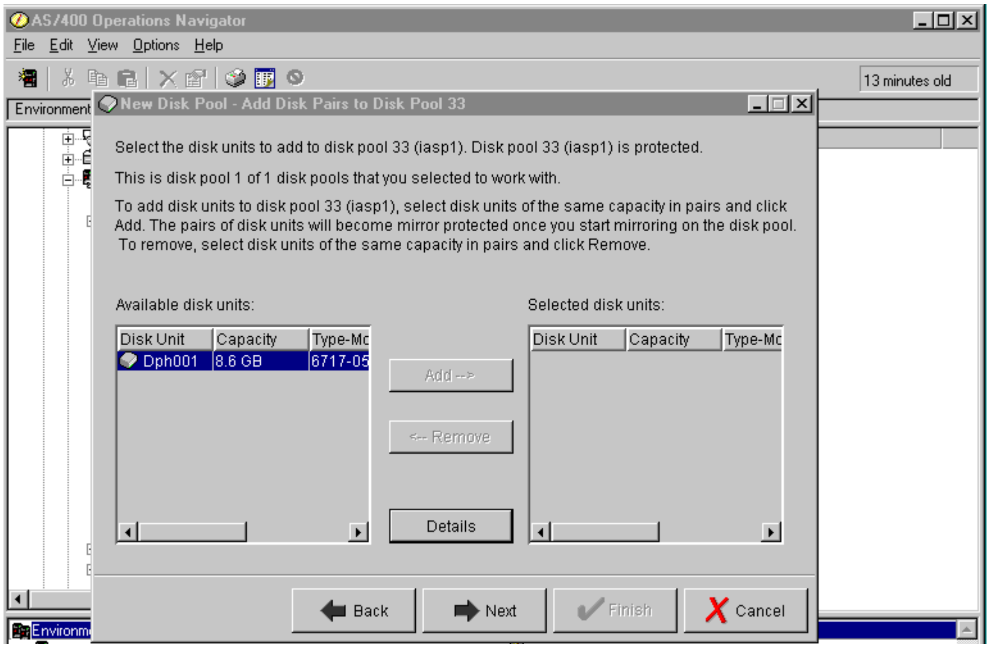
a. Click **OK** to return to the New Disk Pool Welcome panel.

b. Click Next to continue.

The next display lists the disk units available to add to a pool. This display (as illustrated in Figure 7-13) only appears if there are disks that are eligible to be included in an existing device parity set. If so, an option is offered to start parity on the selected units. Similarly, if the eligible disk units support disk compression, an option is offered to start compression on the selected disk units.

Note: Depending on the cluster, some disk units may not be displayed even though they are non-configured.

7. Highlight the disk units to be in the pool. Click **Add**, as illustrated in Figure 7-13.



AS/400 Operations Navigator
File Edit View Options Help
13 minutes old
Environment
New Disk Pool - Add Disk Pairs to Disk Pool 33

Select the disk units to add to disk pool 33 (iasp1). Disk pool 33 (iasp1) is protected.

This is disk pool 1 of 1 disk pools that you selected to work with.

To add disk units to disk pool 33 (iasp1), select disk units of the same capacity in pairs and click Add. The pairs of disk units will become mirror protected once you start mirroring on the disk pool. To remove, select disk units of the same capacity in pairs and click Remove.

Available disk units:

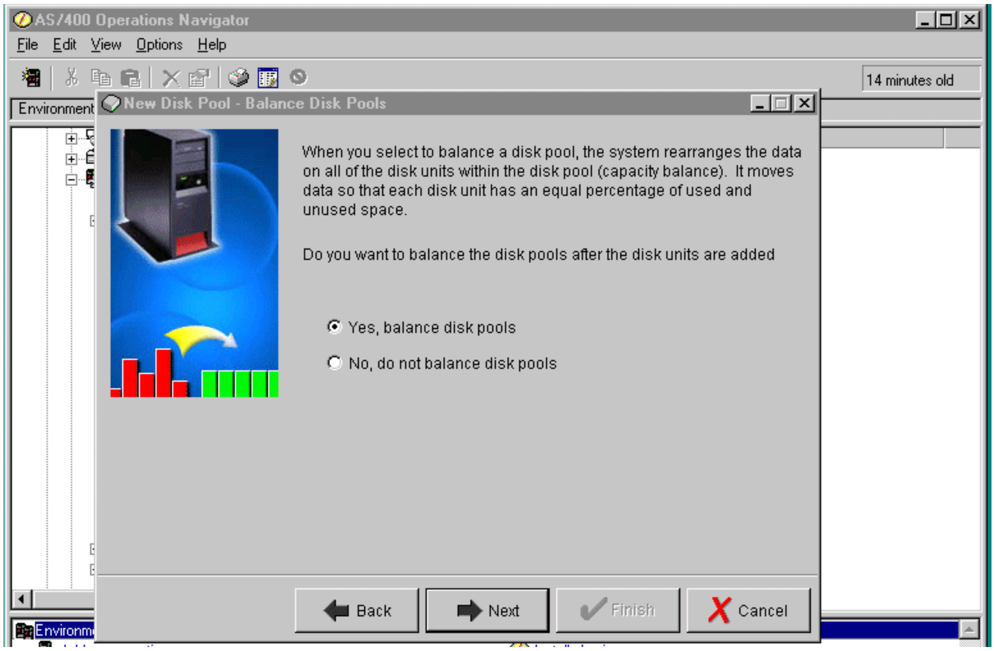
Disk Unit	Capacity	Type-Mc
Dph001	8.6 GB	6717-05

Selected disk units:

Disk Unit	Capacity	Type-Mc

Figure 7-13 Selecting the disk units

8. When finished, click **Next.**
9. To have the system automatically balance the new disk pool, select the **Yes, balance disk pools** option, as illustrated in Figure 7-14.



AS/400 Operations Navigator
File Edit View Options Help
14 minutes old
Environment
New Disk Pool - Balance Disk Pools

When you select to balance a disk pool, the system rearranges the data on all of the disk units within the disk pool (capacity balance). It moves data so that each disk unit has an equal percentage of used and unused space.

Do you want to balance the disk pools after the disk units are added

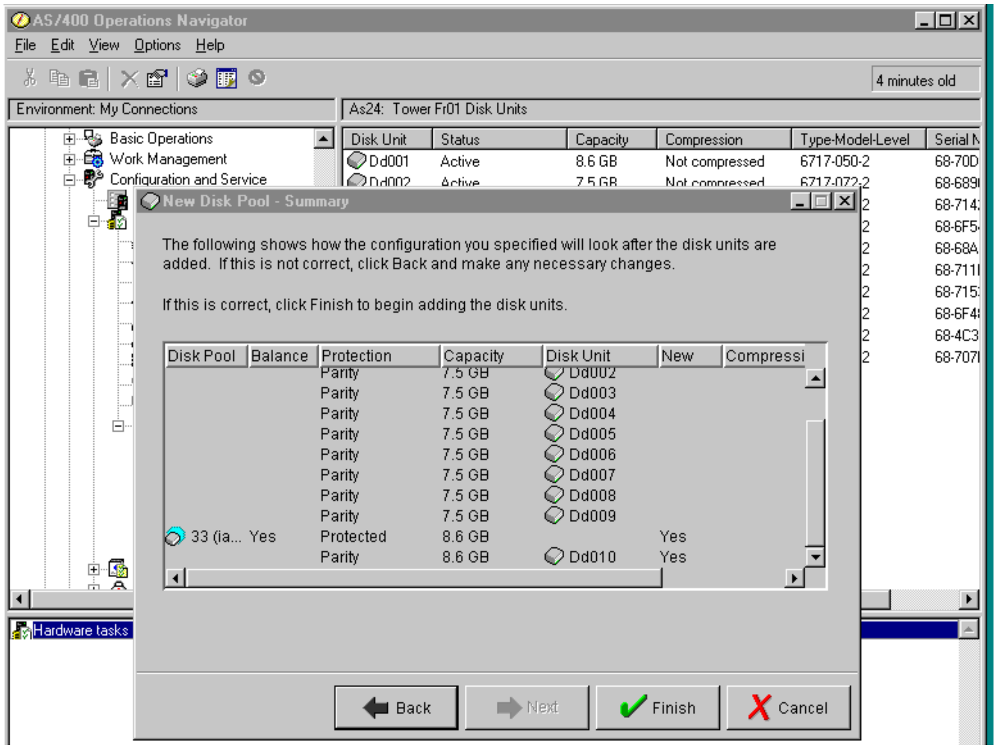
● Yes, balance disk pools

○ No, do not balance disk pools

Figure 7-14 Balancing the disk pools

Note: If the disk pool is comprised of all new (empty) disk units, select the No option.

Click **Next**. A summary dialog (Figure 7-15) of the disk pools on the system appears.



The following shows how the configuration you specified will look after the disk units are added. If this is not correct, click Back and make any necessary changes.

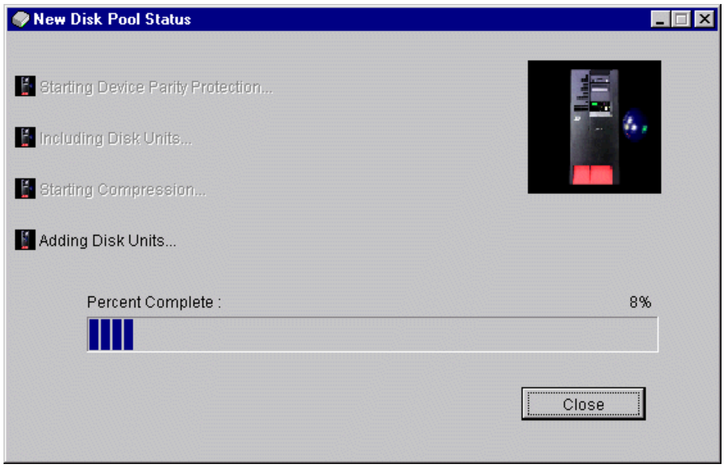
If this is correct, click Finish to begin adding the disk units.

Disk Pool	Balance	Protection	Capacity	Disk Unit	New	Compressi
		Parity	7.5 GB	Dd002		
		Parity	7.5 GB	Dd003		
		Parity	7.5 GB	Dd004		
		Parity	7.5 GB	Dd005		
		Parity	7.5 GB	Dd006		
		Parity	7.5 GB	Dd007		
		Parity	7.5 GB	Dd008		
		Parity	7.5 GB	Dd009		
33 (ia...	Yes	Protected	8.6 GB		Yes	
		Parity	8.6 GB	Dd010	Yes	

Figure 7-15 Summary of the disk configuration

10. Click **Finish** to complete the configuration.

A moving bar (Figure 7-16) indicates the progress of the operation.



Starting Device Parity Protection...

Including Disk Units...

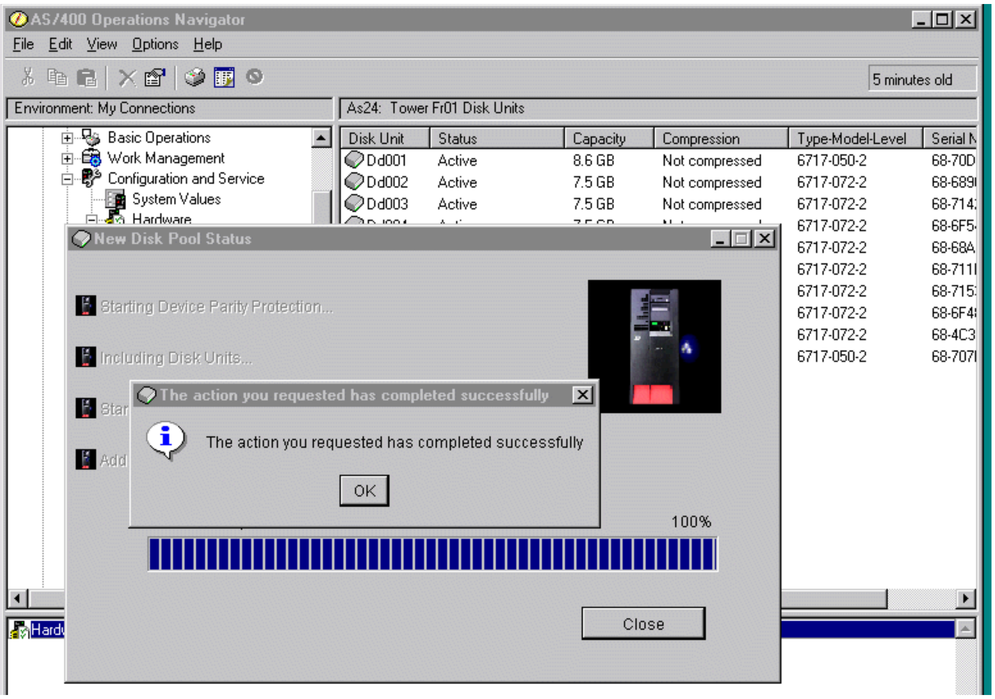
Starting Compression...

Adding Disk Units...

Percent Complete : 8%

Figure 7-16 New Disk Pool Status window

When complete, a successful completion message (Figure 7-17) is displayed.


AS/400 Operations Navigator
File Edit View Options Help
5 minutes old
Environment: My Connections
Basic Operations
Work Management
Configuration and Service
System Values
Hardware
Hardw

As24: Tower Fr01 Disk Units

Disk Unit	Status	Capacity	Compression	Type-Model-Level	Serial N
Dd001	Active	8.6 GB	Not compressed	6717-050-2	68-70D
Dd002	Active	7.5 GB	Not compressed	6717-072-2	68-689
Dd003	Active	7.5 GB	Not compressed	6717-072-2	68-714
				6717-072-2	68-6F5
				6717-072-2	68-68A
				6717-072-2	68-711
				6717-072-2	68-715
				6717-072-2	68-6F4
				6717-072-2	68-4C3
				6717-050-2	68-707

New Disk Pool Status

Starting Device Parity Protection...

Including Disk Units...

Star

Add

The action you requested has completed successfully

The action you requested has completed successfully

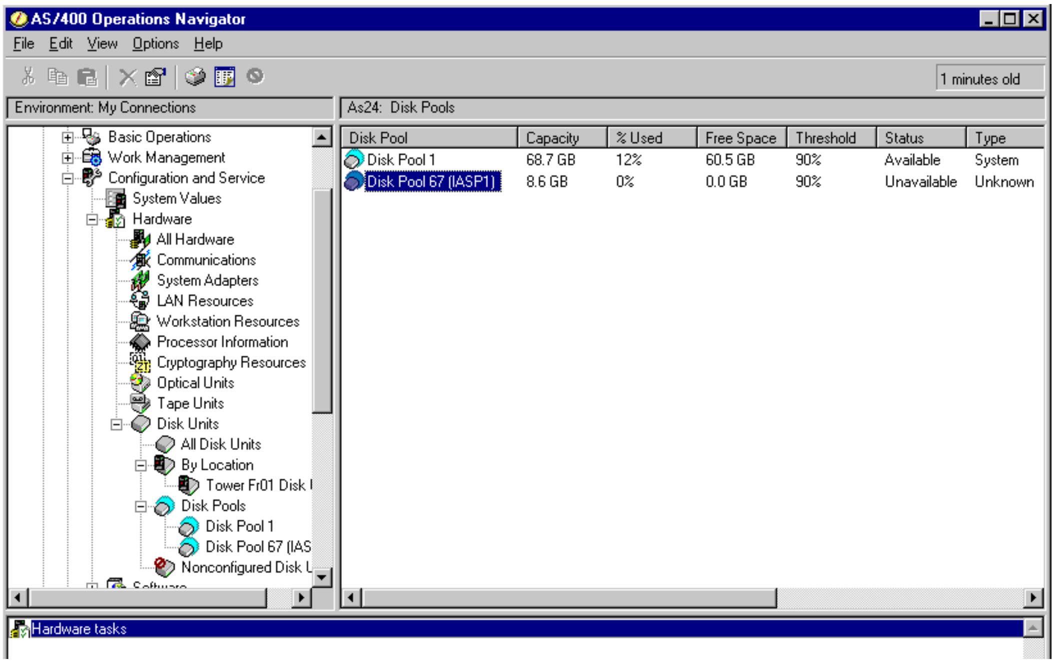
OK

100%

Close

Figure 7-17 Disk pool created

The new IASP is now visible to Operations Navigator (see Figure 7-18).



As24: Disk Pools

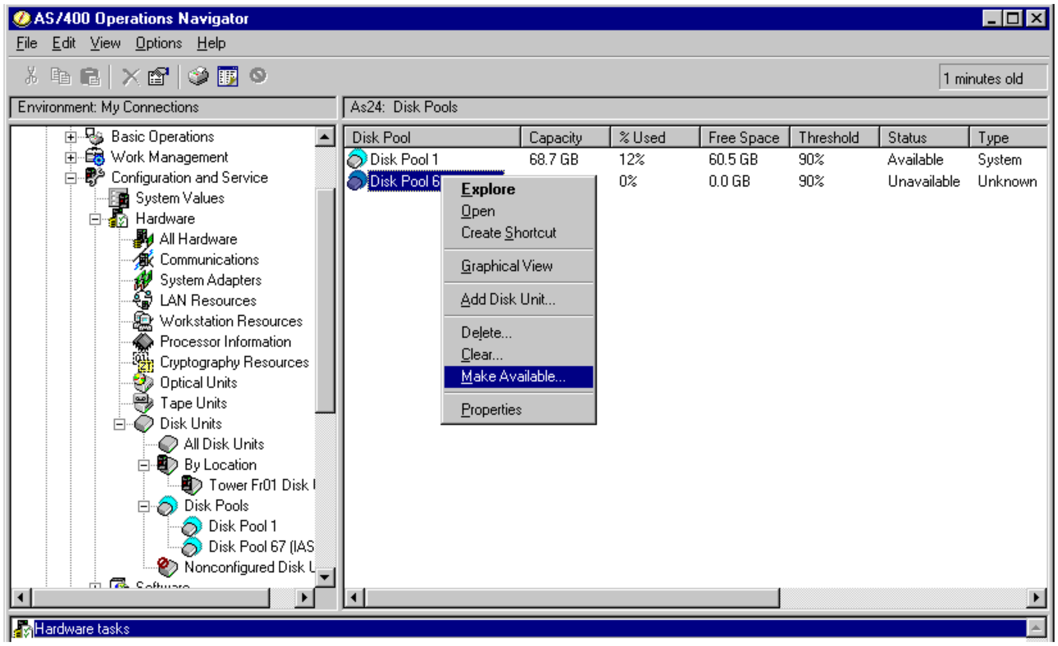
Disk Pool	Capacity	% Used	Free Space	Threshold	Status	Type
Disk Pool 1	68.7 GB	12%	60.5 GB	90%	Available	System
Disk Pool 67 (IASP1)	8.6 GB	0%	0.0 GB	90%	Unavailable	Unknown

Figure 7-18 Disk pool visible to Operations Navigator

The disk pool has been created. The status is *Unavailable*.

Note: Operations Navigator uses the term “unavailable” to indicate “varied off”.

11. Before the resources within a new disk pool (IASP) can be used by the system (node), the IASP must be varied on. To make the IASP available (vary it on), right-click the pool and select **Make Available** (Figure 7-19).

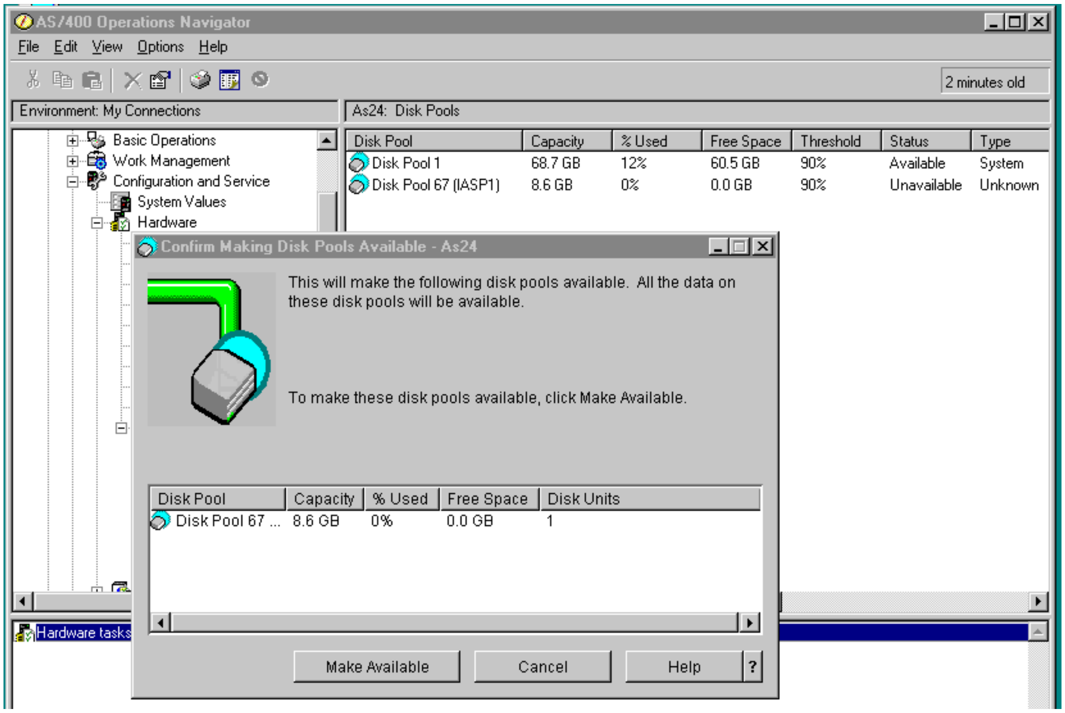


As24: Disk Pools

Disk Pool	Capacity	% Used	Free Space	Threshold	Status	Type
Disk Pool 1	68.7 GB	12%	60.5 GB	90%	Available	System
Disk Pool 6		0%	0.0 GB	90%	Unavailable	Unknown

Figure 7-19 Varying on the IASP

12. Confirm the choice by clicking the **Make Available** button for the independent auxiliary storage pool (Figure 7-20).


AS/400 Operations Navigator
File Edit View Options Help
2 minutes old
Environment: My Connections
Basic Operations
Work Management
Configuration and Service
System Values
Hardware
Hardware tasks
Confirm Making Disk Pools Available - As24
Make Available
Cancel
Help
?

As24: Disk Pools

Disk Pool	Capacity	% Used	Free Space	Threshold	Status	Type
Disk Pool 1	68.7 GB	12%	60.5 GB	90%	Available	System
Disk Pool 67 (IASP1)	8.6 GB	0%	0.0 GB	90%	Unavailable	Unknown

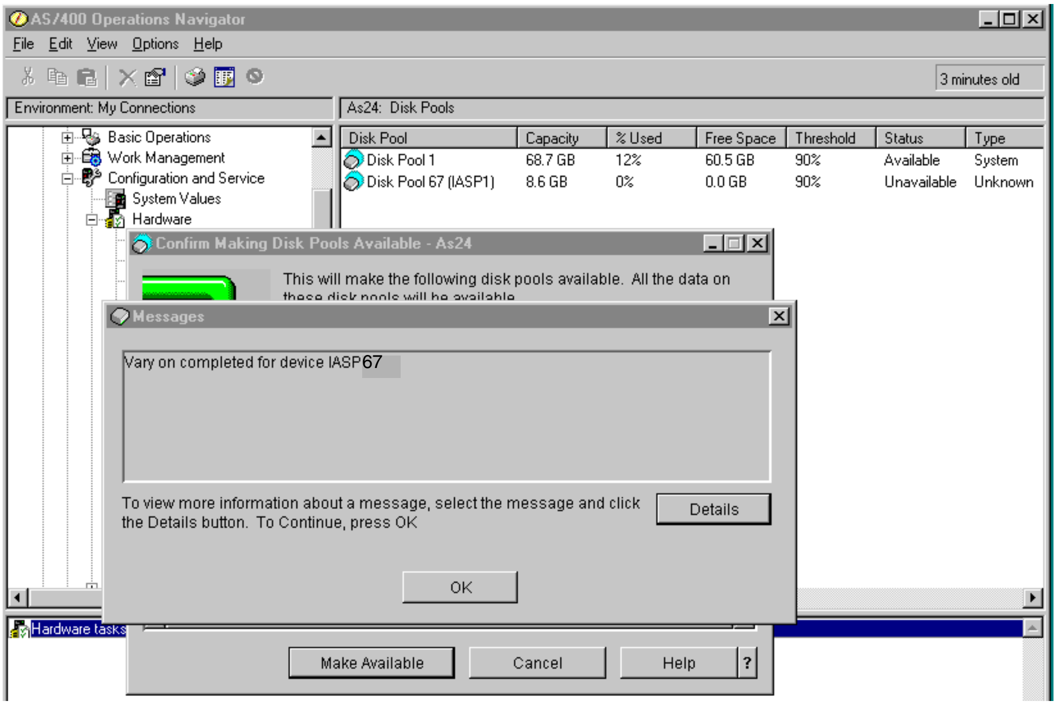
This will make the following disk pools available. All the data on these disk pools will be available.

To make these disk pools available, click Make Available.

Disk Pool	Capacity	% Used	Free Space	Disk Units
Disk Pool 67 ...	8.6 GB	0%	0.0 GB	1

Figure 7-20 Confirming to make the disk pool available

13. A successful vary on completion message is displayed after the vary on operation completes, as illustrated in Figure 7-21.


AS/400 Operations Navigator
File Edit View Options Help
3 minutes old
Environment: My Connections
Basic Operations
Work Management
Configuration and Service
System Values
Hardware
Hardware tasks
As24: Disk Pools
Confirm Making Disk Pools Available - As24
Messages
Details
OK
Make Available
Cancel
Help
?

Disk Pool	Capacity	% Used	Free Space	Threshold	Status	Type
Disk Pool 1	68.7 GB	12%	60.5 GB	90%	Available	System
Disk Pool 67 (IASP1)	8.6 GB	0%	0.0 GB	90%	Unavailable	Unknown

This will make the following disk pools available. All the data on these disk pools will be available.

Vary on completed for device IASP67

To view more information about a message, select the message and click the Details button. To Continue, press OK

Figure 7-21 Vary on completed

14. Click **OK**.

When an IASP device is created, the system automatically generates a user-defined file system (UDFS) with the same name as the device (IASP) name. Place the files to be used by the application environment into the default UDFS (root/iasp-name/mydir).

Directories can be created under the IASP directory. When the IASP is made available (varied on), the default UDFS file system is automatically mounted in the root directory of the system’s Integrated Files System (IFS). The root/dev/iasp-name ASP and UDFS iasp-name are then visible through the Operations Navigator IFS view.

15. Expand the **Integrated File System** option in Operations Navigator.

16. Expand the items under the **Root** directory. The new IASP is the last item listed, as illustrated in Figure 7-22.

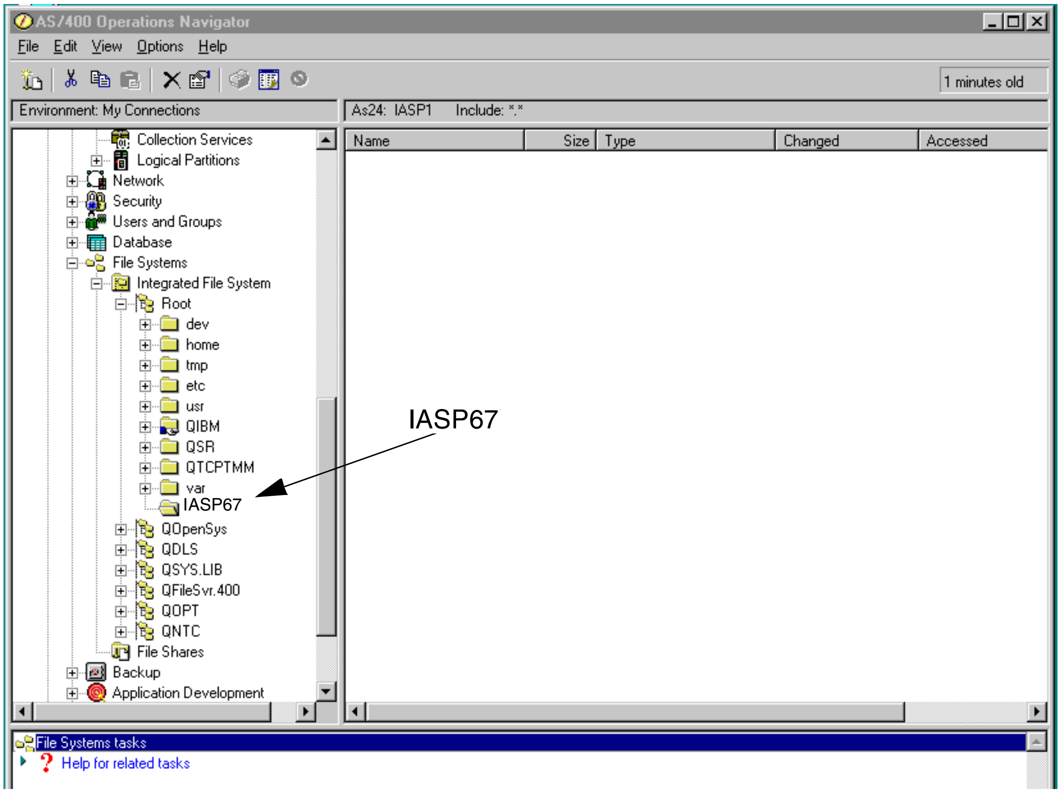
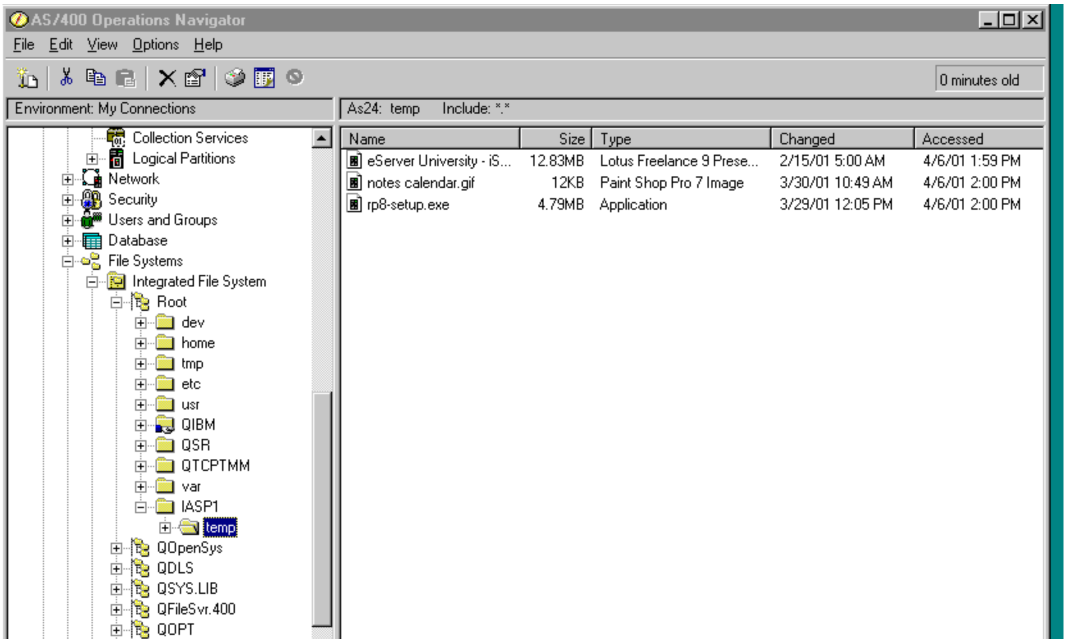


Figure 7-22 IASP visible on the Integrated File System

17. Copy some PC files to the new IASP. Figure 7-23 illustrates the results of a copy.

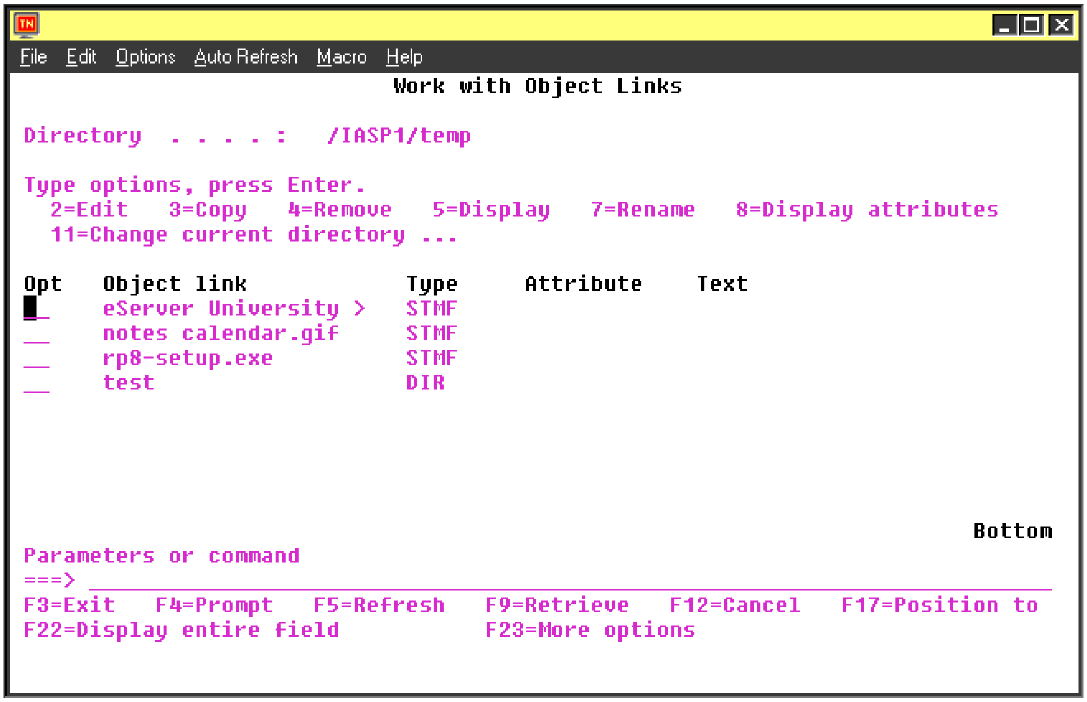


Name	Size	Type	Changed	Accessed
eServer University - iS...	12.83MB	Lotus Freelance 9 Prese...	2/15/01 5:00 AM	4/6/01 1:59 PM
notes calendar.gif	12KB	Paint Shop Pro 7 Image	3/30/01 10:49 AM	4/6/01 2:00 PM
rp8-setup.exe	4.79MB	Application	3/29/01 12:05 PM	4/6/01 2:00 PM

Figure 7-23 PC files on the IASP

Directories can also be created under the IASP directory.

18. Use the Work with Object Links (WRKLNK) command to display the Integrated File System on a green screen, as illustrated in Figure 7-24.



```
                          Work with Object Links

Directory  . . . . :   /IASP1/temp

Type options, press Enter.
  2=Edit   3=Copy   4=Remove   5=Display   7=Rename   8=Display attributes
  11=Change current directory ...

Opt   Object link            Type     Attribute    Text
__    eServer University >   STMF
__    notes calendar.gif     STMF
__    rp8-setup.exe          STMF
__    test                   DIR

                                                                       Bottom
Parameters or command
===> ____________________________________________________________________________
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F12=Cancel   F17=Position to
F22=Display entire field           F23=More options
```

Figure 7-24 Displaying IFS files with WRKLNK

The files copied to the IFS are listed.

7.2.2 Creating a switchable independent auxiliary storage pool

In this section, a switchable IASP named IASP2 is created under a cluster named AS1723R. AS1723R is comprised of two nodes: System AS01B and System AS01C. The Switchable Hardware Group (Device CRG) under which this IASP is managed is called Switch2. This relationship is illustrated in Figure 7-25.

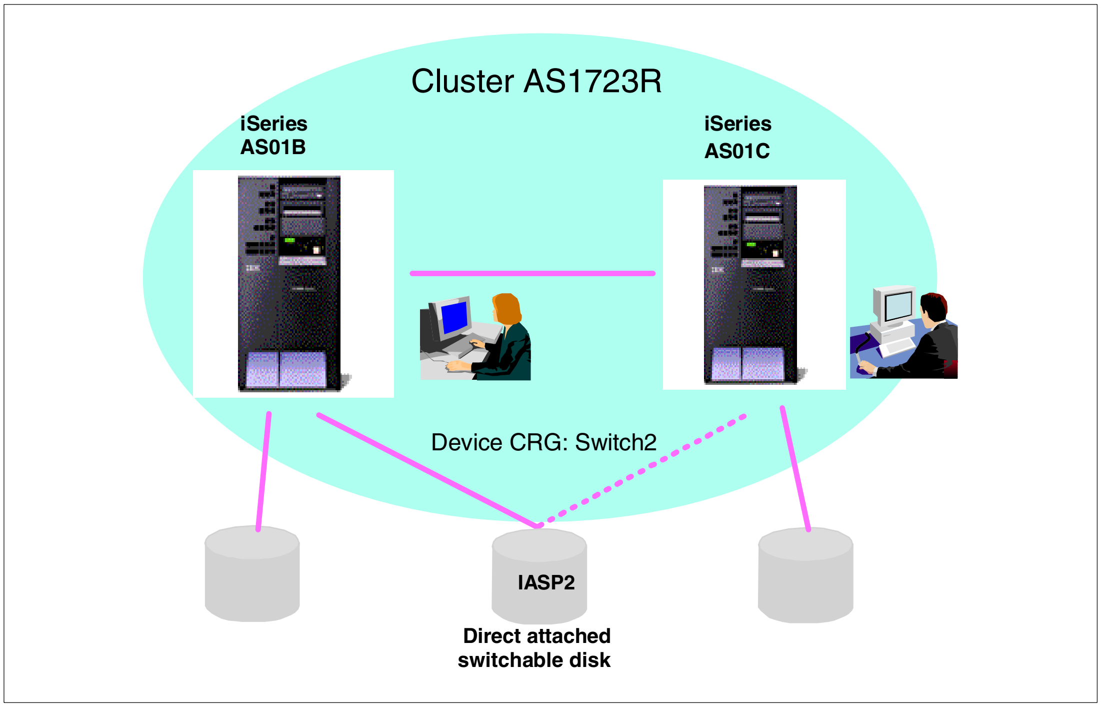


Figure 7-25 Simple two-node cluster using a switchable IASP

Figure 7-25 shows that IASP2 is attached to both system AS01B and system AS01C. It can be switched between both systems. The IASP is typically an expansion tower, or it can be a shared IOP in an LPAR scenario.

In the event that one of the systems fails, the switchable IASP automatically fails over to the other system. Any Integrated File System object stored in the IASP is then visible and usable to the other system. In addition, it is possible to perform a manual switch of the IASP from one system to the other by varying it off on one system and varying it on the other.

Before a switchable IASP is created, a cluster must be defined.

Here are some key points to remember when creating a switchable IASP:

- Create the cluster and ensure that it can be started and stopped.
- Use the New Group option under the Switchable Hardware section of Operations Navigator to create a switchable IASP. The switchable IASP is created as a part of creating the switchable hardware group. To access the switchable hardware section, select **Management Central-> Clusters-> Cluster Name-> Switchable Hardware**.

  Right-click the **Switchable Hardware icon** and select the **New Group** option to start the Switchable Hardware Group wizard. Follow the steps to create the switchable hardware group depending on whether the IASP exists:

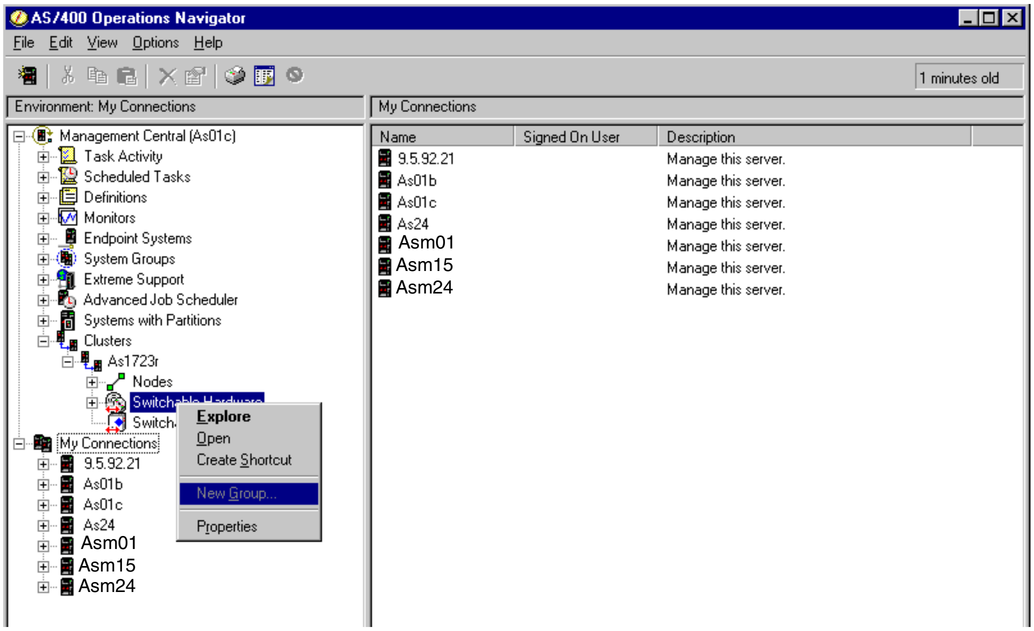
  – To specify an existing IASP, enter the IASP device description name when prompted for the disk pool name. A confirmation and warning message is then displayed to indicate that the name specified already exists. Click **Yes** to use the existing IASP.

  – To create a new IASP, follow the wizard prompts. This is the simplest method and typically presents the least number of problems.

To create a switchable IASP using the wizard, follow these steps:

1. Click the **+** (plus sign) next to the Cluster name to expand the options.

2. Right-click **Switchable Hardware**.
3. Select **New Group**, as illustrated in Figure 7-26.

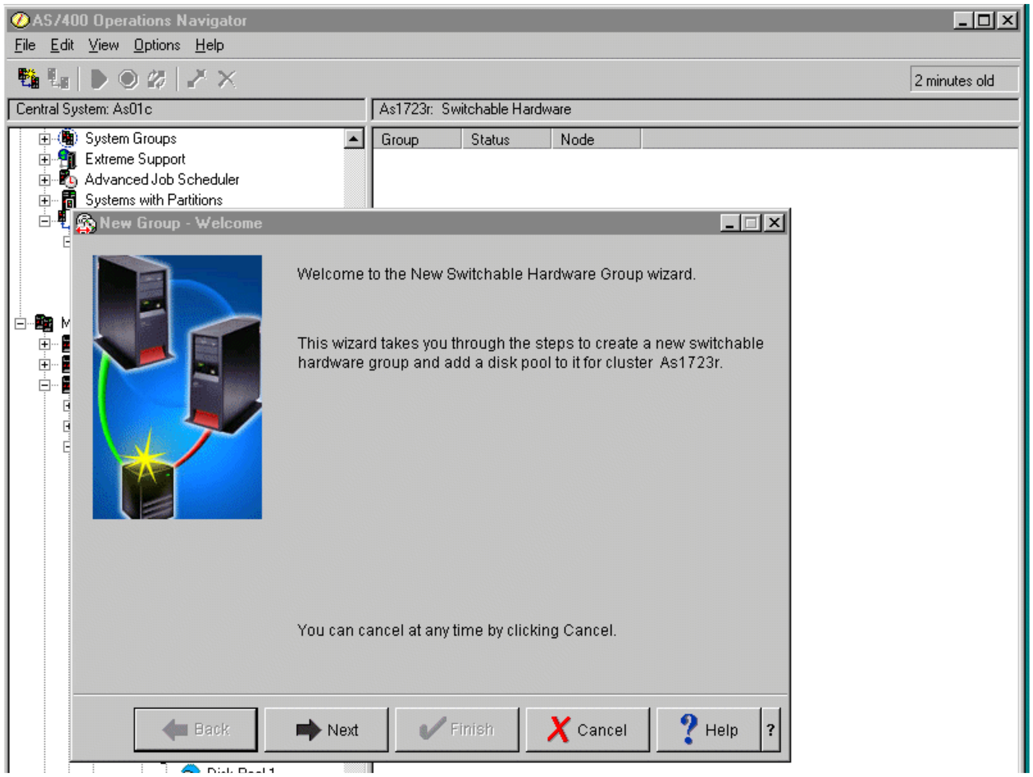


Name	Signed On User	Description
9.5.92.21		Manage this server.
As01b		Manage this server.
As01c		Manage this server.
As24		Manage this server.
Asm01		Manage this server.
Asm15		Manage this server.
Asm24		Manage this server.

Figure 7-26 Creating a new switchable hardware group

A switchable hardware group is another term for a switchable IASP.

4. The Welcome panel (Figure 7-27) of the wizard appears. Click **Next**.



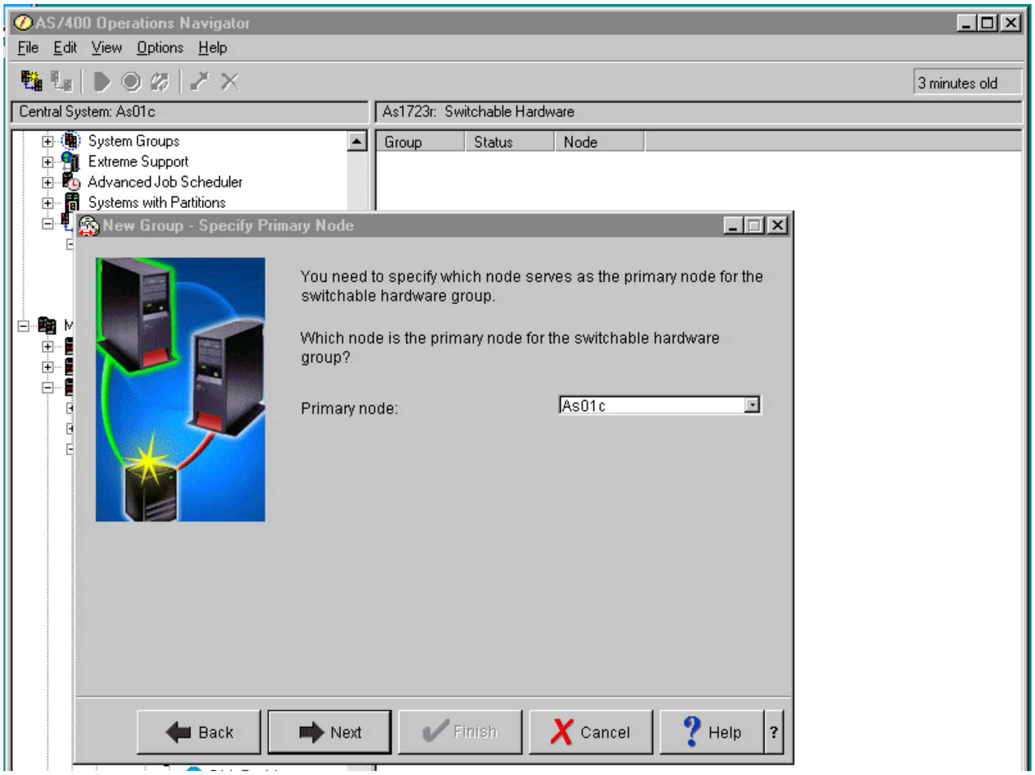
Welcome to the New Switchable Hardware Group wizard.

This wizard takes you through the steps to create a new switchable hardware group and add a disk pool to it for cluster As1723r.

You can cancel at any time by clicking Cancel.

Figure 7-27 The Welcome panel

5. The New Group – Specify Primary Node window appears (Figure 7-28).Type the name of the Primary node of the cluster.



You need to specify which node serves as the primary node for the switchable hardware group.

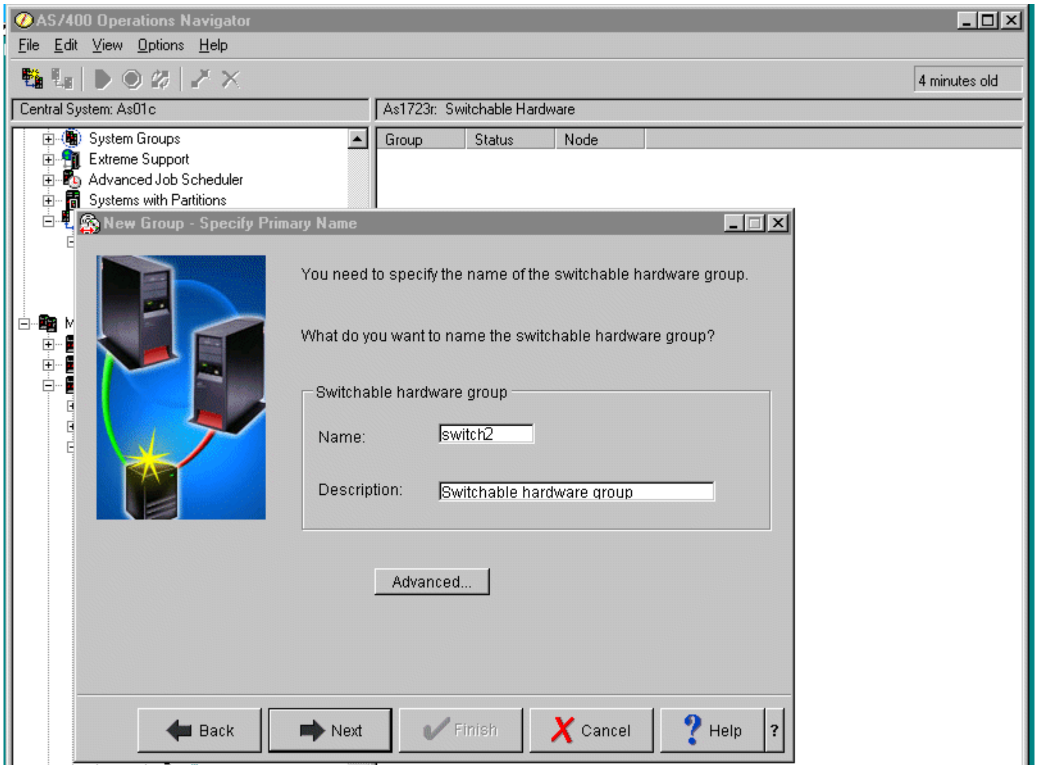
Which node is the primary node for the switchable hardware group?

Primary node: As01c

Figure 7-28 Selecting the primary node

6. Click **Next** to continue.

7. The New Group – Specify Primary Name window appears (Figure 7-29). Specify a name for the Switchable Hardware Group.



You need to specify the name of the switchable hardware group.

What do you want to name the switchable hardware group?

Switchable hardware group

Name: switch2

Description: Switchable hardware group

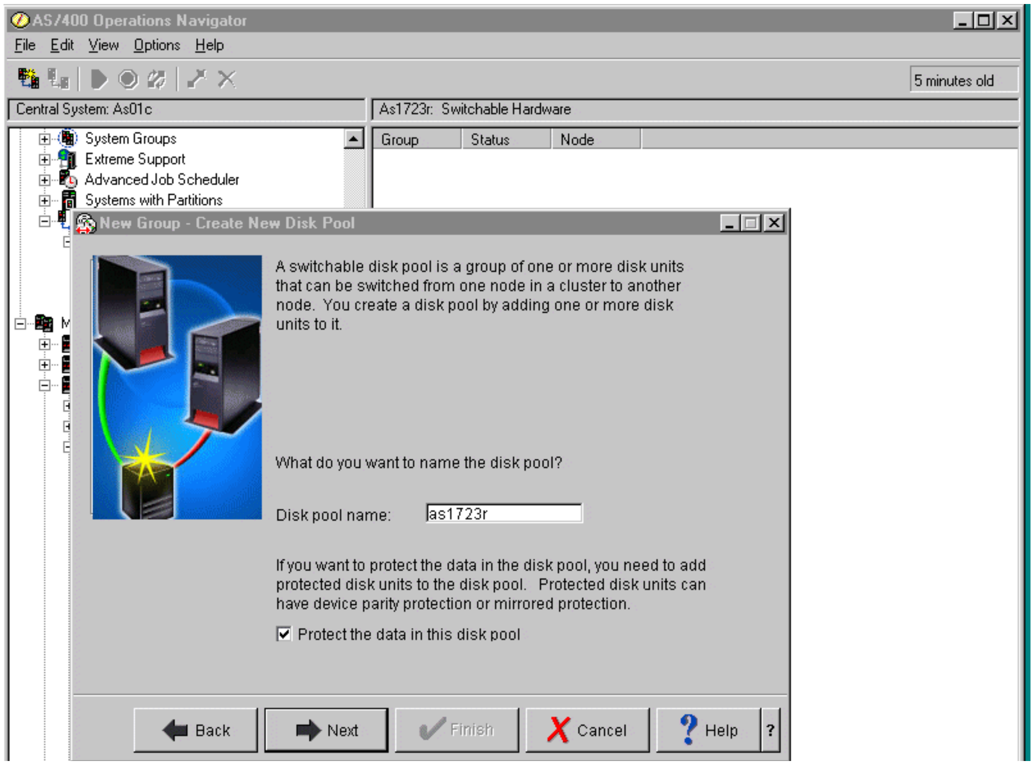
Figure 7-29 Specifying a group name

Note that Switch2 is not the name of the switchable IASP being created. Switch2 is the name of the switchable hardware group (the Switchable Device CRG) under which the switchable IASP is to be managed.

At this point, the independent ASP or disk pool does not yet exist.

8. Specify the name to call the disk pool. Indicate whether to protect the data in the pool, as illustrated in Figure 7-30.

   – If the disk pool does not exist, the New Group – Create New Disk Pool dialog box appears as shown in Figure 7-30.



A switchable disk pool is a group of one or more disk units that can be switched from one node in a cluster to another node. You create a disk pool by adding one or more disk units to it.

What do you want to name the disk pool?

Disk pool name: as1723r

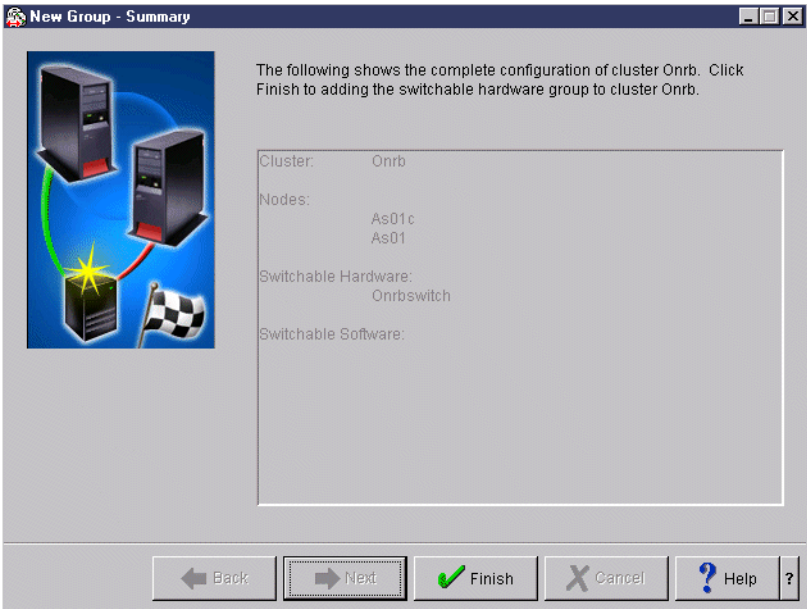
If you want to protect the data in the disk pool, you need to add protected disk units to the disk pool. Protected disk units can have device parity protection or mirrored protection.

☑ Protect the data in this disk pool

Figure 7-30 Specifying the disk pool name

- If the disk pool does exist, a dialogue box appears that asks you to specify whether to use the existing disk pool name.

  If you specify an existing pool name, a dialogue box appears. Click **Yes**. The wizard ends with the New Group – Summary window (Figure 7-31).



The following shows the complete configuration of cluster Onrb. Click Finish to adding the switchable hardware group to cluster Onrb.

Cluster: Onrb

Nodes:
As01c
As01

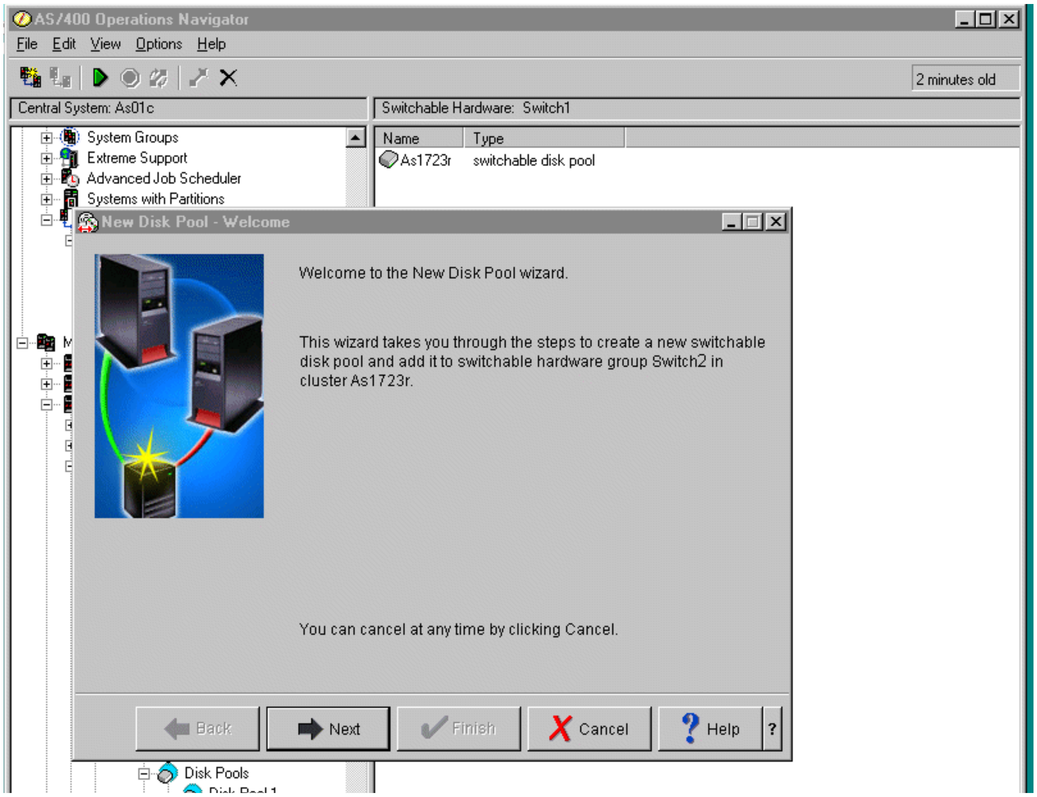
Switchable Hardware:
Onrbswitch

Switchable Software:

Figure 7-31 New Group – Summary

9. Click **Finish**.
10. Click **Next** to continue.

11.For a new disk pool, the Add Disk Pool wizard (Figure 7-32) starts. Click **Next** to continue.



Welcome to the New Disk Pool wizard.

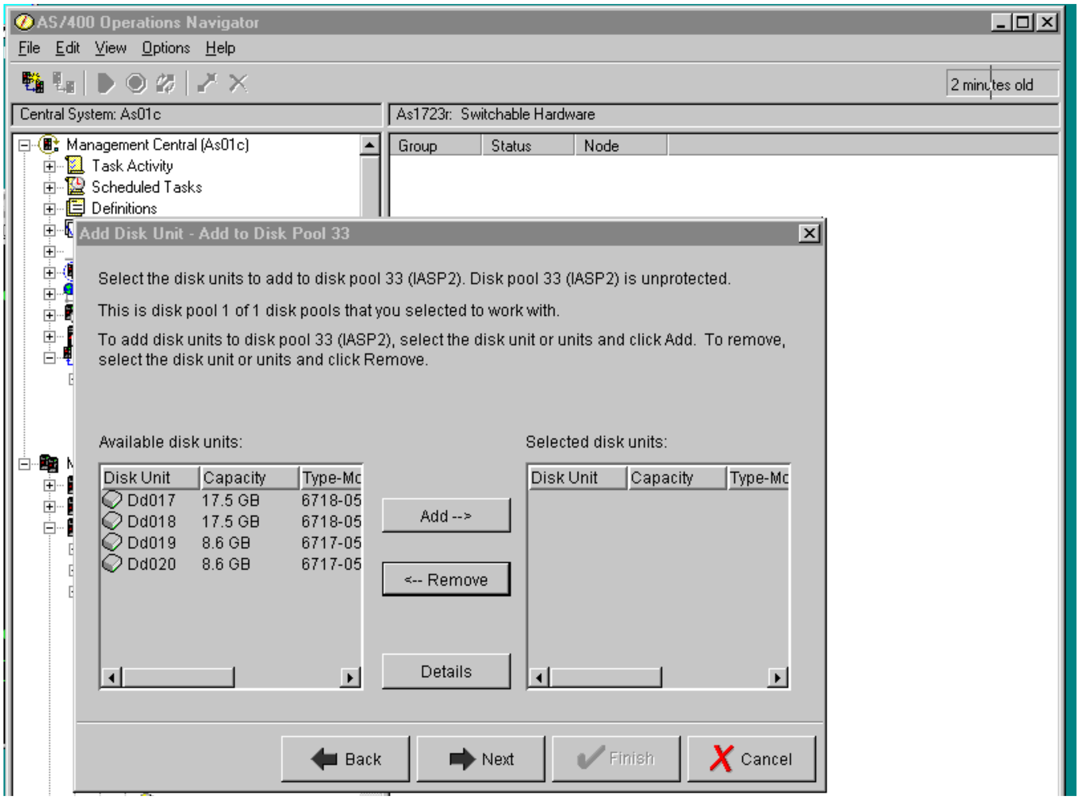
This wizard takes you through the steps to create a new switchable disk pool and add it to switchable hardware group Switch2 in cluster As1723r.

You can cancel at any time by clicking Cancel.

Figure 7-32 Add Disk Pool wizard

12. The Add Disk Unit window (Figure 7-33) appears. The disk units that are available to be part of the new disk pool are located on the left side of the box. Select a disk unit to be part of the disk pool.

    Click the **Add** button. Do this for each of the disk units to be added.


AS/400 Operations Navigator
File Edit View Options Help
2 minutes old
Central System: As01c
As1723r: Switchable Hardware
Management Central (As01c)
Task Activity
Scheduled Tasks
Definitions
Group
Status
Node
Add Disk Unit - Add to Disk Pool 33

Select the disk units to add to disk pool 33 (IASP2). Disk pool 33 (IASP2) is unprotected.

This is disk pool 1 of 1 disk pools that you selected to work with.

To add disk units to disk pool 33 (IASP2), select the disk unit or units and click Add. To remove, select the disk unit or units and click Remove.

Available disk units:

Disk Unit	Capacity	Type-Mc
Dd017	17.5 GB	6718-05
Dd018	17.5 GB	6718-05
Dd019	8.6 GB	6717-05
Dd020	8.6 GB	6717-05

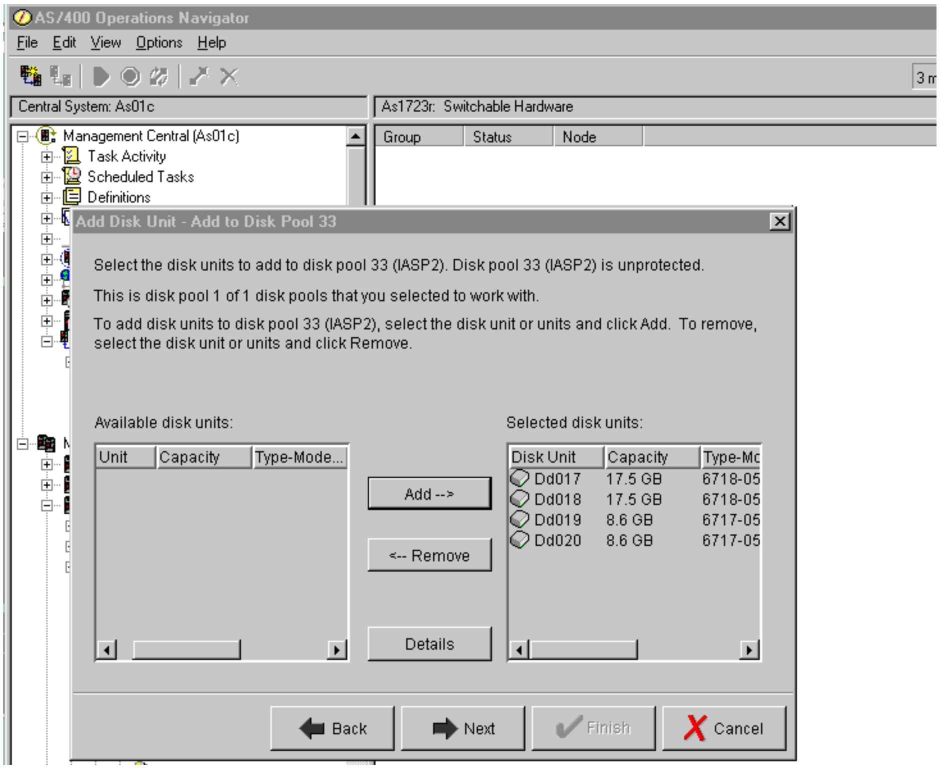
Selected disk units:

Disk Unit	Capacity	Type-Mc

Figure 7-33 Selecting the disk units

Use the **Remove** button to deselect the disk unit if it is selected in error.

An example of how the screen appears after the switchable disk pool disk units are selected is shown in Figure 7-34.


AS/400 Operations Navigator
File Edit View Options Help
Central System: As01c
As1723r: Switchable Hardware
Management Central (As01c)
Task Activity
Scheduled Tasks
Definitions
Group
Status
Node
Add Disk Unit - Add to Disk Pool 33
Add -->
<-- Remove
Details
Back
Next
Finish
Cancel

Select the disk units to add to disk pool 33 (IASP2). Disk pool 33 (IASP2) is unprotected.

This is disk pool 1 of 1 disk pools that you selected to work with.

To add disk units to disk pool 33 (IASP2), select the disk unit or units and click Add. To remove, select the disk unit or units and click Remove.

Available disk units:

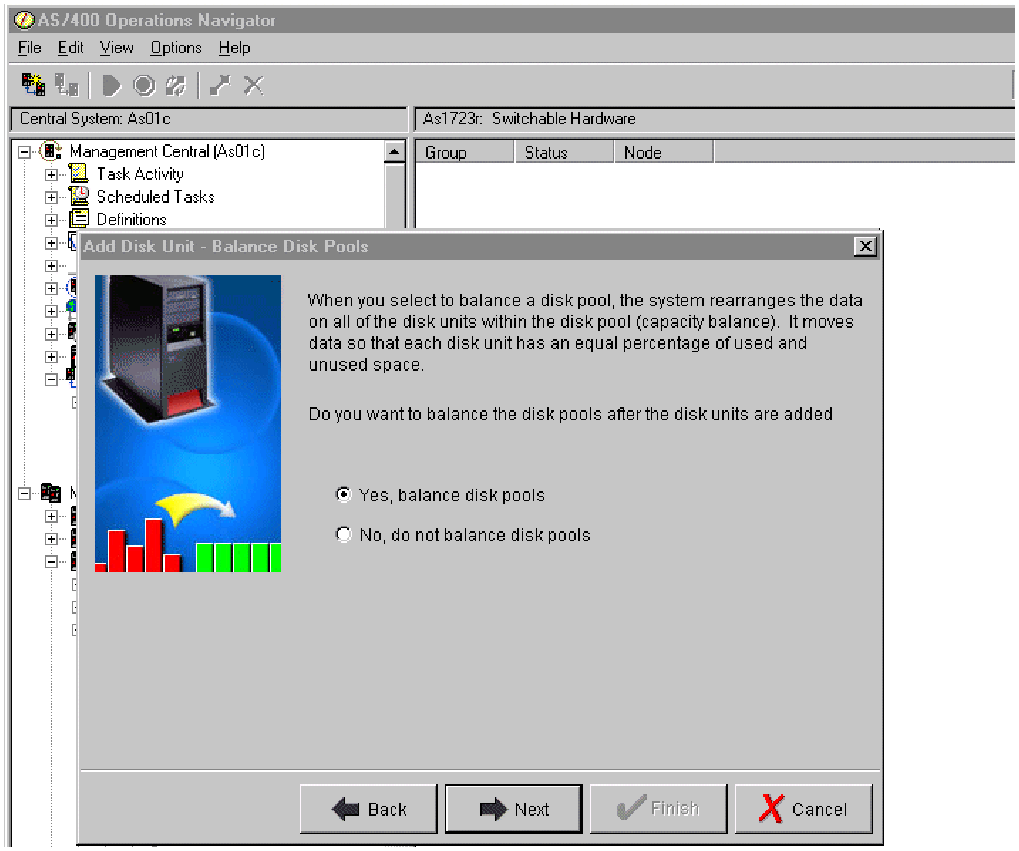
Unit	Capacity	Type-Mode...

Selected disk units:

Disk Unit	Capacity	Type-Mc
Dd017	17.5 GB	6718-05
Dd018	17.5 GB	6718-05
Dd019	8.6 GB	6717-05
Dd020	8.6 GB	6717-05

Figure 7-34 Units selected

13.To balance the disk pool so that an equal amount of data resides on each disk, select **Yes, balance disk pools**, as illustrated in Figure 7-35. Click **Next**.



AS/400 Operations Navigator
File Edit View Options Help
Central System: As01c
Management Central (As01c)
Task Activity
Scheduled Tasks
Definitions
As1723r: Switchable Hardware
Group
Status
Node
Add Disk Unit - Balance Disk Pools

When you select to balance a disk pool, the system rearranges the data on all of the disk units within the disk pool (capacity balance). It moves data so that each disk unit has an equal percentage of used and unused space.

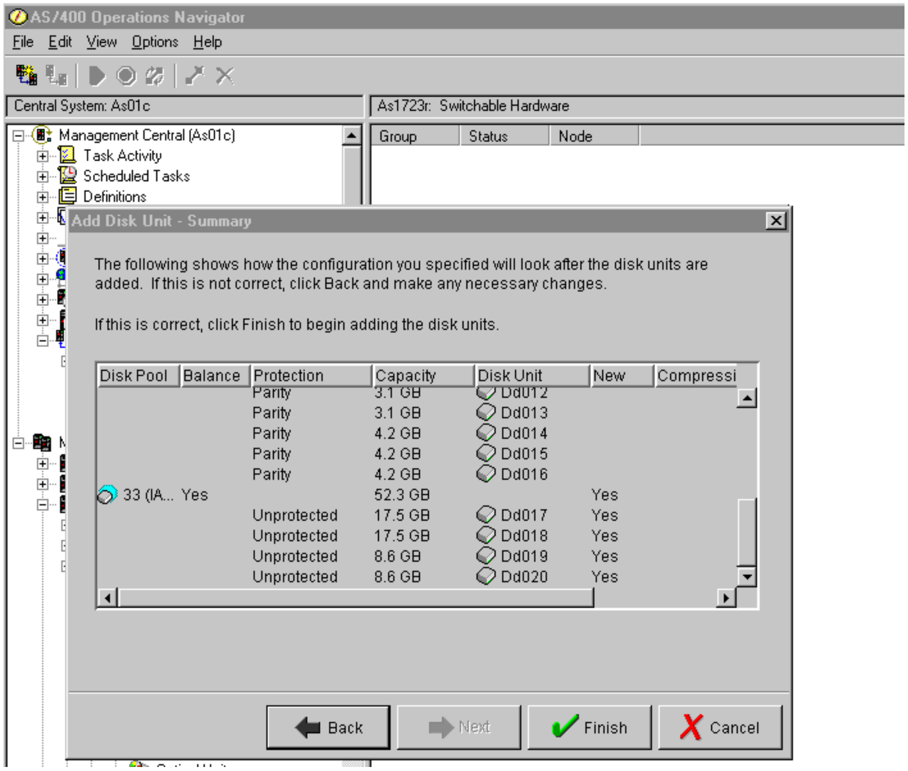
Do you want to balance the disk pools after the disk units are added

● Yes, balance disk pools

○ No, do not balance disk pools

Figure 7-35 Balance Disk Pools

14. A summary of the auxiliary storage pools on the system is displayed, as they appear after the new disk pool is created. See Figure 7-36 for an illustration.



The following shows how the configuration you specified will look after the disk units are added. If this is not correct, click Back and make any necessary changes.

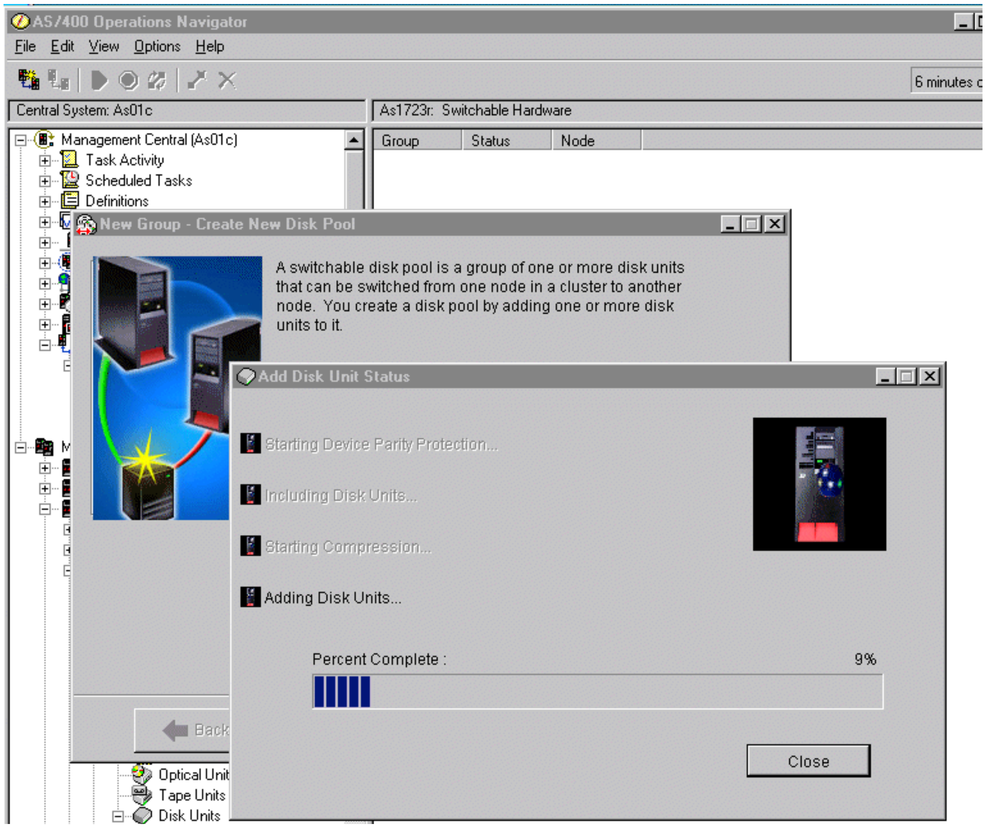
If this is correct, click Finish to begin adding the disk units.

Disk Pool	Balance	Protection	Capacity	Disk Unit	New	Compressi
		Parity	3.1 GB	Dd012		
		Parity	3.1 GB	Dd013		
		Parity	4.2 GB	Dd014		
		Parity	4.2 GB	Dd015		
		Parity	4.2 GB	Dd016		
33 (IA...	Yes		52.3 GB		Yes	
		Unprotected	17.5 GB	Dd017	Yes	
		Unprotected	17.5 GB	Dd018	Yes	
		Unprotected	8.6 GB	Dd019	Yes	
		Unprotected	8.6 GB	Dd020	Yes	

Figure 7-36 Disk pool summary

Nothing is changed on the system up to this point. The disk configuration has not yet been modified.

To proceed with the disk pool creation and commit the changes, click **Finish**. The disk pool is now created. See Figure 7-37 for an illustration.



A switchable disk pool is a group of one or more disk units that can be switched from one node in a cluster to another node. You create a disk pool by adding one or more disk units to it.

Starting Device Parity Protection...

Including Disk Units...

Starting Compression...

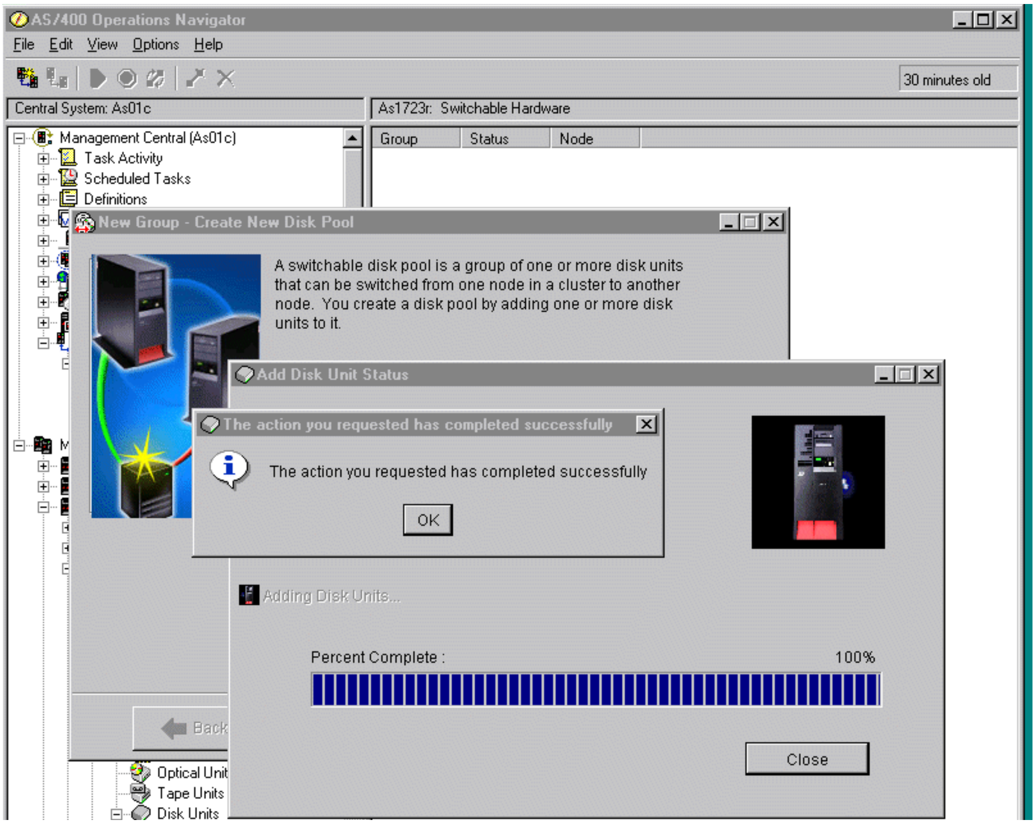
Adding Disk Units...

Percent Complete : 9%

Figure 7-37 Creating disk pool

15.This step can take from a few minutes to one hour or more, depending on the number of disk units to be initialized and added to the IASP.

When the group has been created, a successful creation message is displayed, as illustrated in Figure 7-38.


AS/400 Operations Navigator
File Edit View Options Help
30 minutes old
Central System: As01c
As1723r: Switchable Hardware
Management Central (As01c)
Task Activity
Scheduled Tasks
Definitions
Group
Status
Node
New Group - Create New Disk Pool
Add Disk Unit Status
The action you requested has completed successfully
OK
Back
Close
Optical Unit
Tape Units
Disk Units

A switchable disk pool is a group of one or more disk units that can be switched from one node in a cluster to another node. You create a disk pool by adding one or more disk units to it.

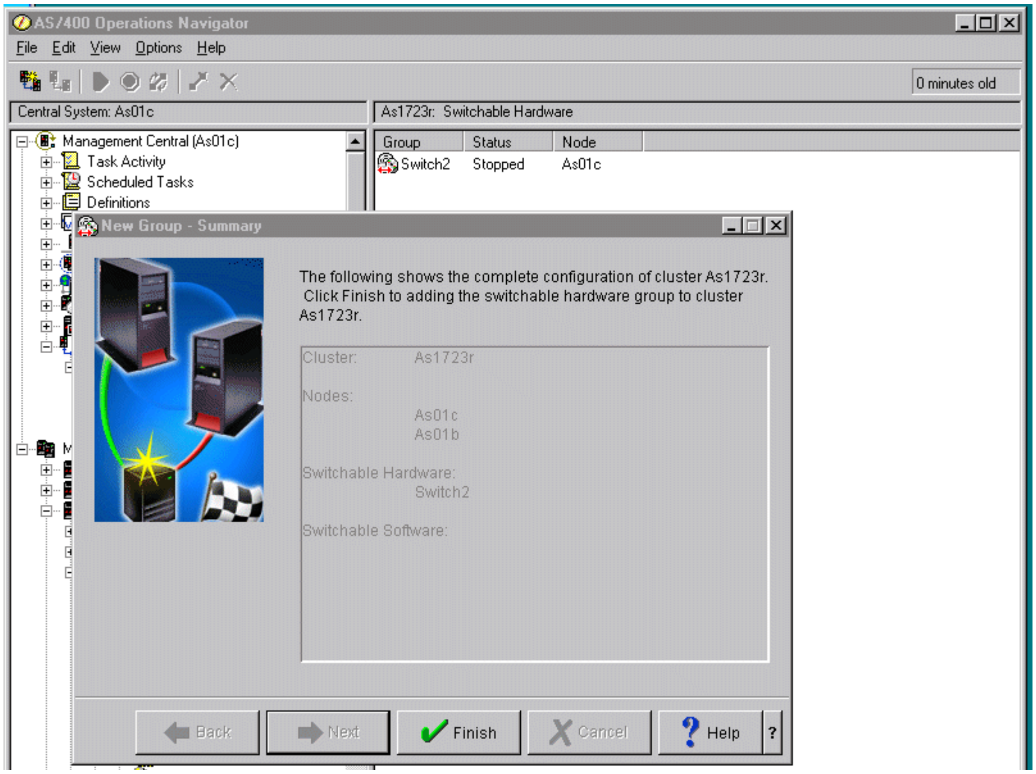
The action you requested has completed successfully

Adding Disk Units...

Percent Complete : 100%

Figure 7-38 Successful creation message

16. The configuration summary of the New Group is displayed. Click **Finish** to continue, as illustrated in Figure 7-39.



The following shows the complete configuration of cluster As1723r. Click Finish to adding the switchable hardware group to cluster As1723r.

Cluster: As1723r

Nodes:
As01c
As01b

Switchable Hardware:
Switch2

Switchable Software:

Figure 7-39 Group summary

17. Right-click the group name to start the new hardware group. Select **Start**, as illustrated in Figure 7-40.

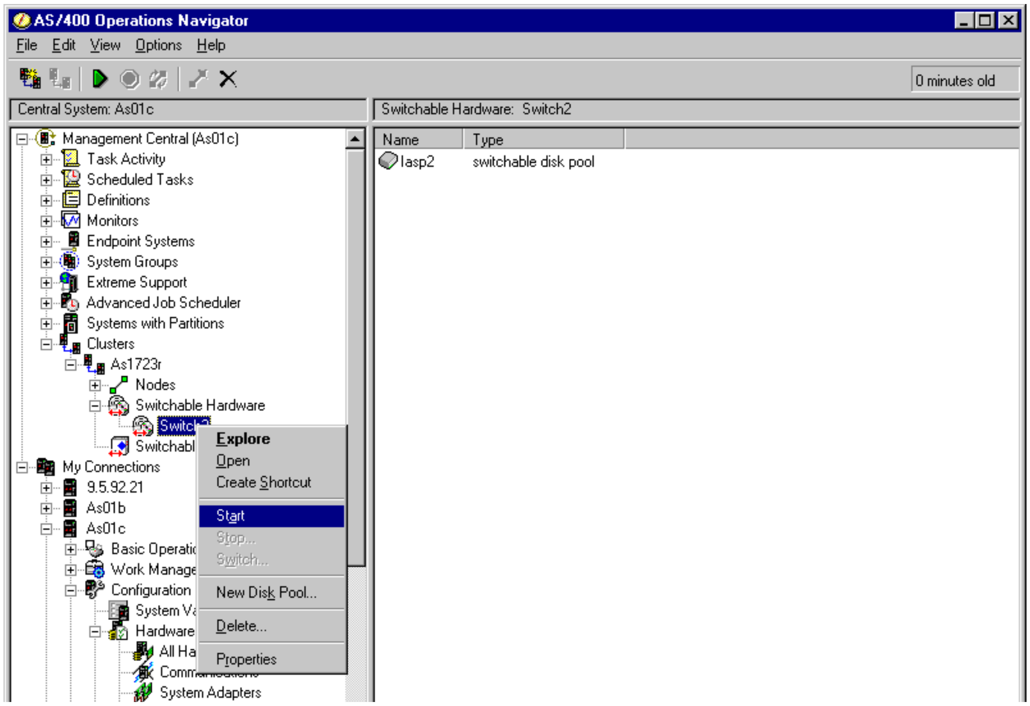


Figure 7-40 Start New Hardware Group

18. The group is, by default, attached to the primary node of the cluster. A window is displayed that shows the IASP as it appears under the Integrated File System of the primary node. The IASP is seen under the Root directory of the IFS. See Figure 7-41 for an illustration.

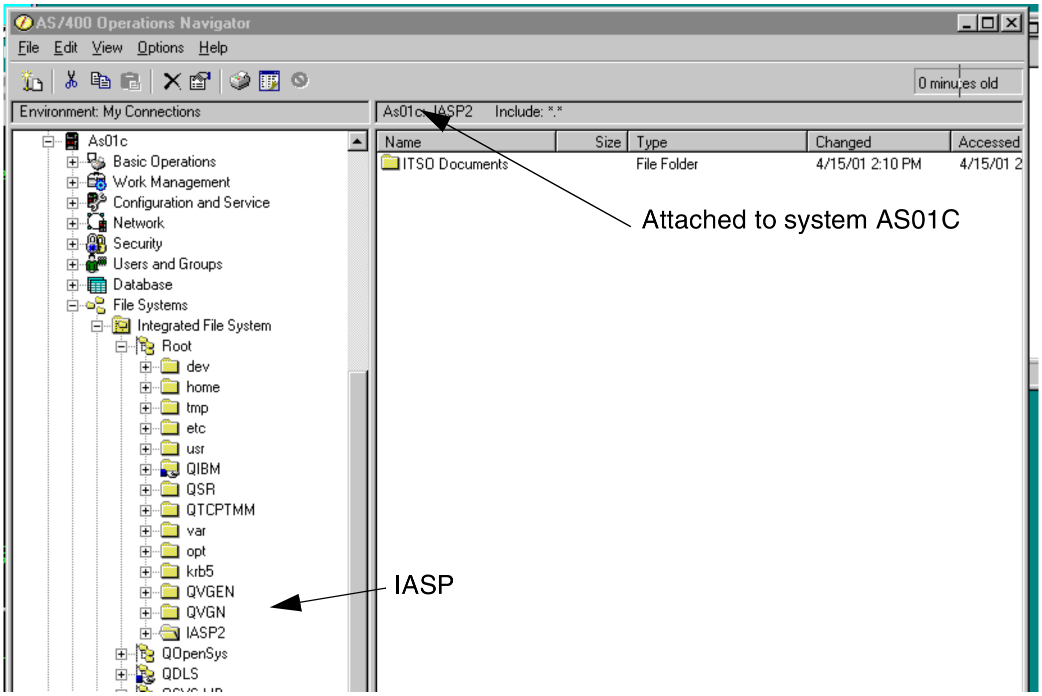


Figure 7-41 IASP before switch

In this example, you see that some PC files have been placed in the IASP. There is a directory named “ITSO documents”.

19. To perform a manual switch of the IASP between systems, right-click the Switchable Hardware Group name. Select **Switch**, as illustrated in Figure 7-42.

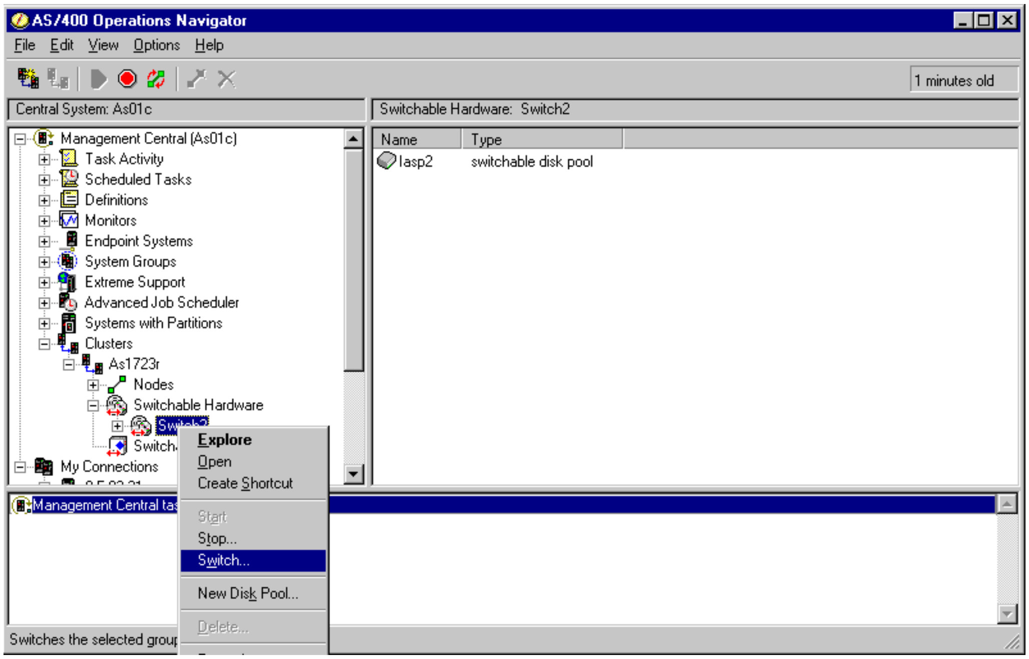
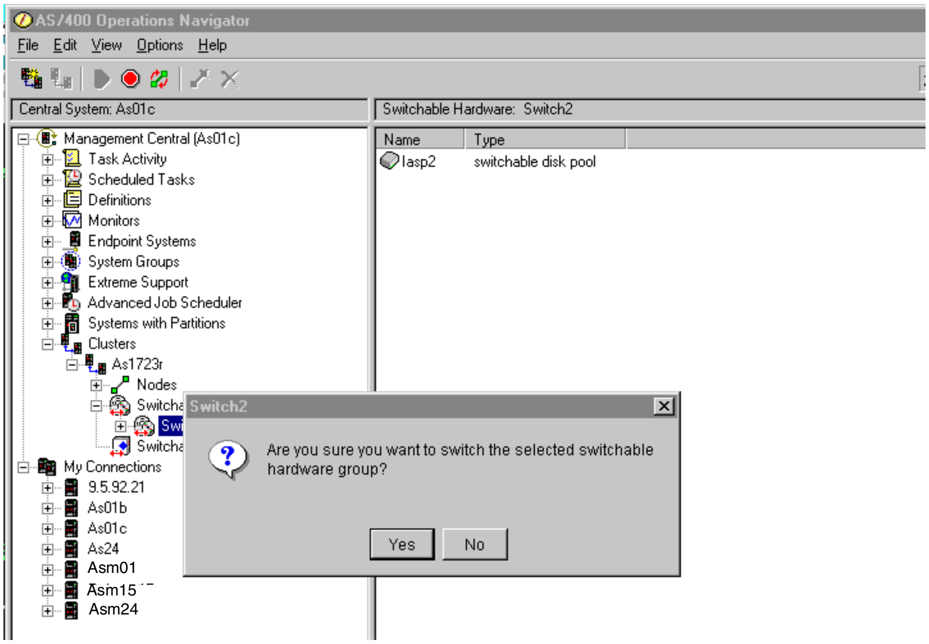


Figure 7-42 Switching the IASP group

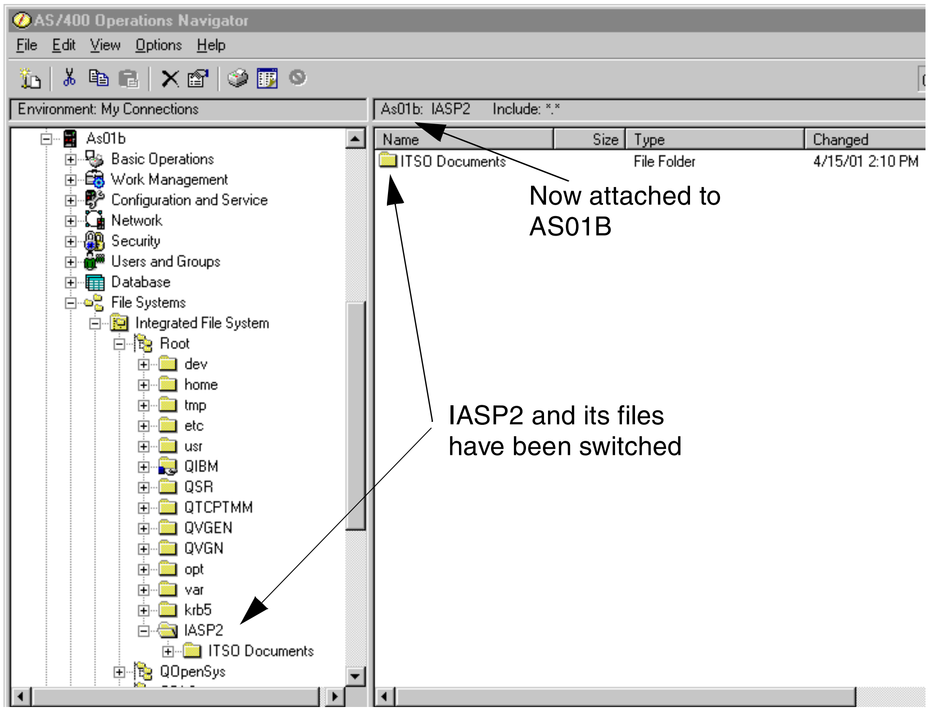
20. A dialogue box appears to confirm the switch of resources. Select **Yes** to confirm this action. The confirmation helps ensure that the IASP is not randomly switched between systems. See Figure 7-43 for an illustration.



Are you sure you want to switch the selected switchable hardware group?

Figure 7-43 Switch confirmation

The switching of resources to the backup system is complete. The disks that represent IASP2 have been switched to AS01B. The switching software handles the vary on. A manual vary on is not necessary. Figure 7-44 illustrates how the results of the switch are indicated on the backup system (AS01B).



Name	Size	Type	Changed
ITSO Documents		File Folder	4/15/01 2:10 PM

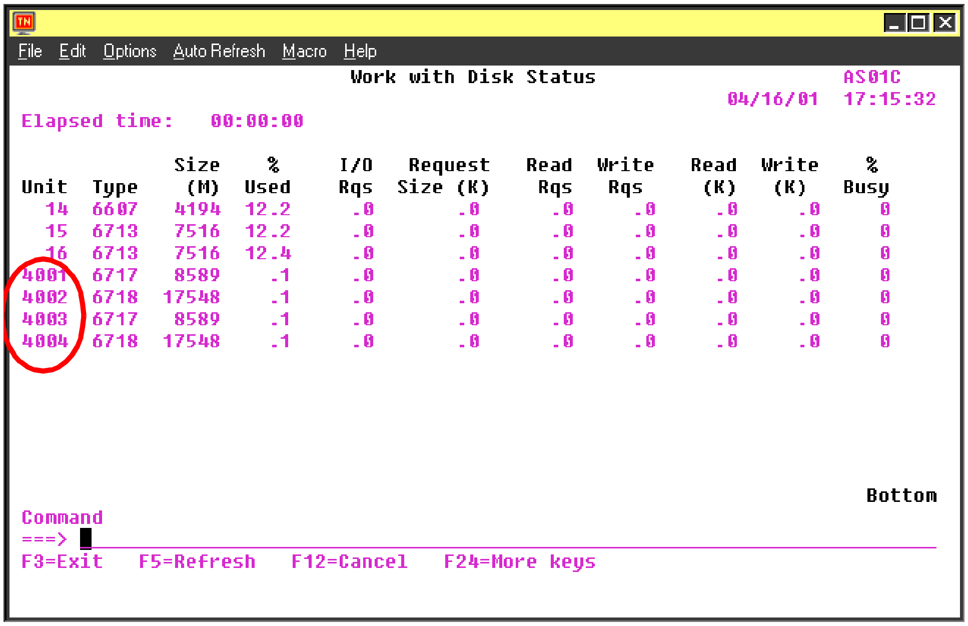
Now attached to AS01B

IASP2 and its files have been switched

Figure 7-44 Disks attached to the backup system

The PC files have been moved and are now available to users on AS01B.

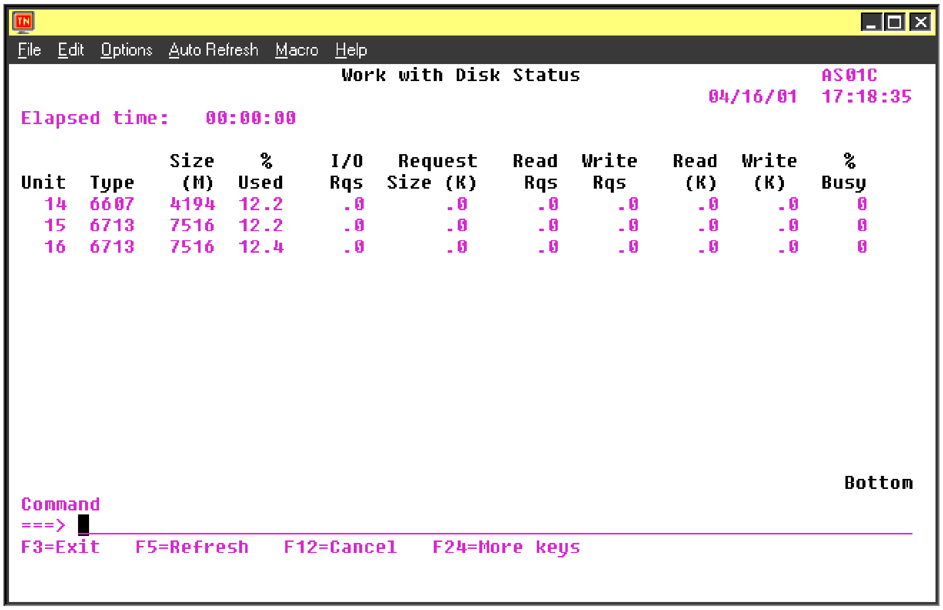
The Work with Disk Status (WRKDSKSTS) command can be used to verify where the IASP disks are attached. Prior to the switch, WRKDSKSTS indicates the disk units are attached to the Primary system (System AS01C). See Figure 7-45 for an illustration.



```
                          Work with Disk Status                        AS01C
                                                             04/16/01  17:15:32
Elapsed time:    00:00:00

             Size    %      I/O   Request   Read  Write   Read  Write    %
Unit  Type    (M)   Used    Rqs  Size (K)    Rqs    Rqs    (K)    (K)   Busy
  14  6607   4194   12.2     .0        .0     .0     .0     .0     .0      0
  15  6713   7516   12.2     .0        .0     .0     .0     .0     .0      0
  16  6713   7516   12.4     .0        .0     .0     .0     .0     .0      0
4001  6717   8589     .1     .0        .0     .0     .0     .0     .0      0
4002  6718  17548     .1     .0        .0     .0     .0     .0     .0      0
4003  6717   8589     .1     .0        .0     .0     .0     .0     .0      0
4004  6718  17548     .1     .0        .0     .0     .0     .0     .0      0




                                                                     Bottom
Command
===>
F3=Exit   F5=Refresh   F12=Cancel   F24=More keys
```

Figure 7-45 WRKDSKSTS shows disks attached to primary node

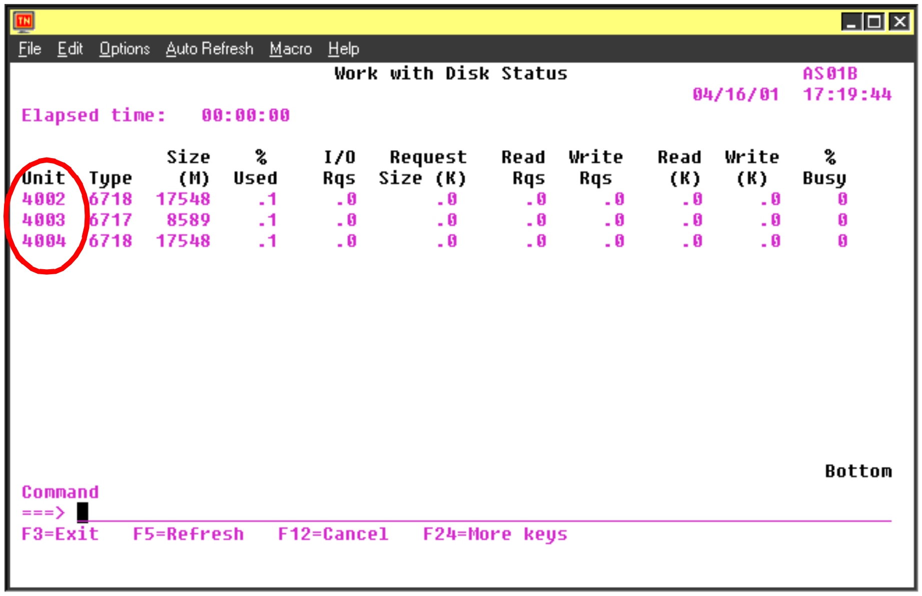
Note that the disk units appear as type 4xxx, which indicates they are switchable. After the switch is performed, the 4xxx disk units are no longer available to the original system. Figure 7-46 illustrates the Primary system (AS01C) after the switch.



```
                          Work with Disk Status                        AS01C
                                                             04/16/01  17:18:35
Elapsed time:    00:00:00

             Size    %      I/O   Request   Read  Write   Read  Write    %
Unit  Type    (M)   Used    Rqs  Size (K)    Rqs    Rqs    (K)    (K)   Busy
  14  6607   4194   12.2     .0        .0     .0     .0     .0     .0      0
  15  6713   7516   12.2     .0        .0     .0     .0     .0     .0      0
  16  6713   7516   12.4     .0        .0     .0     .0     .0     .0      0




                                                                     Bottom
Command
===>
F3=Exit   F5=Refresh   F12=Cancel   F24=More keys
```

Figure 7-46 Disks are not visible after the switch

The IASP is now part of the backup system. Figure 7-47 illustrates this with the output of the WRKDSKSTS command on System AS01B. Note that the disk units appear as type 4xxx, which indicates they are switchable.



```
                              Work with Disk Status                      AS01B
                                                                04/16/01  17:19:44
Elapsed time:   00:00:00

                   Size     %      I/O   Request   Read  Write   Read  Write    %
Unit  Type          (M)   Used     Rqs  Size (K)    Rqs    Rqs    (K)    (K)  Busy
4002  6718        17548     .1      .0        .0     .0     .0     .0     .0     0
4003  6717         8589     .1      .0        .0     .0     .0     .0     .0     0
4004  6718        17548     .1      .0        .0     .0     .0     .0     .0     0









                                                                           Bottom
Command
===>
F3=Exit   F5=Refresh   F12=Cancel   F24=More keys
```

Figure 7-47 IASP attached to backup node

8

ClusterProven for iSeries applications

Disruption of a user’s access to the system is visible to the user through the application interfaces used to process transactions and run applications. True availability is achieved when user applications are consistently, and constantly, available to the user. This application resiliency requirement is recognized and designed into the iSeries cluster architecture.

IBM recognizes the importance of continuous availability and supports the need for applications to participate in a high availability solution. IBM ClusterProven branding is recognized across all four @server platforms to drive the standard for availability higher. ClusterProven branding identifies those applications that take steps to increase application resiliency.

The ClusterProven program has unique criteria for each platform. ClusterProven for iSeries is defined to mean that an application can switch over to an alternate cluster node, provide for automated configuration and activation by cluster management, and return the application user to an application menu screen or beyond after a failover, while the user is active.

ISV applications that meet the requirements are listed in the IBM Global Solutions Directory as ClusterProven and can display the ClusterProven logo (see Figure 8-1) in an approved manner.

This chapter describes the process for applications to achieve the ClusterProven for iSeries designation.

Figure 8-1 IBM logo for ClusterProven applications

8.1 ClusterProven components

The iSeries operating system provides the basic cluster infrastructure that combines the functions of the Cluster Resource Services, cluster middleware software, and the application software to produce the iSeries cluster solution. This partnership recognizes the need for both data resiliency and application resiliency in the solution. The components of a cluster partnership are presented in Figure 8-2.

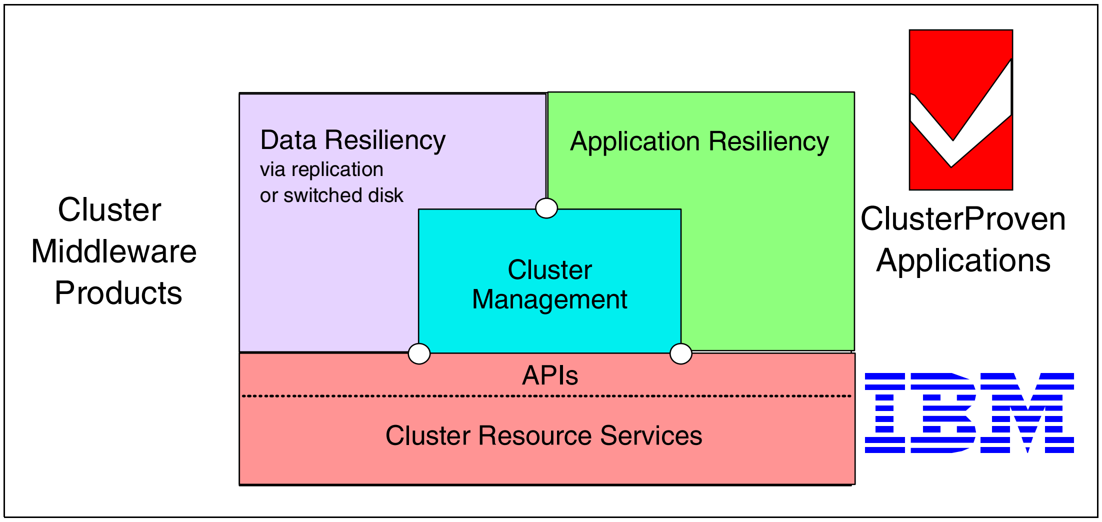


Figure 8-2 iSeries cluster partnership

8.1.1 OS/400 Cluster Resource Services

Cluster Resource Services provide cluster functions and an open set of application programming interfaces (APIs). As part of the iSeries base operating system since V4R4, applications can use these services to create and manage a cluster.

Cluster Resource Services establish the architecture from which all high availability business partners and independent software vendors can build solutions to enable high availability. The Cluster Resources are controlled by the iSeries server.

8.1.2 Data resiliency

Data resiliency means to maintain one or more copies of application data on one or more backup systems or logical partitions so that the data is always available. Data resiliency can be achieved within the iSeries cluster architecture by replication or switched disk technology.

Switched disk technology uses switchable towers, independent ASPs, and other system functions to make the data resilient to the user. A single copy of the data is maintained on disk towers that can be made available to a second server in the event of the loss of the primary server.

Replication technology uses journaling and other techniques to keep an up-to-date copy of the data on a backup server. Cluster middleware products from DataMirror, Lakeview Technology, and Vision Solutions offer products that replicate data objects to backup iSeries servers. Their replication functions are controlled by data CRGs, and their products each provide an associated exit program that seamlessly interfaces with Cluster Resource Services.

8.1.3 Cluster management

Cluster management provides the interface to the cluster operator for controlling the creation and operation of the cluster. It is the primary interface for handling all data resiliency and application resiliency operations in the cluster. Control of the cluster nodes (adding, removing, starting, and ending) and control of all CRGs (starting, stopping, and switchover) are handled from a single interface located on any cluster node.

Cluster management can be provided by Management Central within Operations Navigator for clusters with applications that use device CRGs. Cluster middleware products include sophisticated cluster management utilities for full support of all CRGs (device, data, and application) and clusters with more than two nodes.

Refer to Part 3, “Cluster middleware business partners” on page 227, for further information on the cluster management products offered by DataMirror, Lakeview Technology, and Vision Solutions.

Refer to Chapter 5., “Implementing and managing clusters with IBM solutions” on page 59, for further information on managing two-node switch disk solutions with the cluster functions within Management Central.

8.1.4 Application resiliency

Resilient applications provide for maintaining across a failure, or an automated recovery of, the application user’s state to the primary server and the resulting switchover to a backup cluster node.

Within the iSeries cluster architected solution, a resilient application is one that uses data or device CRGs, application CRGs, and an IP takeover address to maintain resiliency. There are levels of participation in the iSeries cluster. An application can be unaware of the cluster and only have the application data controlled by a data CRG. Or it could have a universal application CRG exit program that allows it to have an IP takeover address and start and stop control of the application. Or, the application CRG exit program can include recovery actions.

The options for iSeries applications operating in an iSeries cluster range from no involvement at all to designing for the highest level of application resiliency with no loss of the end user display during a failover in a client server environment.

Applications that take full advantage of the iSeries cluster functions available to them can qualify to become ClusterProven for iSeries applications.

There are two designations of resiliency for iSeries applications: ClusterProven and Advanced ClusterProven. The criteria for each designation is discussed in the following section.

8.2 ClusterProven defined

ClusterProven for iSeries designation is available for applications that meet specific availability design criteria based on OS/400 Cluster Resource Services functions and architecture. Simple ClusterProven requires returning to an application menu display.

The focus is on application resiliency and the ability to restart the application on the backup server. The key to application restart is in the ability to reproduce the job state on the backup server. The job state includes the current user, internal program variables, partially written spooled files, record pointers, program call stack, and other job-related attributes. Keeping track of all these elements for a traditional, green-screen interactive application can present a difficult challenge because the job state information is stored on the primary system. Client/server applications are more conducive to application restart because the client stores most (and in some cases, all) of the job state information.

As designs are made for higher levels of availability, the user state is maintained in the client. Or the application is designed to maintain the user state on the backup server and actions are taken to return the user to the proper state on a subsequent signon of the particular user after a failover.

An application designated as ClusterProven for iSeries meets the requirements of the specification to:

- Provide information to enable automatic configuration and activation for resilient resources
- Supply an exit program to restart the application
- Integrate with data resiliency services (frequently provided by a cluster middleware provider)

The specifications for ClusterProven, including the Resilient Definition and Status Data Area, Object Specifier File, and the application CRG exit program, are located on the iSeries high availability Web site at: `http://www.ibm.com/eserver/iseries/ha`

A sample exit program is also provided in the QUSRTOOL library (OS/400 Option 7), member name TCSTAPPEXT in the file QATTSYSC. The required effort is application dependent.

You can find the layout of an object specifier file in B.4, “Object specifier file layout” on page 288.

8.2.1 The Advanced ClusterProven for iSeries Program

The *Advanced ClusterProven for iSeries Program* offers a branding logo for solution developer applications that demonstrate highly available characteristics. Criteria for iSeries Advanced ClusterProven include application characteristics that:

- Meet all of the criteria for ClusterProven for iSeries program listed in 8.2, “ClusterProven defined” on page 163.
- Provide enhanced application resilience through more robust handling of cluster events (action codes) by the application CRG exit program.
- Provide greater level of application restart support.
- For host-centric applications, the user is repositioned to a transaction boundary via commitment control or checkpoint functions.
- For client-centric applications, the user experiences a seamless failover with minimal service interruption.

8.3 Obtaining the ClusterProven trademark

Obtaining the ClusterProven for iSeries trademark is a two-step process:

1. Validate the application against the criteria listed in Table 8-1 and submit the results to iSeries PartnerWorld (PWD). PartnerWorld is the final arbiter of whether the criteria for registration as ClusterProven for iSeries is met.
2. The software developer agrees to the terms and conditions covering the use of the ClusterProven trademark.

Regardless of whether an application is verified as meeting the criteria herein, or whether IBM has accepted such verification, no right to use the trademarks is granted until both parties have signed the ClusterProven Trademark Agreement.

Criteria to obtain ClusterProven trademark

The type of criteria used to identify applications as ClusterProven for the iSeries server is listed in Table 8-1.

Table 8-1 ClusterProven for iSeries criteria

Function	Characteristic
Cluster management resilient environment	► An auto configuration of application resiliency is performed using the architected data area in the application produce library. The data area identifies the resources to be resilient. ► Auto-activation of application resiliency from Cluster Management product. ► Changes to the data objects are available to backup nodes due to replication or by storing on a switchable IASP.
Application resiliency	► Application CRG is generated through auto configuration, representing the application function. ► Application CRG exit program handling of action codes, including start, end, restart, switchover, and failover. ► IP takeover is utilized for application failover. ► Failover is coordinated with associated data or device CRGs. ► Application resiliency is configured using the information found in the automated installation data area. ► A switchover operation results in the transfer of the primary data server to the first backup data server defined in the data CRG. ► The IP address of an application CRG moves from the primary application server to the first backup application server as defined in the application CRG. ► For a host-centric application the active session at the point of switchover returns to an application menu screen or beyond following a subsequent log-on at the backup server. ► For a client-centric application, the active session at the point of switchover resumes at the same point on the backup server. ► The application CRG exit program validates that the required data CRG or CRGs are active on the backup node.
Data resiliency	► A data or device CRG is generated through auto-configuration, representing the data function. ► The application-related data objects listed in the object specifier file associated with a data or device CRG are copied to backup cluster nodes or stored in a switchable IASP. ► Following a switchover, the application moves to the current primary data server for subsequent data operations.

You can find additional information about ClusterProven applications at:
http://www-1.ibm.com/servers/eserver/iseries/ha

Note: You may come across terms such as “cluster ready” or “cluster aware” as relating to applications. This means that these applications can, to some extent, interact with an OS/400 cluster, but not as painlessly as described above. They are not part of the ClusterProven designation.

8.4 ClusterProven Domino for iSeries

Lotus Domino for iSeries Release 5.0.7 is certified by the Rochester laboratory as ClusterProven in April 2001. In a switched disk environment, the same Domino server can be configured on a second iSeries server (cluster node) provided both systems can switch access to that server’s data directory by using an independent auxiliary storage pool. The same Domino server can be restarted on that second sever with minimal interruption in the server user’s environment. This means that the Domino application can have its configuration and files switched over to a second system (cluster node) with minimal or no interruption in the running Domino environment.

ClusterProven Domino for iSeries is an OS/400-specific enhancement to Domino that allows V5R1 OS/400 Cluster Management to manage iSeries-based Domino servers. A Domino server's definition is duplicated on multiple iSeries servers (known as nodes) in an OS/400 cluster managed by OS/400 Cluster Management. The same Domino server is automatically defined on all the nodes in the cluster that can access a switchable IASP automatically that stores the servers' data directory.

With OS/400 ClusterProven clustering support, the clustering is done on a server's entire data directory and not at a Domino database level. Instead of the databases being replicated, the definition of the Domino server itself is duplicated on other cluster nodes. The cluster supported server's data directory is kept on a switchable disk device. Once configured, the same Domino server can be started on any iSeries logical partition or server that can potentially access the server's data directory on that switchable device. The OS/400 Cluster Management GUI interface can be used to start, stop, and switch between nodes configured for the clustered Domino server.

Only one cluster node can access a shared IASP resource at a time. A ClusterProven Domino server is prevented from starting if the cluster node cannot access the IASP, since that server’s data directory is not accessible then. The Domino server is prevented from starting on more than one cluster node since the Domino server is managed as a clustering application Cluster Resource Group (application CRG).

Using OS/400 cluster management to support Domino provides automatic failover to backup iSeries servers or LPARs in the case of a system failure. It can also be used to effectively switch a Domino server from one system or LPAR and back. This functionality provides continuous support for the server on a backup system in the case of system downtime for scheduled maintenance or a system IPL.

Figure 8-3 illustrates the clustering solution implemented with Domino on the iSeries server.

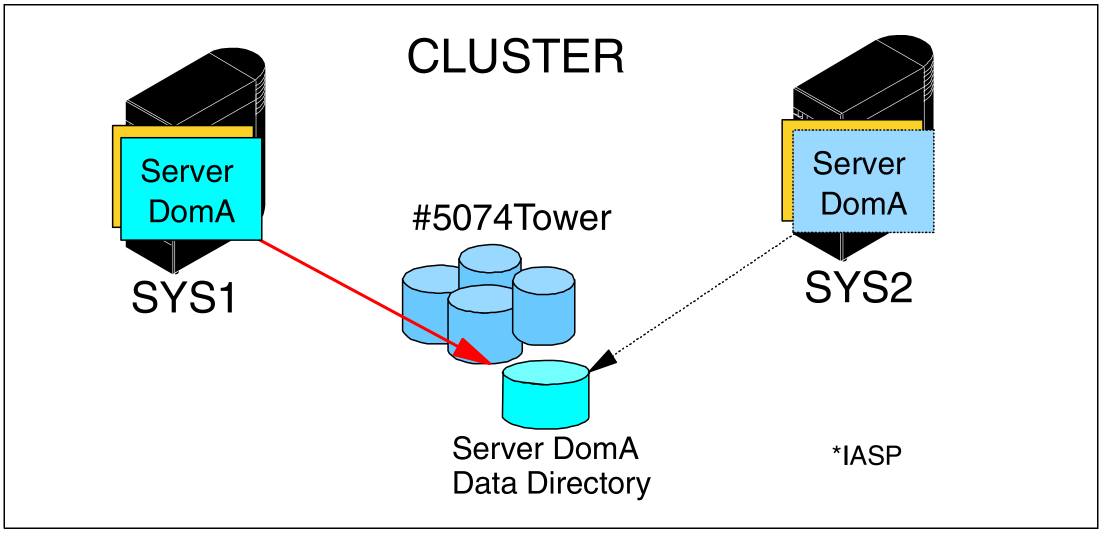


Figure 8-3   ClusterProven Domino for iSeries

The requirements for ClusterProven Domino for iSeries server include:

- OS/400 5.1
- Domino 5.0.7 or later

Note: As illustrated in Figure 8-3, store the Domino server's data directory on a disk storage device that can be switched between all the systems configured as cluster nodes.

8.4.1 ClusterProven Domino customer benefits

The ClusterProven Domino solution offers customers these unique benefits:

- Only one copy of the server's Data Directory and the Domino databases is required

  With traditional Domino clustering, two copies of the Data Directory and the Domino databases are required. The implementation offered by ClusterProven Domino reduces the requirement for disk space by half.

- Reduced or eliminated replication overhead

  With a ClusterProven Domino solution, there is no or very little replication overhead. The Domino database is not replicated.

- You can move the same server between systems or LPARs

  System maintenance can be scheduled without disrupting users access to Domino functions.

- Improved failover operations

  In case of a failure on the primary system, cluster management failover can start the Domino Server on the secondary system.

- Flexible cluster management

  Domino servers can be managed by Cluster Management tools provided by business partners, or by the IBM Cluster Management GUI of Operations Navigator.

8.4.2 Comparing traditional Domino clustering support

It is important to note that the new ClusterProven for Domino support is not the same as Domino clustering support. ClusterProven for Domino can be used in conjunction with Domino clustering support.

Domino database clustering works on all Domino servers on all platforms that support Domino. It is configured on a database level. Each database can be clustered so that if the primary server does not respond, access to that database is rerouted to a backup server where a copy of that database is automatically maintained by Domino.

Domino database clustering is “instantaneous”. Multiple copies of the database are kept. When a server with a database copy fails or is ended, the application or Notes user accessing the database simply gains access to the database copy and starts operating on this copy without much disruption. In that window of time, while accessing the database copy, there is no backup of the data until the original server comes back online and the changes made to the database are replicated to its copy to synchronize things again. The application or Notes user is switched back to the original database copy on the restarted server. In the event that more than one server fails or is ended, to support a database on three (or more) backup nodes, maintain three (or more) copies of the database.

Databases that are clustered at the application level have to be replicated to the backup servers. There is some overhead to do this replication. Traditional Domino clustering support requires a duplicate set of disks to support data replication. That is, copies of the database need to be stored on the backup servers.

Figure 8-4 illustrates the support that is offered with traditional Domino clustering.

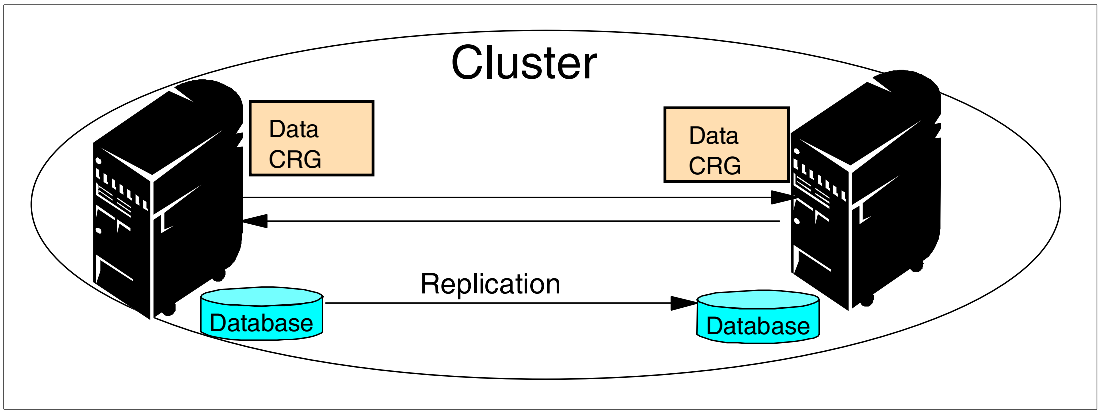


Figure 8-4 Domino replication

For more information on ClusterProven Domino, refer to the “ClusterProven Domino for AS/400” white paper at:
`http://www-1.ibm.com/servers/eserver/iseries/domino/domclust.htm`

9

Making applications continuously available

For an application to support continuous availability, all critical objects defined to the application must be resilient. This includes the programs that make up the application, the data used by the application, and any permanent objects created or used by the application. Each critical object must exist on, or be accessible from, more than one node of the cluster. Switched disk architecture (IASPs) enables the accessibility of objects and application resiliency.

This chapter discusses the treatment of critical objects and actions taken by Cluster Resource Group (CRG) exit programs to support a continuously available environment on iSeries servers.

Note: Example cluster and CRG configuration code is presented in Chapter 10, “Sample cluster environment and applications” on page 183, to support the concepts described within this chapter.

For more information on application considerations and for examples scenarios and coding, watch for the redbook *Moving Applications to Switchable Independent ASP*s, SG24-6802, which will be available later in the second half of 2002.

9.1 Defining critical objects

All critical objects must be replicated or otherwise accessible to support application resiliency.

The first step to make applications resilient is to define which objects are the critical objects. For some applications, this may only require part of the application's environment, while other applications require a complete copy of the environment.

For example, many applications make use of temporary work files, which are recreated each time an application is restarted. It is not necessary to replicate these temporary files to a backup system if this data is not critical in case of an outage.

To address the requirements of resiliency, tools are available that help in the process of making applications ClusterProven. These processes are described in this section. Refer to Chapter 8, "ClusterProven for iSeries applications" on page 161, to further understand the iSeries ClusterProven identity.

9.1.1 Non-ClusterProven applications

Identifying critical objects can be difficult for non-cluster proven applications. With a detailed knowledge of the application, the objects can be identified and specified individually.

Cluster middleware provider products have selection tools to expedite the selection and exclusion process for objects. Generics are supported as a means to specify a complete set of objects, or all objects within a library. Cluster middleware providers have tools available to synchronize and journal the specified objects.

If knowledge of the critical objects is limited, or not currently available, then an alternative approach is to select all objects for replication.

Selecting all objects is not generally recommended, but it does guarantee that all critical objects are replicated. However, depending on the application and communications bandwidth, a lot of extraneous information can be sent to the backup system, causing a communications back log to occur. If an outage occurs (planned or unplanned) while this backlog exists, information can be lost, or an unexpected delay can occur while the backlog is processed.

As more knowledge is gained about the application, eliminate non-critical objects from the replication process to help streamline the operation.

9.1.2 ClusterProven applications

If the application is a ClusterProven application, then the critical objects are determined by the ISV. Application critical objects are predefined in the object specifier file. The object specifier file contains a list of all objects that are critical to making the application resilient.

When a resilient application environment is established, the object specifier file is used by the Cluster Management GUI to ensure that the application environment is synchronized between the primary system and the other recovery domain nodes. The Cluster Management GUI can also ensure that the objects are ready for the replication process, for example, by verifying that journals exist. This is explained in more detail in the following sections.

You can find a layout of the object specifier file in B.4, "Object specifier file layout" on page 288.

9.2 Cluster Resource Group exit program

Once the critical objects are identified, and the cluster is created, the process of defining the recovery domain can begin. Recovery domain situations can be handled manually by the user or automatically by the exit program. This section discusses how the CRG exit programs can be used to establish and manage the recovery domain.

Note: The segments of code documented in this section represent example user exit programs only. The examples show a data and application CRG exit program that supports two nodes: a primary and a backup.

The exit program is responsible for establishing and managing the environment necessary for both data and application resiliency within a cluster. Cluster Services calls the exit program during different phases of a cluster application. As cluster APIs are run, the exit program is called. The functions that the exit program perform depend on the type (action code), status, and role of the node that gains control.

The CRG exit program is called when:

- A node leaves the cluster unexpectedly
- A node leaves the cluster as a result of the End or Remove Cluster Node API
- The cluster is deleted
- A node is activated
- Communication is re-established
- A CRG API is run (except the List API)

Consider these concepts and rules regarding CRG exit programs:

- The exit program name and library name are specified when a CRG is created.
- The exit program must exist in the same library on all nodes within the recovery domain.
- The exit program runs in a named activation group or the caller’s activation group.
- The exit program can be passed up to 256 bytes of data when it is called. This data is specified when the CRG is created and can be modified at other times by the application.

9.2.1 Cluster Resource Group interface

All types of CRGs provide interfaces to exit programs. Exit programs are required for data and application CRGs, and are optional for device CRGs.

As the name implies, the data CRG controls the resiliency of the data and objects. The application CRG handles the application and takeover IP addresses. And the device CRG controls the switchable devices, such as an independent auxiliary storage pool (IASP).

CRG exit programs and the Cluster Management GUI communicate by updating and reading information from architected data areas. The data areas are provided by ISVs and are initialized and updated by applications and cluster middleware products.

9.2.2 Input data area

A clustering input data area named QCSTHAAPPI contains information about the application, application resilience information, and information about required data information. Generally, this data area should only be written to by application CRG exit programs, but can be read by all types of CRG exit programs and the Cluster Management GUI.

The general layout of the QCSTHAAPPI input data area is:

- **Application information**: One portion of the input data area contains information about the application. Such information includes the version of the program, and other information that may be valuable to the application provider and any cluster middleware product.
- **Application CRG information**: One portion of the input data area contains information to create one or more application CRGs.
- **Resilient data information**: One portion of the input data area contains information to identify which objects to make resilient. This information is needed to create the necessary data CRG or device CRG.

Refer to Table B-1 on page 284 for a complete layout of the QCSTHAAPPI input data area.

9.2.3 Output data area

A clustering output data area named QCSTHAAPPO contains information that reflects the results of setting up the resilient environment for the application. Generally, this data area should only be written to by the Cluster Management GUI or the data CRG exit program, but can be read by all types of CRG exit programs.

The general layout of the QCSTHAAPPO output data area is:

- **Application CRG information:** This portion of the output data area contains information about the cluster and the application CRG created.
- **Data and device CRG information**: This portion of the output data area contains information about the CRG created and the replication or switched disk environment established for the application CRG.

Refer to Table B-2 on page 285 for a complete layout of the QCSTHAAPPO output data area.

9.2.4 Returning from the exit program

Before the exit program ends, the value of the success indicator must be set.

If the exit program process is successful, set the success indicator to successful (success indicator = 0). Cluster resources continue with the appropriate function. However, if a non-recoverable error occurs during the exit program, set the indicator to unsuccessful (success indicator = 1 or 2).

Depending on the action code, Cluster Services calls all nodes with an action code of 15 (Undo), which reverses the unsuccessful processes.

9.2.5 Using the exit program to establish the environment

An exit program can be used to configure and verify that the recovery domain is setup correctly.

> **Important:** The data CRG exit program is typically provided by the cluster middleware program that performs data replication. The code snippets provided in this chapter are for example purposes only.
>
> An example application CRG exit program is provided with QUSRTOOL at V5R1.

Follow these steps when a CRG is established:

1. Create the resilient objects on the recovery domain nodes.
2. Synchronize the resilient objects on the recovery domain nodes.
3. Create the journaling environment on the recovery domain nodes.
4. Journal the critical objects (as required).
5. Start the replication process.
6. Synchronize the data.

Note: Although some of these steps are clearly for a data CRG, similar steps may be necessary for an application CRG to establish the resilient environment and to handle application state or application control information.

Table 9-1 and the examples in this section identify how the exit program can be used to handle these environmental issues. The examples provided contain more detail on the specific functions of the exit programs. In each example, the first description shows the data CRG. The second description shows the application CRG.

Table 9-1 summarizes the actions taken by the data CRG and application CRG exit programs for a subset of the action codes.

Table 9-1 Action code and exit programs

Action code	data CRG exit program	Application CRG exit program
1 = Initialize	Prime and put data on all nodes in recovery domain.	Prime and put applications on all nodes in recovery domain.
2 = Start	Start the remote journal and start the replication.	Starts the application.
4 = End	Stop the remote journal and stop the replication.	End the application.
11 = Add	Perform an Initialize (action code 1). If CRG is active, perform Start (action code 2).	Perform an Initialize (action code 1).
13 = Change	Redirect replication and journaling if necessary.	Nothing.
10 = Switch (planned)	Stop replication and stop remote journal.	Stops application on primary and starts it on the backup.
9 = Switch (failover)	Redirect remote journal receivers.	Start application on backup.

Note: Only a subset of all possible action codes are described here. For a full listing, refer to the iSeries Information Center at: http://www.ibm.com/eserver/iseries/infocenter

Creating a Cluster Resource Group

When a CRG is created, the exit programs on each node are called with an action code of 1 (initialize). Figure 9-1 shows an example of the data CRG exit program when a CRG is created.

```
Scenario A: /* Initialize - called when Cluster Resource Group created.*/

     /* Init haappo dataara status to available */
     setHaappoStatus(dataAvailable);

     /* Get Journal details, then create. */
     if( 0 == memcmp(g.nodeInfo.node1Name.c,
                     ibmData->This_Nodes_ID, 8) )
       CRTJRNENV(g.jrnName.objName.c, g.jrnName.libName.c,
                 g.nodeInfo.node2Name.c);
     else
       CRTJRNENV(g.jrnName.objName.c, g.jrnName.libName.c,
                 g.nodeInfo.node1Name.c);

     /* Start Journaling on the pertinent files */
     startJournalingFile(userData);
     break;
```

Figure 9-1 Scenario A: Creating a data CRG exit program example (EXITPGM)

The exit program performs these functions:

- Sets the QCSTHAAPPO data area to a status of “Data Is Available (A)”. This sets the status field at a known starting point for subsequent operations.
- Creates the journal environment. In this example (Scenario A), it creates a remote journal environment. Therefore, it calls a function to return the node information. It then calls the CRTJRNENV function with the appropriate node name.
- Journals the appropriate files.

Note: The list of files to start journaling on is retrieved from the object specifier file. This process involves these steps:

1. Retrieve the name of the object specifier file from the QCSTHAAPPI data area.
2. Open the object specifier file.
3. Read a record from the object specifier file.
4. If the entry is a generic name, find all objects which match
5. Start journaling on that file.
6. Repeat steps three, four, and five until all records have been processed.

Consider these additional functions for the exit program to perform:

- Create the resilient objects on all nodes in the recovery domain
- Verify the objects are synchronized

Important: The data CRG exit program is typically provided by the cluster middleware program that performs the data replication. The code snippets provided in this chapter are for example purposes only.

An example application CRG exit program is provided with QUSRTOOL at V5R1.

Figure 9-2 shows an example of the application CRG exit program when a CRG is created. This example exit program (for Scenario A) sets the QCSTHAAPPI data area status flag to “Available (A)”.

```
Scenario A:* Create - called when Cluster Resource Group created. */

    /* Init haappo dataara status to available */
    setHaappiStatus(applicationAvailable);
    break;
```

Figure 9-2 Scenario A: Creating the application CRG exit program example (EXITPGMAPP)

Consider these additional functions for the exit program to perform:

- Copy the pertinent applications on all nodes in the recovery domain
- Prime all nodes in the recovery domain

Starting the Cluster Resource Group

When a CRG is started, the exit programs on each node in the recovery domain are called with an action code of 2 (Start). Figure 9-3 shows an example (Scenario B) of the data CRG exit program when a CRG is started. The exit program queries the node information to determine if this node is the primary or backup node:

- If on the primary node, it starts the remote journal.
- If on the backup node, it:

  a. Sets the QCSTHAAPPO flag to “Switch In Progress (I)”.
  b. Starts the apply process

  In most cases, this is a call to the cluster middleware data replication start procedure.

```
Scenario B /* Start - called when Cluster Resource Group started. */

     /* Get the nodeinformation. */
     if( (g.nodeInfo.node1Role == primary  &&
          0 == memcmp(g.nodeInfo.node1Name.c, ibmData->This_Nodes_ID, 8)) ||
         (g.nodeInfo.node2Role == primary  &&
          0 == memcmp(g.nodeInfo.node2Name.c, ibmData->This_Nodes_ID, 8)) )
     {
       if( 0 == memcmp(g.nodeInfo.node1Name.c, ibmData->This_Nodes_ID, 8) )
         STRRMTJRN(g.jrnName.objName.c, g.jrnName.libName.c,
                   g.nodeInfo.node2Name.c);
       else
         STRRMTJRN(g.jrnName.objName.c, g.jrnName.libName.c,
                   g.nodeInfo.node1Name.c);
     }
     else
     {
       /* Change the backup haappo dataara status to 'Switch In Progress’ */
       setHaappoStatus(dataSwitchInprogress);

       /* Start the apply process. */
       SBMAPYJOB(g.jrnName.objName.c, g.jrnName.libName.c, -1);
     }
     break;
```

Figure 9-3 Scenario B: Starting the CRG Data exit program example (EXITPGM)

Note: Even though a switch is not actually in progress, the status flag must be set to indicate that the data is not available on the backup system.

Figure 9-4 shows an example of the application CRG exit program (Scenario B) when a CRG is started. The exit program queries the node information to determine if this node is the primary or backup node:

- If it is on the backup node, it returns a successful completion code. No additional function is required.
- If it is on the primary node, it:
  a. Starts the application
  b. Sets the QCSTHAAPPI data area status flag to “Application In Progress (I)”.
  c. Loops until the QCSTHAAPPI data area status flag is set to “Available for Switch (A)”. In this example (Scenario B), the program waits for 30 seconds before checking the flag again.

```
Scenario B:   /* Start - called when Cluster Resource Group started. */

   /* If on primary, start application, set HAAPPI status. */
   if( (g.nodeInfo.node1Role == primary  &&
        0 == memcmp(g.nodeInfo.node1Name.c, ibmData->This_Nodes_ID, 8)) ||
       (g.nodeInfo.node2Role == primary  &&
        0 == memcmp(g.nodeInfo.node2Name.c, ibmData->This_Nodes_ID, 8)) )
   {
     STRORDENT();
     setHaappiStatus(applicationInProgress);
     while( getHaappiStatus() == applicationInProgress )
       sleep(30);
   }
   break;
```

Figure 9-4 Scenario B: Starting the CRG application exit program example (EXITPGMAPP)

Important: The application exit program must stay active at this point. When this exit program returns, Clustering Services issues an End CRG (action code 4) request to all nodes in the recovery domain.

Adding a node

When a node is added to a CRG, the exit programs on each node in the recovery domain called with an action code of 11 (Add node).

Handling the add node action code is similar to handling the initialize and start action codes. However, these actions are only handled on the node being added.

The data CRG exit program should perform these functions:

- Query the node role to see if this is the node being added:
  – If it is not the node being added, an *action successful* code is returned. No additional function is required.
  – If it is the node being added, follow these steps:
    i. Set the QCSTHAAPPO data area to a status of “Data is Available”.

ii. Create the journal environment.
iii. Journal the appropriate files.

- Query to see if the nodes status is active.

  If it's *Inactive*, an *action successful* code is returned. No additional function is required.

- Query the node information to determine if this node is the primary or backup node.

  – If it is on the primary node, start the remote journal.
  – If it is on the backup node, follow these steps:
  i. Set the QCSTHAAPPO flag to switch in progress (I).

  > **Important:** The status flag must be set to this value even though a switch is not actually in progress. This setting indicates that the data is not available on the backup system.

  ii. Start the apply process. In most cases, this is a call to the cluster middleware data replication start procedure.

Consider these additional functions for the exit program to perform:

- Create the resilient objects on all nodes in the recovery domain.
- Verify the objects are synchronized.

The application CRG exit program should set the QCSTHAAPPI data area status flag to “Available (A)”.

Consider these additional functions for the exit program to perform:

- Copy the pertinent applications on all nodes in the recovery domain.
- Prime all nodes in the recovery domain.

Changing the node

When a node is changed within a CRG, the exit programs on each node in the recovery domain is called with an action code of 13 (Change node). The exit programs are only called if the recovery domain is changed.

> **Note:** Do not call the application CRG with this action code.

9.2.6 Managing the environment

The exit program can be used to manage the recovery domain. Cluster Services calls the exit program as a result of:

- Application CRG or API activity, for example to initiate a switchover
- Activity not specifically initiated by the end user, for example to initiate a failover

Actions that should be handled by the exit program include:

- Controlling a planned switch
- Controlling a unplanned switch
- Ending a Cluster Resource Group
- Ending journaling if required
- Ending replication if required
- Ending the application

Ending a Cluster Resource Group

When a CRG is ended, the exit programs on each node in the recovery domain are called with an action code of 4 (End).

The data CRG exit program should query the node information to determine if this is the primary or backup node:

- If its on the primary node, end remote journaling.
- If its on the backup system, end the replication immediately.

> **Note:** The apply process (provided with cluster middleware code) must save its ending point so it knows where to start applying once the process is restarted.

The application exit program should end the application program.

Planned switch

When a planned switch occurs via the Initiate Switchover API, the exit program on each node in the recovery domain is called with an action code of 10 (Switchover). When both a data and application CRG are to be switched, the switching process must be synchronized between the two exit programs.

The synchronization process is handled by writing to and monitoring the status flags within the architected QCSTHAAPPI and QCSTHAAPPO data areas. See 4.7.5, “Exit programs” on page 47, for a discussion of the data areas.

> **Note:** When the exit program is called for a switchover, the roles of the nodes have been switched by cluster services. Therefore, the old primary system is now a backup node, and the old backup one node is the primary node.

Figure 9-5 shows an example of the data CRG exit program when the CRG switches (Scenario C). The exit program queries the node information to determine if this node is the primary or backup node:

- If it’s on the primary node:
  a. Wait until the application has been shut down. This occurs after the application CRG ends the application. In this case, a two-second wait occurs between checks.
  b. Send a journal entry to the journal of the old backup system. This journal entry is a user created entry which signifies to the apply process to end. Therefore, all transactions before this entry are applied before the apply process ends on the old backup system.
  c. End remote journaling.
  d. Set the QCSTHAAPPO flag to “Switch In Progress (I)”.

  > **Note:** At this point, a switch is no longer in progress. However, this status flag must be set to this value to indicate that the data is not available on the backup system.

  e. Start the apply process
- If it’s on the backup system:
  a. Wait until the apply process has processed all transactions and has ended. In this example, the apply process sets a flag in a data area indicating that it is done.
  b. Start remote journaling.

c. Set the QCSTHAAPPO flag to “Data Available (A)”.

```
Scenario C: /* Switchover (Stop replication on primary) */

  /* Get the node information */
  if( (g.nodeInfo.node1Role == primary  &&
        0 == memcmp(g.nodeInfo.node1Name.c, ibmData->This_Nodes_ID, 8)) ||
      (g.nodeInfo.node2Role == primary  &&
        0 == memcmp(g.nodeInfo.node2Name.c, ibmData->This_Nodes_ID, 8)) )
  {
     while( getHabpStatus() != acNone )
       sleep(2);

    /* Start remote journaling in the opposite direction. */
    if( 0 == memcmp(g.nodeInfo.node1Name.c, ibmData->This_Nodes_ID, 8) )
      STRRMTJRN(g.jrnName.objName.c, g.jrnName.libName.c,
              g.nodeInfo.node2Name.c);
    else
      STRRMTJRN(g.jrnName.objName.c, g.jrnName.libName.c,
              g.nodeInfo.node1Name.c);

    /* Set the 'new' primary side data status field */
    setHaappoStatus(dataAvailable);
  }
  else
  {
    while( getHaappiStatus() != applicationAvailable )
      sleep(2);

    /* create journal entry (UXI) */
    sendUXIEntry();

    /* End remote journal */
    if( 0 == memcmp(g.nodeInfo.node1Name.c, ibmData->This_Nodes_ID, 8) )
      ENDRMTJRN(g.jrnName.objName.c, g.jrnName.libName.c,
                g.nodeInfo.node2Name.c);
    else
      ENDRMTJRN(g.jrnName.objName.c, g.jrnName.libName.c,
                g.nodeInfo.node1Name.c);

    /* Set the 'new' apply side data status field */
    setHaappoStatus(dataSwitchInprogress);

    /* Start the apply process. */
    SBMAPYJOB(g.jrnName.objName.c, g.jrnName.libName.c, -1);
  }
  break;
```

Figure 9-5 Scenario C: Switchover CRG data exit program example (EXITPGM)

Figure 9-6 shows an example of the application CRG exit program (Scenario C) when the CRG switches. The exit program queries the node information to determine if this node is the primary or backup node:

- If it’s on the primary node:

  a. End the application
  b. Set the QCSTHAAPPI data area status flag to “Application Available (A)”.

- If it’s on the backup node:
  a. Wait until the QCSTHAAPPO data area status flag is set to “Data Available (A)”. This indicates that the data is quiesced, and the applications can start up on the old backup system.
  b. Start the application
  c. Set the QCSTHAAPPI data area status flag to “Application in Progress (I)”.

```
Scenario C: /* Switchover - planned switch */

   /* If on primary, end application, and set haappi status. */
   if( (g.nodeInfo.node1Role == primary  &&
        0 == memcmp(g.nodeInfo.node1Name.c, ibmData->This_Nodes_ID, 8)) ||
       (g.nodeInfo.node2Role == primary  &&
        0 == memcmp(g.nodeInfo.node2Name.c, ibmData->This_Nodes_ID, 8)) )
   {
     /* Else must be backup, wait til data is flushed. */
     while( getHaappoStatus() != dataAvailable )
       sleep(2);

     /* Restart application. */
     STRORDENT();

     /* Set the 'new' primary side data status field */
     setHaappiStatus(applicationInProgress);
   }
   else
   {
     ENDORDENT();
     setHaappiStatus(applicationAvailable);
   }
break;
```

Figure 9-6 Scenario C: Switchover CRG application exit program example (EXITPGMAPP)

Unplanned switch (failover)

When a cluster resource detects a node failure or resource failure, the exit programs on each node in the recovery domain are called with an action code of 9 (failover).

9.2.7 Rejoining or restarting the application

Restarting an application can be the direct result of an application failure. The application CRG allows for the situation where the application attempts to be restarted (up to three times) on the node currently acting as primary.

When a node defined to a CRG comes back online after a node or resource failure, an action code of 8 (Rejoin) is sent to all exit programs within the recovery domain. The data CRG exit program for a rejoin action should query the node information to determine if this node is the primary or backup node:

- If it’s on the primary node, return, nothing to do (should never get called).
- If it’s on the backup node:
  a. Resynchronize the data.
  b. If the node status is active, start the replication process.

The application CRG exit program should query the node information to determine if this node is the primary or backup node:

- If it’s on the primary node, if the node status is active, start the application.
- If it’s on the backup node, return, nothing to do.

10

Sample cluster environment and applications

This chapter illustrates the use of the cluster commands that are available with the V5R1 QUSRTOOLs library. The setup and administration of a cluster configuration is typically performed through a Cluster Management GUI interface, as provided either by a cluster middleware provider or the IBM Simple Cluster Management GUI (V5R1 Operations Navigator).

To illustrate the concepts in this redbook from an application perspective (in particular, Chapter 9, “Making applications continuously available” on page 169), this chapter defines the setup of a three-node cluster that allows applications on two systems to be backed up on a common backup system. Four Cluster Resource Groups are created to handle this configuration. This chapter also discusses changes to a sample order entry application first to support remote journals and then to make the application more highly available.

To allow data resilience across the systems, remote journaling is used.

> **Note:** The application code to support high availability, as represented in this chapter, is provided to serve as an *example* only. The code does not represent a real application. you can find additional examples in the redbook *Moving Applications to Switchable Independent ASP*s, SG24-6802, which is scheduled for publication later in the second half of 2002.

10.1 Example cluster configuration

The example cluster shown in Figure 10-1 has three systems, with names of M20, M27, and M22. The plan for availability is to have an application from System M20 backed up on System M22 and to have an application from System M27 backed up on System M22. These three systems comprise a cluster by the name of CLUSTERA.

The Cluster Resource Group defining the *data recovery domain* for Systems M20 and M22 is named CRG1. The Cluster Resource Group defining the *application recovery domain* for Systems M20 and M22 is named CRG1APP.

Similarly, the Cluster Resource Group defining the data recovery domain between System M27 and M22 is named CRG2. And the Cluster Resource Group defining the application recovery domain for Systems M27 and M22 is named CRG2APP.

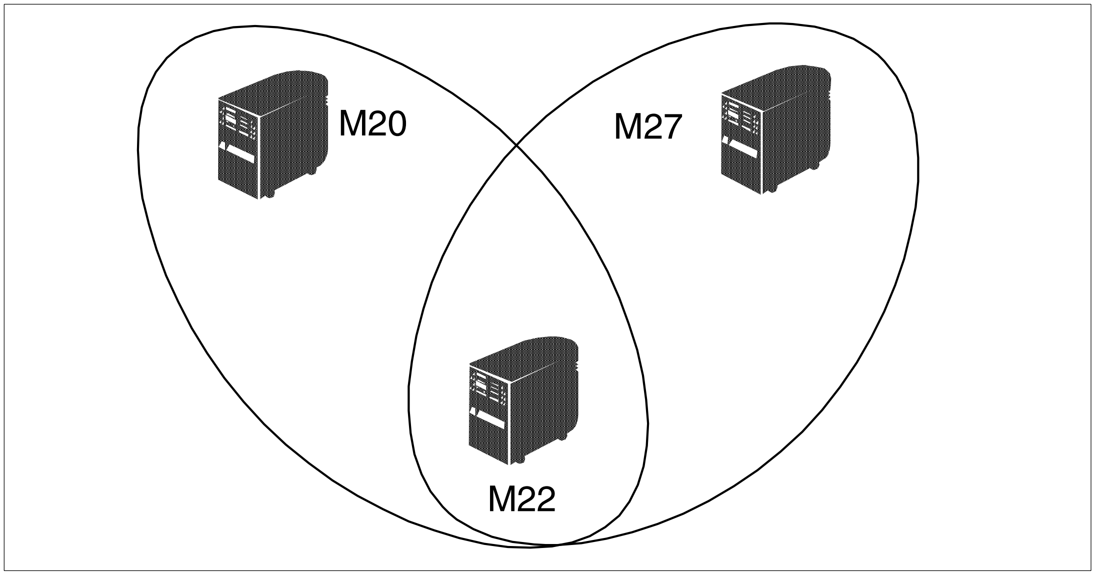


Figure 10-1 Application cluster CLUSTERA

Before the clustering environment is setup, a link between the systems is required to propagate the cluster setup commands. In the example illustrated in Figure 10-2, the three systems were set up on a common LAN. TCP/IP is started for the systems.

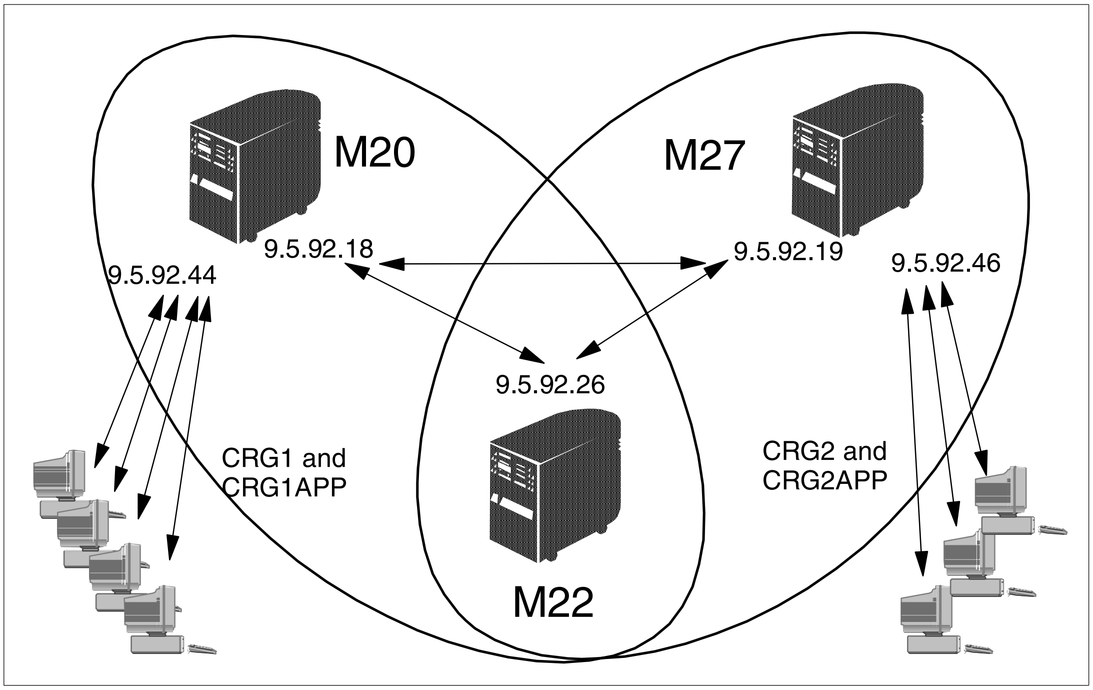


Figure 10-2 Assignment of IP addresses in cluster CLUSTERA

The three systems are setup with the following IP addresses for the primary system access and cluster control:

- System “M20” on IP address 9.5.92.18
- System “M27” on IP address 9.5.92.19
- System “M22” on IP address 9.5.92.26

The IP address that users of application 1 are to use is address 9.5.92.44. The IP address that users on application 2 are to use is 9.5.92.46.

Address 9.5.92.44 is defined on both the primary system (M20) and the backup system (M22). Address 9.5.92.46 is defined on both the primary system (M27) and the backup system (M22). Each of these addresses is started only on the system serving as the primary for the application.

Note: The IP addresses of 9.5.92.44 and 9.5.92.46 must be available addresses on the network. There must be no devices currently using them. If these addresses are in use elsewhere in the network, Cluster Resource Services cannot setup the application Cluster Resource Groups.

10.1.1 Creating the sample clustering environment

For this example, the CRTCLU, CRTCRG, STRCLUNOD, and STRCRG cluster commands available with V5R1 QUSRTOOL are used to setup and control the cluster environment. These commands call the system APIs listed in Table B-3 on page 286 and Table B-4 on page 287.

High availability business partners have a GUI utility to serve this purpose, or they use a different form of calling the cluster APIs.

Example cluster setup

Figure 10-3 through Figure 10-5 show the command and parameters used in the setup of the sample three node cluster described in 10.1, “Example cluster configuration” on page 184, and illustrated in Figure 10-1 on page 184.

```
/* Setup cluster environment for three node network   +
          between M20 and M22 with M20-primary M22-backup +
      and between M27 and M22 with M27-primary M22-backup */
             PGM
             CHGJOB     LOG(4 10 *SECLVL) LOGCLPGM(*YES)
/*   Create three node cluster */
             CRTCLU CLUSTER(CLUSTERA) NODE('9.5.92.18'/M20 +
                          '9.5.92.19'/M27 '9.5.92.26'/M22)
/* Start TCP server INETD on all nodes    */
/*           STRTCPSVR SERVER(*INETD)    */
             STRCLUNOD  CLUSTER(CLUSTERA) NODE(M20)
             MONMSG     MSGID(CPFBB05) EXEC(GOTO CMDLBL(STRCLUERR))
             STRCLUNOD  CLUSTER(CLUSTERA) NODE(M27)
             MONMSG     MSGID(CPFBB05) EXEC(GOTO CMDLBL(STRCLUERR))
             STRCLUNOD  CLUSTER(CLUSTERA) NODE(M22)
             MONMSG     MSGID(CPFBB05) EXEC(GOTO CMDLBL(STRCLUERR))
             GOTO       CMDLBL(NEXT1)
StrCluErr:
             SNDPGMMSG  MSG('Error on Start Cluster Node - Check +
                          Joblog for details') TOPGMQ(*EXT)
             GOTO       CMDLBL(ENDPGM)
```

Figure 10-3 Example cluster setup (Part 1 of 3)

The first cluster command listed in Figure 10-1, Create Cluster (CRTCLU), creates the three node cluster for nodes M20, M27, and M22.

Since multiple cluster nodes are defined by CRTCST rather than individual use of the Add Cluster Node Entry (ADDCLUNODE) command, the nodes are not automatically started. The start indicator parameter is ignored on the CRTCLU command. Therefore, the STRCLUNOD command must be run for each node.

The CRTCLU command gives each node an ID name and interface address (IP address). This IP address is used by Cluster Resource Services to communicate with other nodes within the cluster. It cannot be the IP address used by the application users. A second interface address can be specified for backup use by Cluster Resource Services, but it is not to be used by the application users either.

The systems defined as cluster nodes must not be setup with the Allow Add to Cluster (ALWADDCLU) system network attribute set to *None. Use the Change Network Attribute (CHGNETA) command to change this attribute.

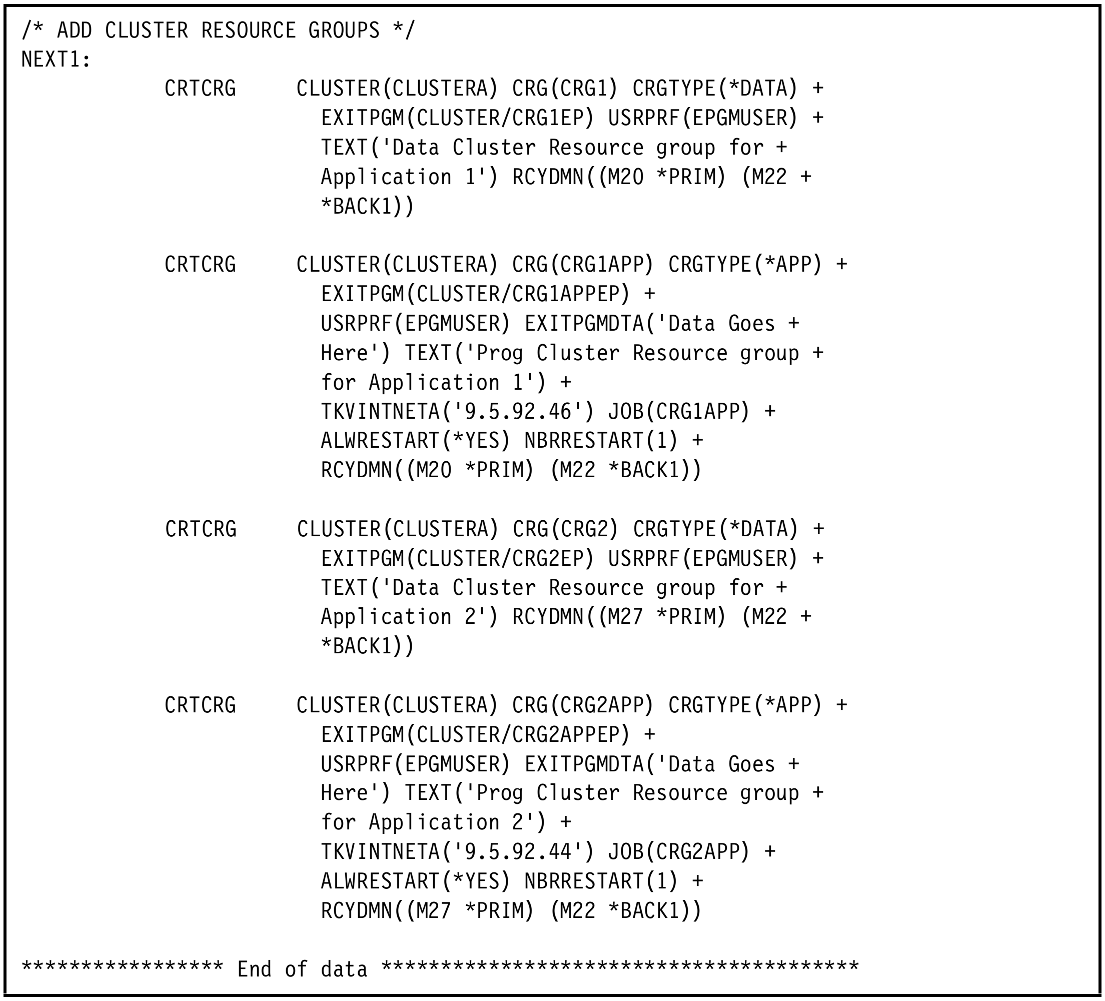
As with other OS/400 commands, the Monitor Message (MONMSG) command allows for verification of the desired action. In our example, feedback message CPFBB05 indicates that the cluster node could not be started. CPFBB05 occurred several times.

This failure to start the remote cluster node can occur because the remote system does not have the INETD server running. The solution is to run the STRTCP SERVER(*INETD) command on the remote system.

The output of the List Cluster Information (PRTCLUINF) command gives useful information about the definitions and status of the cluster. Use PRTCLUINF to verify the cluster name with the status of the individual nodes within the cluster.

Example resource group setup

Figure 10-4 shows the commands used to define the recovery domains for the example configuration.



```
/* ADD CLUSTER RESOURCE GROUPS */
NEXT1:
           CRTCRG     CLUSTER(CLUSTERA) CRG(CRG1) CRGTYPE(*DATA) +
                        EXITPGM(CLUSTER/CRG1EP) USRPRF(EPGMUSER) +
                        TEXT('Data Cluster Resource group for +
                        Application 1') RCYDMN((M20 *PRIM) (M22 +
                        *BACK1))

           CRTCRG     CLUSTER(CLUSTERA) CRG(CRG1APP) CRGTYPE(*APP) +
                        EXITPGM(CLUSTER/CRG1APPEP) +
                        USRPRF(EPGMUSER) EXITPGMDTA('Data Goes +
                        Here') TEXT('Prog Cluster Resource group +
                        for Application 1') +
                        TKVINTNETA('9.5.92.46') JOB(CRG1APP) +
                        ALWRESTART(*YES) NBRRESTART(1) +
                        RCYDMN((M20 *PRIM) (M22 *BACK1))

           CRTCRG     CLUSTER(CLUSTERA) CRG(CRG2) CRGTYPE(*DATA) +
                        EXITPGM(CLUSTER/CRG2EP) USRPRF(EPGMUSER) +
                        TEXT('Data Cluster Resource group for +
                        Application 2') RCYDMN((M27 *PRIM) (M22 +
                        *BACK1))

           CRTCRG     CLUSTER(CLUSTERA) CRG(CRG2APP) CRGTYPE(*APP) +
                        EXITPGM(CLUSTER/CRG2APPEP) +
                        USRPRF(EPGMUSER) EXITPGMDTA('Data Goes +
                        Here') TEXT('Prog Cluster Resource group +
                        for Application 2') +
                        TKVINTNETA('9.5.92.44') JOB(CRG2APP) +
                        ALWRESTART(*YES) NBRRESTART(1) +
                        RCYDMN((M27 *PRIM) (M22 *BACK1))

***************** End of data ****************************************
```

Figure 10-4 Example cluster setup (Part 2 of 3)

The Create Cluster Resource Group (CRTCRG) command is run once for each of the four Cluster Resource Group objects to be setup. The Cluster Resource Groups define the recovery domain for the applications. The implied recovery domain is defined by the nodes within the Cluster Resource Group.

In this example, the two Cluster Resource Groups named CRG1 and CRG1APP define a recovery domain of nodes M20 and M22. The preferred mode of M20 is as primary node. The preferred mode for M22 is as backup node. CRG1 defines a data resource group. CRG1APP defines the application resource group.

When the CRTCRG command is run, it attempts to start the related exit program to pass the initialize action code. If the defaults are used, the API submits a job to the QBATCH job queue to start the exit program. The CRTCRG command then waits for confirmation that the exit program has handled the initialize action code.

If both of these jobs are submitted to a job queue with a maximum activity of one, the CRTCRG waits forever. Therefore, it is a good idea to run the CRTCRG either interactively or from a separate job queue.

Example startup of recovery domains

Figure 10-5 shows the commands to start the data and application recovery domains.

```
 Pgm:
             STRCRG     CLUSTER(CLUSTERA) CRG(CRG1)
             STRCRG     CLUSTER(CLUSTERA) CRG(CRG1APP)
             STRCRG     CLUSTER(CLUSTERA) CRG(CRG2)
             STRCRG     CLUSTER(CLUSTERA) CRG(CRG2APP)
EndPgm:
             ENDPGM
```

Figure 10-5 Example cluster setup (Part 3 of 3)

The List Cluster Resource Groups (PRTCLUINF) command provides information about the cluster resource definitions. It provides information on the number, names, and status of the individual resource groups.

The List Cluster Resource Group Information (PRTCRGINF) command produces a report to verify the detailed definitions of the resource group. This provides the current information on the individual resource groups as defined in the create and succeeding change commands.

10.1.2 Sample setup of journaling

Figure 10-6 through Figure 10-8 display a Command Language (CL) program written to journal files.

```
/* STRJRNFLS Starts journaling the files needed for the Order Entry Application +
     If journaling already exists then remove it and start again */

             PGM
             DCL        VAR(&APPLIB) TYPE(*CHAR) LEN(10) +
                          VALUE('AVAIL2')
             DCL        VAR(&JRNLIB) TYPE(*CHAR) LEN(10) +
                          VALUE('AVAIL2')
             DCL        VAR(&JRN) TYPE(*CHAR) LEN(10) VALUE('JRN')
             DCL        VAR(&JRNRCV) TYPE(*CHAR) LEN(10) +
                          VALUE('JRNRCV0000')

/* Check to see if journal is defined */
             CHKOBJ     OBJ(&JRNLIB/&JRN) OBJTYPE(*JRN)
             MONMSG     MSGID(CPF9801) EXEC(GOTO CMDLBL(BUILDNEW))

/* If journal exists stop journaling and remove objects */
             ENDJRNPF   FILE(*ALL) JRN(&JRN)
             DLTJRN     JRN(&JRNLIB/&JRN)
             DLTJRNRCV  JRNRCV(&JRNLIB/&JRNRCV)
/* Remove remote journal if it exists */
             RUNRMTCMD  CMD('DLTJRN JRN(&JRNLIB/JRN)') +
                          RMTLOCNAME(AS22 *IP) RMTUSER(USERAAA)
```

Figure 10-6 Sample journal setup (Part 1 of 3)

```
/* Add remote journal to Backup system */
             ADDRMTJRN  RDB(AS22) SRCJRN(&JRNLIB/JRN) TEXT('Remote +
                          Journal on M22')
/* Start Remote journal on Backup system */
             CHGRMTJRN  RDB(AS22) SRCJRN(&JRNLIB/JRN) JRNSTATE(*ACTIVE)
/* Add files to be jounaled to journal */
             STRJRNPF   FILE(&APPLIB/STOCK) JRN(&JRNLIB/&JRN) +
                          OMTJRNE(*OPNCLO)
             STRJRNPF   FILE(&APPLIB/CSTMR) JRN(&JRNLIB/&JRN) +
                          OMTJRNE(*OPNCLO)
             STRJRNPF   FILE(&APPLIB/DSTRCT) JRN(&JRNLIB/&JRN) +
                          OMTJRNE(*OPNCLO)
             STRJRNPF   FILE(&APPLIB/ORDERS) JRN(&JRNLIB/&JRN) +
                          OMTJRNE(*OPNCLO)
             STRJRNPF   FILE(&APPLIB/ORDLIN) JRN(&JRNLIB/&JRN) +
                          OMTJRNE(*OPNCLO)
             STRJRNPF   FILE(&APPLIB/USERSTATUS) JRN(&JRNLIB/&JRN) +
                          OMTJRNE(*OPNCLO)
EndPgm:
             ENDPGM
```

Figure 10-7 Sample journal setup (Part 2 of 3)

```
/* Build a new environment and start journaling the files */
BuildNew:
             CRTJRNRCV  JRNRCV(&JRNLIB/&JRNRCV)
             MONMSG     MSGID(CPF0000) EXEC(DO)
             SNDPGMMSG  MSG('Journal Receiver' *BCAT &JRNLIB *TCAT +
                          '/' *TCAT &JRNRCV *BCAT 'was not built') +
                          TOPGMQ(*EXT)
             GOTO       CMDLBL(ENDPGM)
             ENDDO

             CRTJRN     JRN(&JRNLIB/&JRN) JRNRCV(&JRNLIB/&JRNRCV)
             MONMSG     MSGID(CPF0000) EXEC(DO)
             SNDPGMMSG  MSG('Journal' *BCAT &JRNLIB *TCAT '/' *TCAT +
                          &JRN *BCAT 'was not built') TOPGMQ(*EXT)
             GOTO       CMDLBL(ENDPGM)
             ENDDO

/* Add remote journal to Backup system */
             ADDRMTJRN  RDB(AS22) SRCJRN(&JRNLIB/JRN) TEXT('Remote +
                          Journal on M22')
```

Figure 10-8 Sample journal setup (Part 3 of 3)

10.1.3 Journal environment to allow reverse remote backup

When setting up the journal environment, it is desirable to use the same library and object names on both systems of a cluster. This allows the CRG exit programs and any programs they call to be the same on both the primary and backup systems.

With the arrangement described in this section, the roles of the two systems can be reversed. The same files and journal names can appear in the same libraries. The journaling environment shown in Figure 10-9 and Figure 10-10 explains how this works.

> **Note:** The remote journal must be setup as a remote journal type 2 to allow the remote journal to be in a different library than the local journal it duplicates.

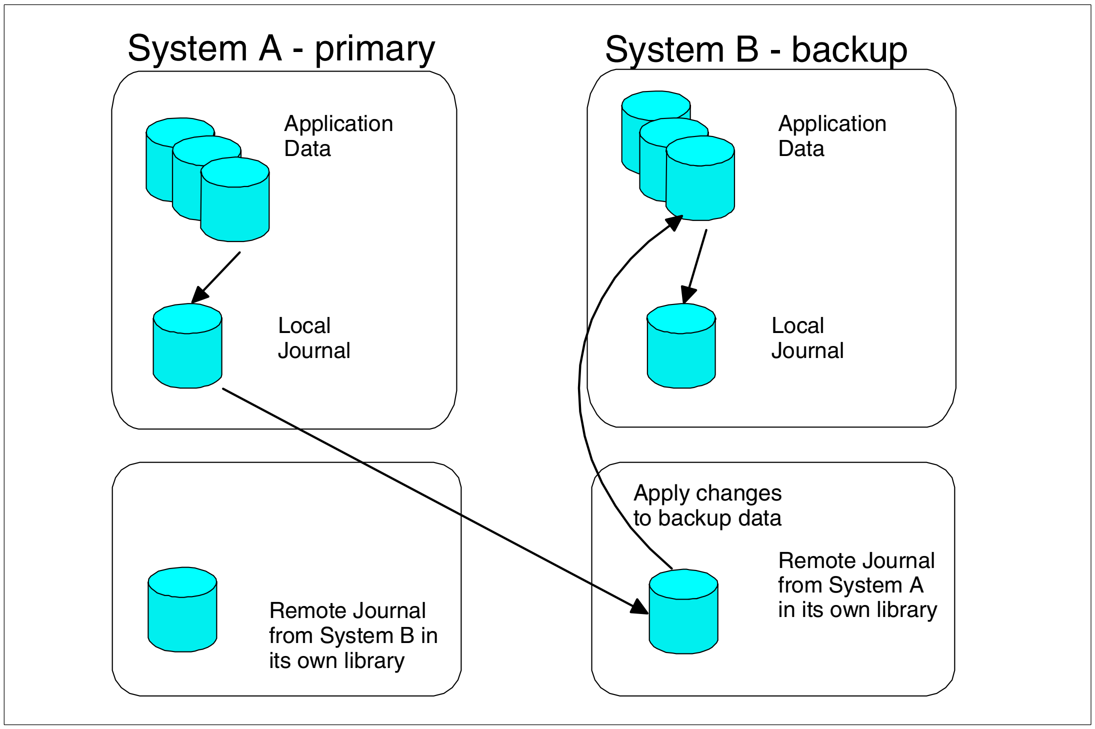

System A - primary
System B - backup
Application Data
Local Journal
Application Data
Local Journal
Apply changes to backup data
Remote Journal from System A in its own library
Remote Journal from System B in its own library

Figure 10-9 Resilient data from System A to B

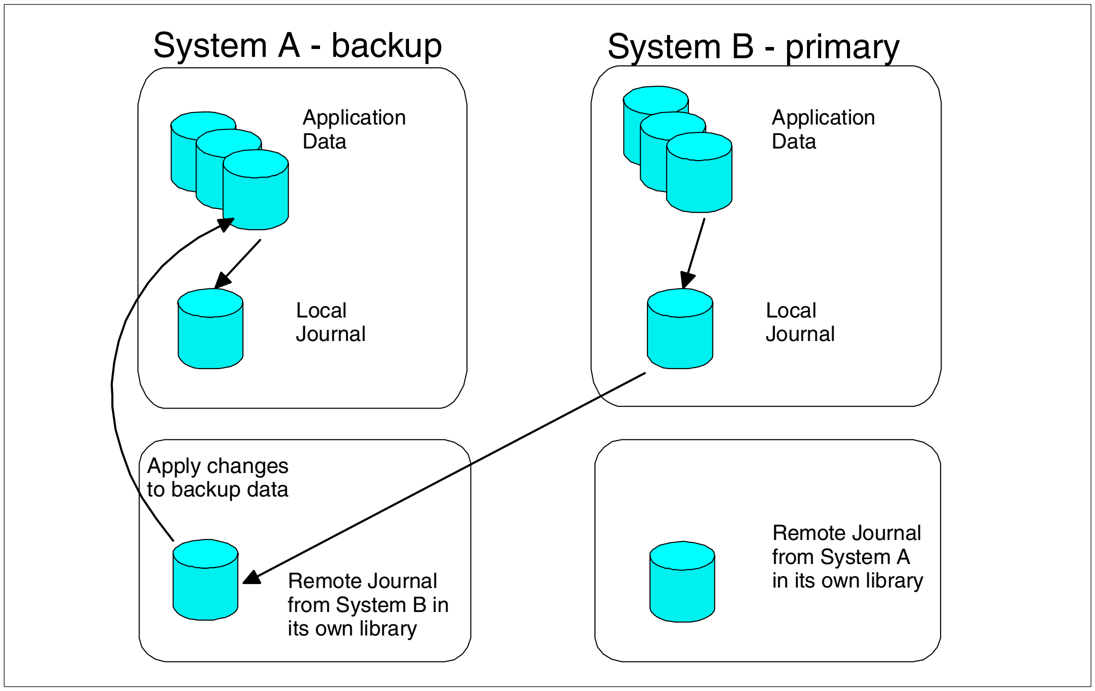

System A - backup
System B - primary
Application Data
Local Journal
Application Data
Local Journal
Apply changes to backup data
Remote Journal from System B in its own library
Remote Journal from System A in its own library

Figure 10-10 Resilient data from System B to A

10.2 Sample RPG order entry application

To illustrate the coding required for an application to support high availability, this section describes a sample order entry program. For this application, high availability is implemented in two stages:

- Stage 1: Cluster application to use remote journals describes the changes to allow the application to take advantage of the iSeries cluster features.
- Stage 2: Cluster application to support a highly available environment outlines further changes to the application that make it more highly available, the data more reliable and also provide a more finite restart capability.

10.2.1 Original RPG application

This sample order entry application is designed to allow multiple users to create orders for customers. It validates the customer number and credit balance, validates the item number and stock balance, assigns a sequential order number, creates an order header record, creates an order detail record for each item sold, updates the customer account, and passes off a trigger to a batch print program to print the pick/pack slip.

The application updates the order number at the start of building an order and retains the order detail information in memory until the order is eventually closed. At this time, the order files, stock files, and customer file are updated. Therefore, an interruption of the system can cause the loss of any detail lines entered before the order closes.

There is no check for interference from one user to another. This means that if more than one order taker is working with the same inventory item at the same time, the first one to close their order updates inventory from stock. This potentially leaves the second order taker in an out of stock situation, which does not show up until the order is closed.

The sample application does not use journaling or commit control and does not provide for system integrity of the data files.

In its original design, the trigger from the order program to the print program is a data queue. Prior to V5R1, data queues are not automatically captured in a journal. Therefore, if the journaled data is moved to a second system in a backup apply process, the integrity of the print trigger data queue is not maintained.

With V5R1, the data queues can be journaled, and adding journaling to the application for data integrity protects the data queue trigger. However, this example does not use journaling of the data queue, but illustrates an alternate solution.

10.2.2 Stage 1: Cluster application to use remote journals

To provide for a more available backup, remote journaling is used to assist the replication of data to the backup system. The data queue used in the original program as the print trigger, is replaced by a logical view of unprinted orders based on a status in the order header file. Therefore, the information passed by the data queue is moved to the order file to allow replication on the remote database.

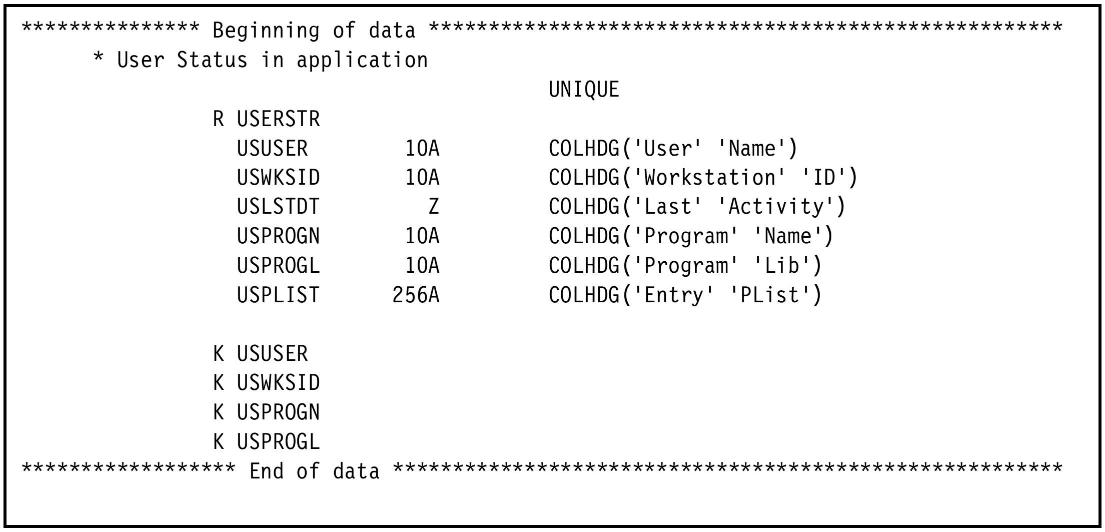
As a first step in keeping the database of the backup system in synchronization with the primary system, the apply journal change program uses the Retrieve Journal Entry (RTVJRNE) command (or equivalent system API) on the remote backup system.

Clustering support is used to detect a failover situation and trigger a rollover from the primary to the backup system. The Cluster Resource Group exit programs are written to control the Cluster Resource Groups that are defined.

Stage 1 application changes to support remote journals

To support remote journaling, these changes are required at the first stage:

- Remote journaling is started on the database files.
- A user status file is created to allow restart from within the active programs in use by the user. Figure 10-11 shows the DDS specifications used to create this user status log file.



```
*************** Beginning of data *****************************************************
      * User Status in application
                                            UNIQUE
                R USERSTR
                  USUSER        10A         COLHDG('User' 'Name')
                  USWKSID       10A         COLHDG('Workstation' 'ID')
                  USLSTDT         Z         COLHDG('Last' 'Activity')
                  USPROGN       10A         COLHDG('Program' 'Name')
                  USPROGL       10A         COLHDG('Program' 'Lib')
                  USPLIST      256A         COLHDG('Entry' 'PList')

                K USUSER
                K USWKSID
                K USPROGN
                K USPROGL
****************** End of data ********************************************************
```

Figure 10-11 Definition of the StatusLog file

Functions provided in programs that are to handle this design change include:

- The application must log into this *StatusLog* file at each potential point it from which it is required to restart.
- At the logical end of the program, the program clears out the log status file for itself.
- A service module is written to log the current status of the application program. It is called at various stages of the application to log the development through the order entry for a customer order.
- For the order entry program, the status log module is called when an order number is assigned to the customer and an order header record written. The status is updated when the order is complete. A log is generated for the customer maintenance program, at the time of starting to work with customer data, then again when the customer update is complete.
- To complement the status log module, a second module is written to check for outstanding entries in the log file and determine if a restart is required for one or more of the application programs.
- The log saves the *entry plist parameters that may be passed back to the application program in a restart mode.
- An initial program is written for the users that check this status log. If there are entries, the appropriate program is called with the saved parameters.
- The application programs are modified to use the passed *Entry parameters as the starting point for the respective user.

- A program is called by the application exit program to apply the remote journal changes.
- The programs are changed to allow them to restart part way through the process. For example, the order entry program accepts entry parameters that bypass the first data entry screens if the order is currently open and incomplete. The program sets up the display and working data to the same point as the last invocation for the open order and customer.
- A flag is added to the order header file to indicate the print status. This flag is used by the printer to control printing of the order. The data queue processing is removed.

10.2.3 Stage 2: Cluster application to support a highly available environment

To provide for a higher degree of data integrity, commitment control is added to the application. The application implication of this change is that a more discrete commit boundary is implemented. By changing the logical unit of work (LUW), or the commit boundary, to a detail line item on the order, and changing the order entry program to restart part-way through the processing, the user is allowed to continue with the order after the last order line commits to the database.

These changes add features to the program that have advantages other than a disaster restart. The application program can now be a called program from other steps within the application, to continue with the order processing for other reasons. By processing a commit at the order detail time, the order taker can ensure that the data is still valid by the time the order is complete.

To implement these changes, the application must be further changed from stage 1 to:

- Commit after each order detail line
- Add a function to allow backing out of previously committed stages, allow a cancel of committed order lines, or cancel of all committed items for the order
- Start and continue working with incomplete orders

11

Considerations when planning for iSeries clusters and recoverable applications

There are strong benefits to implement high availability. But to achieve success can be a long process. A typical high availability installation can take a day to complete. But the planning, configuration, customization, training, and implementation of new operational procedures can take several months. The success of the project depends on the many factors, each based on management support, and planning is critical.

Once the impact of downtime is understood, develop a business continuity plan. Gather input from everyone from the CEO to network administrators, application programmers, and end users. Each employee must be aware of the methods to avoid and minimize a disruption to the business.

This chapter discusses many of the areas to investigate when planning for an iSeries cluster solution. It presents considerations when developing an implementation plan, for application support of clustering, and planning for the ongoing management of a cluster.

The intent of the material that is provided is to provoke thoughts and to initiate actions to represent as tasks in a clustering implementation plan. These actions are not easy to objectively define. As such, the information provided within this chapter is not by itself a project plan, but is rather a checklist of important considerations for a successful clustering implementation.

Tip: The tasks necessary to implement a clustering and independent ASP on the iSeries server are unique to the customer, based on where a customer's high availability solution is positioned. A project plan can be built using the information in this redbook, as well as in:

- *Roadmap to Availability on the iSeries 400,* REDP0501
- *High Availability on the AS/400 System: A System Manager's Guide,* REDP0111

11.1 Planning for iSeries clustering

It is important to carefully plan for the implementation of a clustering solution. Planning helps prepare for the technical installation of the cluster setup and facilitates a smooth installation. The planning tasks depend on the type of cluster to be created.

In general, make sure the planning process includes these key activities:

- Thoroughly understanding the concepts of the proposed clustering solution
- Clearly defining what is to be accomplished by implementing clusters or independent ASPs on the iSeries server
- Obtaining a service level agreement. Agree on service hours, planned downtime, and so forth. Clearly set and document the correct level of expectations.

In particular, the setup and management of a clustered environment is a nontrivial task. The tasks depend on which type of cluster you want to create. In general, the tasks involved are:

- Initial configuration
  a. Specify systems in a cluster
  b. Create Cluster Resource Groups (CRGs) for data and application
  c. Create device domains for resilient devices
  d. Activate the cluster
- Configuration modifications (as required)
  a. Add or remove a node to or from the cluster
  b. Create new CRGs or delete existing CRGs
  c. Change the property of a CRG, for example primary or backup, or a change in the order
  d. Identify resilient resources
  e. Start and end a node or CRG
- Application requirements
  a. Modify application to work with clusters
  b. Enable restart features for interactive, batch, and client-server jobs
  c. Modify procedures to allow for application maintenance in a clustered environment
- Administration and management tasks
  a. Display cluster topology and cluster resources
  b. Display CRG contents
  c. Initiate switchover
  d. Delete cluster configuration
- Systems management
  a. Obtain service level agreements with service providers
  b. Implement a capacity and performance plan
  c. Implement standards for security and user profiles

Tip: The investment in skills is significant and cannot be achieved quickly. Consider developing skills early in the clustering implementation.

Many of the planning tasks involved for implementing a clustering solution are further described in this section.

11.1.1 Measuring the impact of an outage to the business

To understand the value of a high availability clustering solution, it is helpful to understand the impact of a computer outage to a business. It is important to determine an estimated cost of downtime. What direct and indirect costs are affected by the business when the system (or one of its components) is not available?

Investigate what system outages have occurred over the past couple of years. Include both planned and unplanned outages. Obtain an estimate of the cost of these outages.

Separate outage costs into two categories:

- Tangible losses, which include:
  - Loss to share holdings and profits
  - Losses incurred through product or campaign delays
  - Employee idle time waiting for the system (or application) to become available
  - Employee overtime to enter transactions not entered during the outage
  - Cost of facilities and equipment during the idle period
  - Consequential loss through penalties charged by customers or suppliers for delayed delivery
  - Goods lost through damage or aging
- Intangible losses, which include:
  - Credibility in the marketplace
  - Lost revenue when customers buy elsewhere
  - Market share

Once these costs are tallied, other scenarios can be planned. Calculate an estimated cost for the loss of each application. This helps set the priority of applications.

11.1.2 Determining the desired level of availability

Once the financial justification of availability of the business is established, decide the level of availability that the business can afford. Do not be deterred by the results of the analysis. If the desired level of availability cannot be achieved at first, consider implementing a tactical solution before moving to the more strategic solution.

An example of a tactical solution is to run with an application that has a basic ClusterProven status only. A switchover may not be seamless in this scenario. If there are 5250 devices, the end users need to sign on to the backup system. However, when these users open their application session, they are positioned back to the menu they were last.

With this tactic, after an outage, the I/T group needs to establish the integrity of data before the application and users start performing updates.

Although a tactical or practical solution is not the ideal, it may yet be a better or more structured solution than what currently exists in the organization. A tactical solution can be implemented relatively quickly, and provide high availability while a strategic solution is developed.

A strategic solution can be to implement applications that meet Advanced ClusterProven requirements. An Advanced ClusterProven status demands that commitment control for application rollback, or a similar function is implemented.

An Advanced ClusterProven solution takes longer to develop, depending on the recoverability of the existing application.

Tip: Start simple and build the solution.

Remember that the more systems that are in the proposed cluster solution, the more complex the implementation is. Nodes can always be added to the cluster. Additional Cluster Resource Groups related to other applications and data sets can always be started after the initial cluster is setup.

11.1.3 Understanding the required configuration of a cluster

To determine the configuration required to implement clustering, it is important to understand the roles, location, and size of the systems (nodes) to be in the cluster.

If there are a number of iSeries servers in the business, decide how much of the operational environment and application is to be replicated. In the event that the system and one key application is replaced, the business can be supported by a simple cluster solution. Use the benefits by this simple cluster to develop experience and skills to support a more complete cluster. The remaining hardware and application can be moved to the cluster environment at a later time.

An understanding of the business' infrastructure is required when implementing clustering. The key tasks that are involved are to:

- Perform an inventory of all installed iSeries server hardware, including:
  - Processor
  - Storage
  - Disk unit, capacity, and utilization
  - IOPs
  - IOAs
- Determine the operating system release level installed, identify the applications involved, and determine the release level and compatibility of all software products.
- Document other related systems that are not to be part of the cluster, yet affect the same sphere of business operation as the cluster.

  These related systems can be other systems (servers or clients) plus their operating systems, databases and applications; network hardware (LAN, Internet Service Providers, routers, topology); and peripheral devices (tapes, printer, and displays).
- Select the applications that are to be clustered.
- Decide the most appropriate location (node) to run these applications.
- Size the systems at these locations to determine if there is spare capacity to implement the availability features and the resilient application under all conditions. Decide where any additional capacity is needed.

  With today's price per performance structure, it is not uncommon to find production systems running 30% utilized. In these cases, the production machine may have plenty of capacity to run additional recovery features.
- Ensure the capacity of the backup system is sufficient to handle any application load that may be moved to it in the event of data or application failover or switchover.

11.2 Making applications recoverable

Making applications recoverable is key to enable high availability solutions for the iSeries customer. The ability of a program to recover from an outage involves a change of the design or coding of the program, to support recoverability.

A discussion of application design and considerations, and user recovery both in interactive and batch jobs, is found in the Redpaper *High Availability on the AS/400 System: A System Manager's Guide,* REDP0111. You can also learn more about application considerations in the redbook *Moving Applications to Switchable Independent ASP*s, SG24-6802, which is scheduled for publication later in the second half of 2002.

Whether an application is developed in-house, or offered by an ISV, it is a worthwhile, yet not a difficult task, to cluster-proof applications. ClusterProven applications are recoverable. And there are resources available to assist with cluster-proofing applications. Refer to Chapter 8, "ClusterProven for iSeries applications" on page 161, and Appendix C, "iSeries cluster resources" on page 291, to understand the support and efforts involved.

Detailed programming recommendations are beyond the scope of this redbook. This section offers general guidelines for enabling the recovery characteristics of a typical application.

11.2.1 Application object inventory

To prepare for ClusterProving an application, first look at the applications that are running throughout the business. Consider both iSeries applications and applications running on other platforms. Determine for which areas of the business you want to require continuous availability. Determine which applications to make recoverable. Changes may be required to applications on the iSeries server and on the systems with which the iSeries server interfaces.

Determine what data is to be resilient. Build an inventory of the objects used by the selected applications that need to be on the backup system in the event of an application failover or switchover. Add to this list any other objects that are to be on the backup system.

For iSeries applications, there are several objects types that require special handling, for example:

- Temporary files
- Data spaces
- Data queues

Cluster middleware providers and ISV programmers are proficient with the special handling procedures to replicate these types of objects. Work with these IBM partners for a replication solution.

11.2.2 Resilient data

Resilient data is data that survives a switchover.

When planning a clustering installation, establish which objects on which nodes are to be resilient. This can be done by completing the inventory procedure, as described in 11.2.1, "Application object inventory" on page 199. Once these objects are identified, replicate the objects. This replication involves journaling and replication of objects between two or more systems in the cluster.

Enter these resilient objects into the object specifier file associated with the QCSTHAPPI data area for the application. In this way, the cluster management tool can automatically create data resilience as represented by a data CRG, and then set up the replication or switched disk environment.

The layout of an object specifier file can be found in B.4, “Object specifier file layout” on page 288.

Contact a cluster middleware provider, as identified in Part 3, “Cluster middleware business partners” on page 227, for middleware software to replicate objects.

11.2.3 Resilient applications

Resilient applications are applications that survive a switchover.

When implementing a clustering solution, decide the level of availability for each application. Select an application that is ClusterProven. Decide whether Basic or Advanced ClusterProven is required.

For applications developed in-house, create a development plan for any modifications required to meet the ClusterProven criteria for high availability. Review “Criteria to obtain ClusterProven trademark” on page 165 for more information.

Plan the recovery domains and the IP-takeover addresses to be related to application CRGs. Decide which nodes the applications are to run on and which nodes are to be the backup nodes.

Determine what the characteristics of the device switchover should be. For 5250 devices (displays and printers), include a hardware switch in the configuration if 5250 types of users are to switch or failover to a backup machine.

The 5250 devices are switched manually or automatically as a result of a switch or failover occurring. An automatic switch requires third-party software. This software can be available as part of the cluster middleware providers product.

For IP devices, such as PCs that browse to the node, a simple refresh of the browser reselects the IP takeover address on the new node. The user can then re-access their application.

11.2.4 Switchover

Once an application becomes ClusterProven, and the node to run the application on is known, consider the switchover characteristics of the business itself.

The tasks involved with a switchover include:

- Move existing users off the application
- Prevent new users access to the application
- Stop the application
- Complete the apply tasks

Some applications may have already implemented these functions, especially in the banking and services sector. These organizations tend to have stringent end-of-day (EOD) routines that require all users to be off the system for a short period of time while the EOD tasks run. These applications have methods for removing users from the application.

11.2.5 Failover

A failover is similar to a switchover, but with less control. In theory, if the failover is completely seamless, it can be used as the switchover. For example, a switchover means press the Off button. However, given the option, it is safer to switch.

The tasks involved in planning for a failover ensure that the users can be returned to the same position they were at before the failure.

11.2.6 Job restart

OS/400 job restart is a complicated task. Careful planning is required to achieve the fastest and safest restart of jobs on the backup node.

The restart characteristics of different types of jobs running on an iSeries server are described in this section.

Interactive jobs

Interactive jobs are run from a twinaxial display or in an emulator session.

In a switchover situation, interactive jobs can normally be ended in a controlled fashion. All transactions typically complete without loss of data.

In a failure condition, interactive jobs end abnormally. Transactions are incomplete and temporary objects are lost. However, with a good understanding of the application, the losses can be contained and the recovery can be planned.

Since these interactive devices use a 5250 data stream, they cannot be controlled by IP takeover. They must be manually switched. This hardware switch process can be linked to the IP takeover.

Batch jobs

With a long running single thread batch job, establish whether this job can have restart points added or whether it must be completely restarted. Restarting the batch job can be necessary. However, it can be a long running operation that can seriously effect the overall availability of the application or business.

Multi-threaded batch jobs are more complex. Restart points may not be available within the job. The job may need to be rolled out and restarted.

Client/server jobs

Client/server is perhaps the easiest iSeries environment to cluster. The state of the client is not important to the iSeries server. It is important to consider the client under a different part of the cluster or high availability project.

Clients are typically IP-connected devices. IP takeover can handle the movement of these devices from primary to the backup. In most cases, the client should see very little impact from a failure or switchover.

11.2.7 Application maintenance

Application maintenance is a potentially difficult area to support high availability in a business. If the application only needs changes to the programs running within it, maintenance is minimized. However, if the application requires changes to the underlying database, maintenance is more complicated.

The design of an application must support these events to afford resilient functionality:

1. Switchover the primary system to the first backup node.
2. Replication is ended between the primary and first backup system.
3. With the normal primary system offline, application maintenance is carried out. A database mapping program is installed.

   These three functional steps are illustrated in Figure 11-1. Figure 11-1 through Figure 11-3 illustrate how to retain full protection for the cluster while performing an application maintenance task.

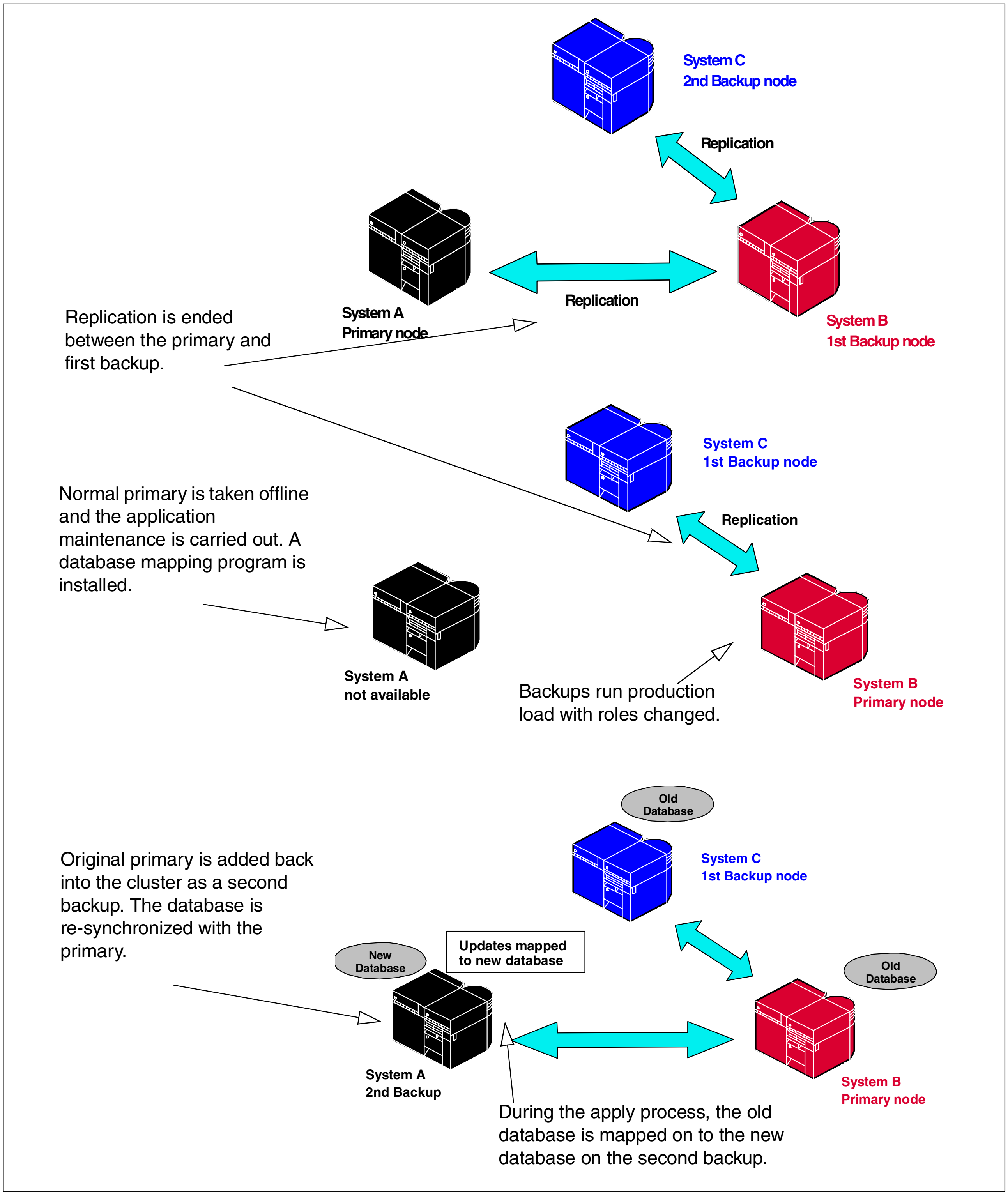
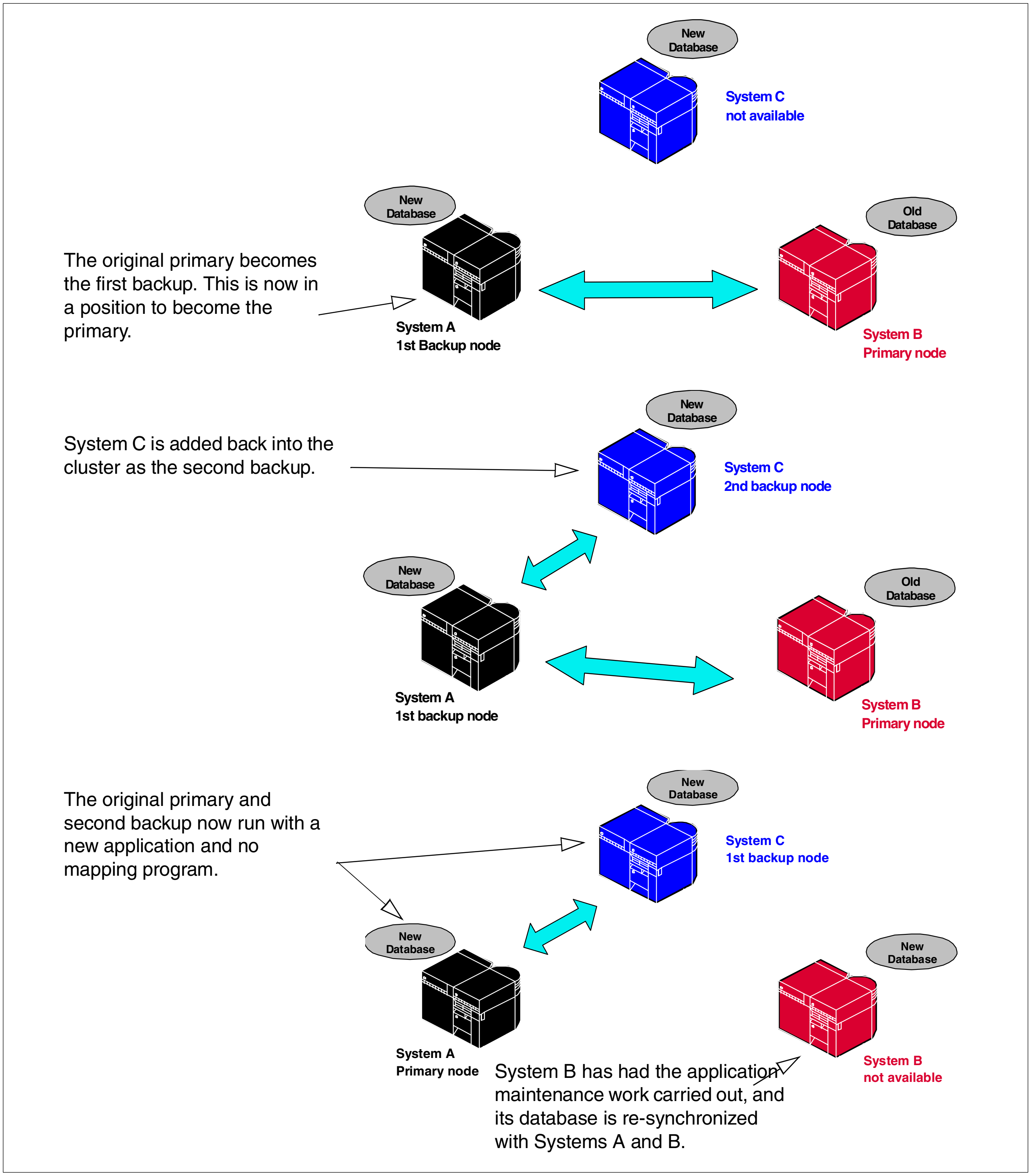

System C
2nd Backup node
Replication
Replication
System A
Primary node
System B
1st Backup node
Replication is ended between the primary and first backup.
System C
1st Backup node
Replication
Normal primary is taken offline and the application maintenance is carried out. A database mapping program is installed.
System A
not available
Backups run production load with roles changed.
System B
Primary node
Old
Database
System C
1st Backup node
Original primary is added back into the cluster as a second backup. The database is re-synchronized with the primary.
New
Database
Updates mapped
to new database
Old
Database
System A
2nd Backup
System B
Primary node
During the apply process, the old database is mapped on to the new database on the second backup.

Figure 11-1 Application maintenance in a cluster (Part 1 of 3)

4. The original primary node is added back into the cluster as a second backup. The database is resynchronized with the primary system.
5. The role is swapped between the first and second backups. System C is taken offline to maintain the application.

6. During the apply process, the old database is mapped to the new database on the second backup system.
7. The original primary becomes the first backup. It is now in a position to become the primary system.

These four steps are illustrated in Figure 11-2.



The original primary becomes the first backup. This is now in a position to become the primary.

System C is added back into the cluster as the second backup.

The original primary and second backup now run with a new application and no mapping program.

System B has had the application maintenance work carried out, and its database is re-synchronized with Systems A and B.

Figure 11-2 Application maintenance in a cluster (Part 2 of 3)

8. System C is added back into the cluster as a second backup. The old database is replicated to the first backup and remapped to the new database. The new database is then replicated from the first backup to the second backup.
9. The original primary and second backup now run with the new application and no mapping program.
10. System B has had the application maintenance work carried out. Its database is re-synchronized with Systems A and B.
11. In the final phase, System B is added back into the cluster as the second backup. Then, its role is switched to the first backup. The cluster is returned to its original state.

    These four step are illustrated in Figure 11-3.

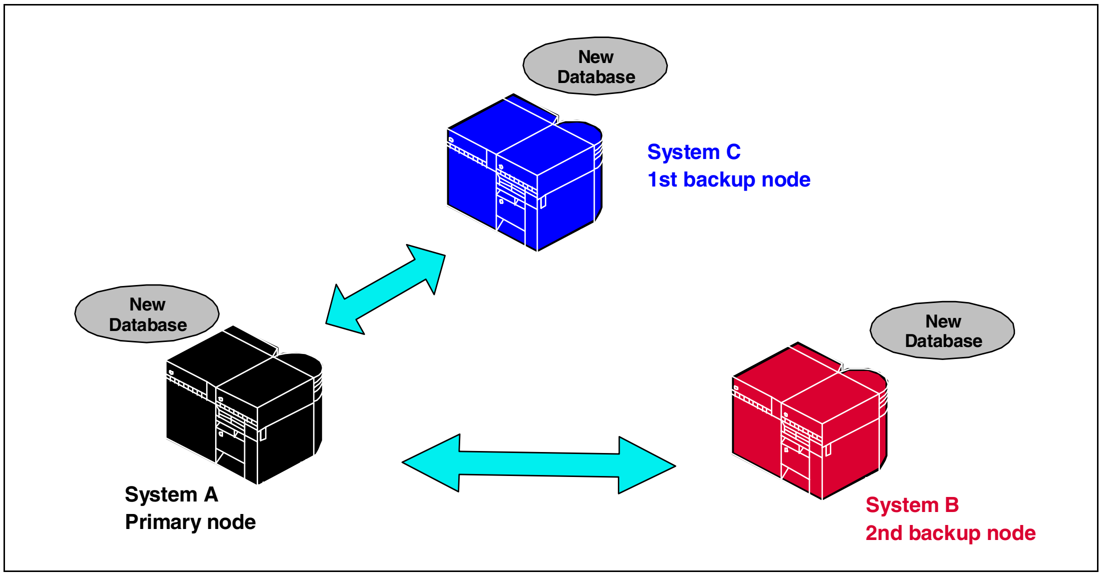


Figure 11-3 Application maintenance in a cluster (Part 3 of 3)

11.2.8 Database performance

If replication software is currently running, you need to understand whether there is a database performance problem. Database performance is not directly related to creating a clustered solution. A resolution to the performance problem on nodes of a cluster is no different than if the problem occurred on a single system.

The use of journaling and commitment control in the application is needed for full deployment of a clustered solution. Journaling does add performance overhead to an application.

Note that technology improvements in OS/400 and the iSeries hardware have minimized performance degradation with journaling and commitment control. These areas continue to be enhanced and improved with each release of OS/400.

However, the significant recovery options facilitated with journaling and commitment control are not without a compromise in performance. And adding journaling and commitment control to a poorly designed database can make things run even slower.

11.3 Systems management

System management is critical component of a clustering implementation plan. Cluster management is eased with the support of stable system management policies. Incorporate a discipline for system management before you implement a clustering project.

Effective systems management involves plans that define and support a business' expectations for service, operations, managing problems and change, planning for capacity, optimizing performance, and protecting the system, as discussed in this section.

11.3.1 Service level agreements

When considering service level agreements, start with the business executives. They are the key sponsors for developing service level agreements that support the needs of the business. The executives are also the source of the requests to the finance department.

Much of the planning for a service strategy is done when developing the financial case for the availability solution.

11.3.2 Operations management

In a single system environment, the operations staff typically has a proven management system. Daily run schedules and overnight batches are well understood. Normally there is a simple but long running recovery process.

In a cluster environment, a more lights-out operation can be run. That is, many tasks can be performed automatically.

Establish a cutover plan from the existing environment to the clustering environment. This can mean changing work-shift patterns and possibly adding staff to cover tasks that could run over weekends.

The skills and resources that are required to implement and manage a cluster solution are different from most current I/T departments. Critical skills are database management and networking. Most customer shops have programming skills and some networking skills. As the mission critical application starts, the ability to remedy the problem very quickly is important to meet service level agreements. Having the right skills is essential.

Once these steps are complete, update or re-write the operational documentation.

11.3.3 Problem and change management

Establishing an effective problem and change management strategy is critical when managing clustered systems. If there are problem and change management processes in place to support a single system, modify them to manage a clustered environment. Account for different service level agreements.

In a clustered setup, it is more important to reporting and analyze problems as quickly as possible. Minimize any time spent waiting for a key person to return when a problem occurs.

To make fast informed decisions, improve the escalation process of reporting problems. For example, when the operator notes errors reported by disk units of an impending failure, do not wait for the return call of a Customer Engineer. Pump the disk. Prepare for a switchover if the system performance degrades.

Analyze each system throughout the business for a risk assessment. Prioritize each system's availability. Document failover plans. Include the decision points in the failover plan, with directions on which step to take next.

Consider this example. A processor sends a warning of an overload. One of the applications running on that processor needs to be switched to another processor that has spare capacity.

Prioritize the applications on the overloaded system. Implement the predefined plan. Initiate the planned switch for the application. Exit programs cause the CRGs, IP address, and users to be switched to the backup system. This application remains on the backup system until the primary processor is analyzed and the problem rectified. Then switch the application back to the primary system.

Many of the normal problems of maintaining a highly available single system are automated or disappear. Now, different problems exist and need to be addressed, such as:

- Partition state
- Application maintenance
- Managing a mix of applications with a different resilience characteristics
- Managing batch applications in a cluster

11.3.4 Capacity planning

Cluster functions cause minimal overhead to system resources. Nevertheless, size the systems to support the cluster setup.

The easiest way to size a cluster is to first plan the capacity as a single system. Then model the load produced by journaling and the apply process. Use BEST/1 to model these functions and to provide a relatively accurate model of the single node.

Planning for capacity can be more complex depending on the roles of the nodes, and the applications run in different settings. For example, one system is planned to be the primary node for an application in one recovery domain. With a failure or switchover, this node supports another application.

Include this additional application workload in the planning process.

Review each node and model for the worst case scenario. Make sure that the capacity of other components related to the node are included, for example, I/O processors, network connections, and backup devices.

Once the cluster is up and running, regularly monitor and re-evaluate the capacity of all the nodes. Do not allow the resources on the backup nodes to become overutilized. If a failure or switchover is necessary, the additional load on the backup machine can create an availability problem of its own.

11.3.5 Performance planning

Performance considerations for clusters are similar to capacity planning for clusters. Take performance measurements regularly to monitor that service levels are achieved. Register any out-of-line situations. Take corrective action to reduce the risk of losing the integrity of the cluster.

When a switchover occurs for the purposes of routine maintenance, take recordings from the backup node to ensure that it still meets the performance expectations and that the backup node is not degraded by the added workload.

11.3.6 Security and password considerations

The Allow Add to Cluster (ALWADDCLU) network attribute specifies whether a node allows itself to be added to a cluster. Use the Change Network Attribute (CHGNETA) command on any system that is to be set up as a cluster node. Set a value for the ALWADDCLU network attribute, before you add a node to a cluster.

Depending on the option chosen for this attribute, the use of X.509 digital certificates can be required. A digital certificate is a form of personal identification that can be verified electronically.

Because there is no central security administration to update nodes, user profiles are not automatically updated across cluster nodes. Be sure to update the security information across all nodes to ensure that any public or private authorities associated with any cluster objects, Cluster Resource Groups, applications, or data have the correct security level. Synchronize the user profiles across systems in a recovery domain so that the user profile name, the UID, and the GID are the same on all systems.

One mechanism to update security information and synchronize user profiles is through the Management Central framework. Perform administrator or operator functions across multiple systems and groups of systems. Another option is to use software functions provided by a cluster middleware provider.

11.4 Hardware considerations

Hardware is relatively inexpensive when compared to the cost of failure. When investigating the cost of hardware, do not simply look at the hardware related to the computer system. There are many other hardware components of a complete continuously available solution.

It is relatively easy to provide a configuration that includes redundant hardware in a continuously available system complex. However, redundant hardware adds complexity to overall system management.

When planning the total computing solution, consider a redundant configuration for these components:

- Processor complex (includes bus)
  - Disk redundancy
  - Adapter redundancy
  - Remote site redundancy

  This redundancy can consist of a complete remote location or a remote controller at the main site.
- Site
  - Machine room
  - Air conditioning
  - Power supply
  - Office space
  - Telephone services
- Network hardware

  From routers to remote controllers, review all network hardware. If there are critical requirements for remote access, provide alternative network paths. This can be as straightforward as a dial-up link, or as complex as a multi-path private network.

- Network connection

  If planning to extend business services to an intranet or the Internet, consider the impact of a single Internet Service Provider (ISP). The ISP is another single point of failure. For the highest availability, use multiple network providers.

11.4.1 Network planning

Network planning includes capacity and accessibility factors. The network must be able to maintain the same level of availability as the nodes of the cluster.

Communication providers must provide guarantees that are available and have sufficient capacity for all possible switch scenarios. There must be alternative network paths to enable the cluster services to manage the cluster resources. These redundant paths should prevent a cluster partition occurring.

Figure 11-4 illustrates redundant communications paths between the nodes in the cluster.

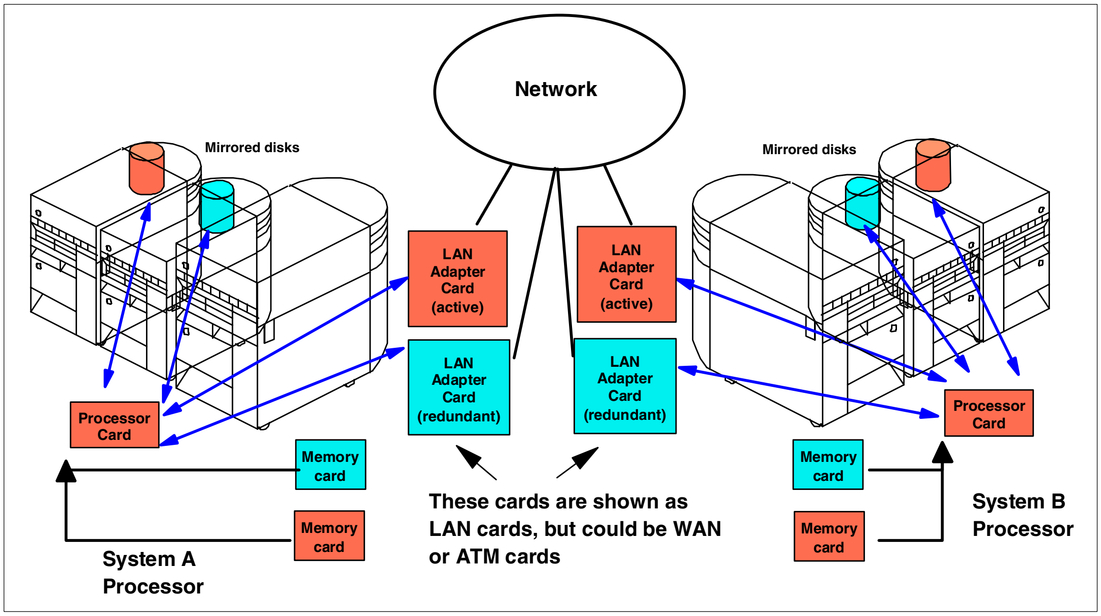

Network
Mirrored disks
Mirrored disks
LAN Adapter Card (active)
LAN Adapter Card (active)
LAN Adapter Card (redundant)
LAN Adapter Card (redundant)
Processor Card
Processor Card
Memory card
Memory card
Memory card
Memory card
These cards are shown as LAN cards, but could be WAN or ATM cards
System A Processor
System B Processor

Figure 11-4 Redundant network connections

A *redundant communications path* is when there are two communication lines configured between two nodes in a cluster. If a failure occurs, the second communication path can take over to keep communications running between the nodes. This minimizes the conditions that can put one or more nodes of the cluster into a cluster partition situation.

Note that if both communication lines are defined to go into the same adapter on the system, both lines are at risk if the single adapter fails. Redundant adapters are recommended.

See 12.3, “Recovering from a cluster partition state” on page 220, for more information on cluster partitions.

11.5 Testing clusters

Testing is a well-established and controlled part of most development shops.

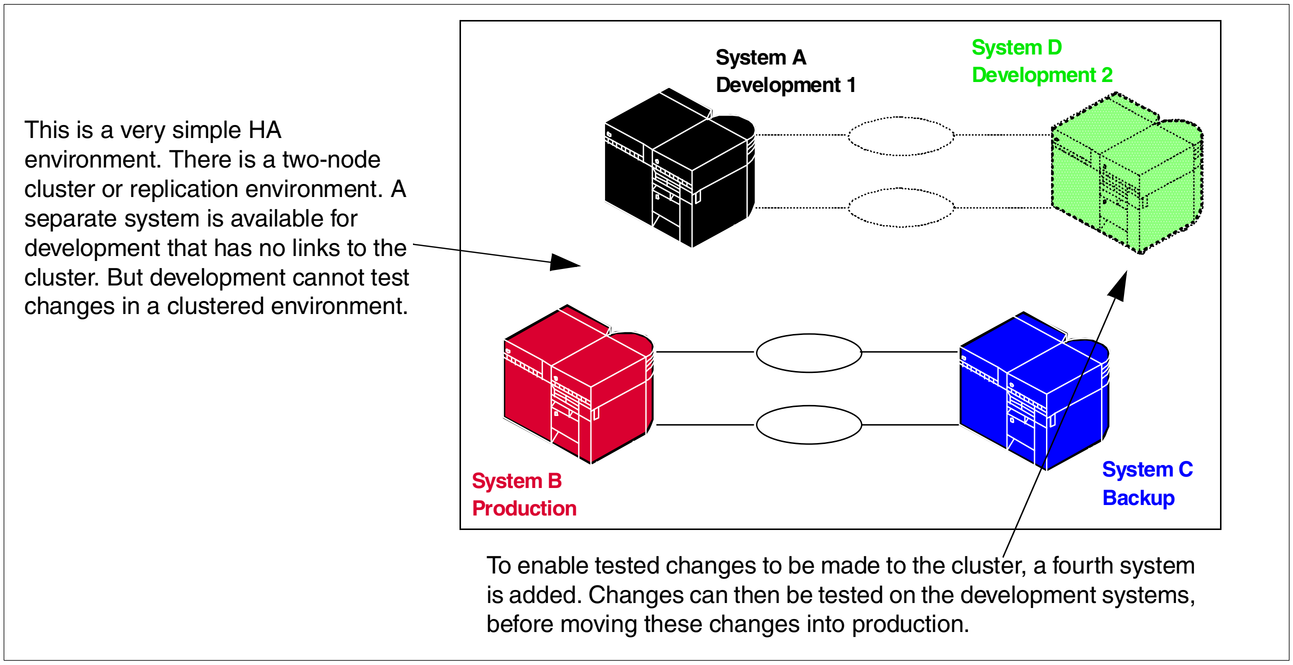
A second system or separate LPAR partition enables testing of the operating system and application test environments. Unfortunately, this testing is not always extended to all facets of the business. Network, hardware, and external link testing are often overlooked.

A classic example of the need for testing is Web page design. A Web page is built and tested on the internal LAN and then placed onto the Web. The site is tested from the LAN out to the Web and back into the Web site. High speed links are typically used. However, remote testing is not done. When the remote site connects with, for example, a 14.4 Kbps modem on the other side of the world, and the application is made available, it takes an extended time to load. Visitors do not stay on a site with slow response time.

Determine whether the development environment is a critical resource. If it is, include the development system in the cluster. A totally separate cluster test configuration is needed.

Figure 11-5 illustrates how a simple customer setup can be changed to produce a more effective test environment. The test scenario illustrated is a simple high availability environment. It is a two-node cluster or replication environment. A separate system is available for development which has no links to the cluster.

Development cannot test changes in a clustered environment. To enable tested changes to be made to the cluster, a fourth system is added. Changes can then be tested on the development systems before moving these changes into production.



This is a very simple HA environment. There is a two-node cluster or replication environment. A separate system is available for development that has no links to the cluster. But development cannot test changes in a clustered environment.

To enable tested changes to be made to the cluster, a fourth system is added. Changes can then be tested on the development systems, before moving these changes into production.

Figure 11-5 Cluster test scenario

Figure 11-5 shows the basic systems. It does not show the routes, LANs, WANs, etc. that need to simulate the real environment.

Creating a separate cluster with two small systems meets most testing needs. An issue with this arrangement is the possibility that certain types of hardware and peripherals may not work with smaller systems, and it may be difficult to do any accurate volume testing.

Whether a highly available solution is implemented, or building a cluster is the plan, consider a test environment during the implementation. Testing is as critical as any application on the system. The cost of a test system is relatively trivial. Do not try to save money in this area. Establish an ongoing problem and change management test environment.

11.5.1 System management tests

System management testing is mainly aimed at performance and operations. These areas are tested:

- Application process (normally part of development testing)
- Application volume
- Hardware and peripheral (tape, DASD, IOPs, remote devices, clients)
- Interoperability
- Network performance
- Network hardware

When performing volume-related testing, it is important to have a well-documented script for producing the test. If the capacity is not available on the local test machines, consider an external testing source, for example, one of the IBM Benchmark Centers. Visit this Web site to see more IBM testing sites: `http://www.developer.ibm.com`

11.5.2 Cluster management tests

Some of the scenarios to test before moving the cluster into production are:

- Planned switch
- Failover
- Rejoin
- Adding a cluster node

Re-test these scenarios after a system upgrade or major change to any of the cluster components.

11.6 Roles and responsibilities when implementing clustering

IBM supports the cluster middleware business partner solutions for clustering on the iSeries server. Implementation of a clustering solution for an iSeries customer is a cooperative effort between IBM, the cluster middleware provider selected by the customer, and the application business partner responsible for the application running at the customer site.

This section describes the roles fulfilled by IBM, the cluster middleware provider, and the application provider when implementing a full cluster solution.

11.6.1 The role of IBM and OS/400

IBM's role is to provide the foundation for clustering services. To fulfill that role, OS/400 delivers integrated system services and a set of APIs to create, change, delete, and manage clusters, nodes and CRGs.

One critical clustering system service is activation of the exit program whenever an event occurs for a recovery domain. Calling the exit program is much like calling a trigger program when a database event occurs. The programmer determines what happens when the program is called, but OS/400 initiates the program automatically on all the nodes in the affected recovery domain. When any change to the cluster environment occurs, OS/400 calls the exit program with information such as the current role of the node and the action code – addition of a node, for example. Most of the clustering APIs allow developers to specify up to 256 bytes of information to be passed to the exit program.

Other clustering system services, while less visible, are still important because some functions are implemented more efficiently at the operating-system level than they can be with any third-party product. For example, an IP address takeover (or IP takeover) makes it possible for multiple nodes in the recovery domain to have the same IP address at different times. (Two nodes can never have the same IP address at the same time.) IP takeover facilitates a transparent switchover in a TCP/IP environment.

Perhaps the most critical clustering function, a function that forms the basis of all clustering support from OS/400 V4R4 onward, is *heartbeat monitoring*. Heartbeat monitoring constantly checks communications between the nodes in a cluster. If communication between two nodes is lost, heartbeat monitoring attempts to reestablish communications. If a node fails, heartbeat monitoring reports the node failure to the rest of the cluster.

Although the high availability business partners (HABPs) can implement a form of heartbeat monitoring, IBM's system-level implementation of heartbeat monitoring consumes fewer resources and provides a more accurate view of a node’s status. Heartbeat monitoring, in conjunction with other integrated cluster services, ensures that all nodes have a consistent view of the cluster.

To assist the users who prefer to work with clusters themselves, resources are available in the QUSRTOOL library that allow the user to build a set of Cluster Resource Service commands. These commands provide a user friendly interface to the Cluster Resource Service APIs.

11.6.2 The role of a cluster middleware provider

Although it is possible for a user to build a clustered environment, this work is often performed in cooperation with, or by, a cluster middleware business partner such as those described in Part 3, “Cluster middleware business partners” on page 227.

OS/400 APIs are available to define resilient resources and initiate planned switchovers. Cluster middleware providers use these APIs to deliver interfaces and tools to manage clusters. These tools complement their existing high-availability offerings and insulate application developers from coding directly to the clustering APIs.

For managing cluster objects, IBM has defined a standard for cluster management utilities. Some cluster middleware providers provide a graphical tool with its own unique personality that satisfies this standard and integrates OS/400 clustering functions with functions in their high-availability product.

11.6.3 The role of an application developer

Application developers handle all application-specific clustering tasks. Some of these tasks work in conjunction with the high-availability solutions. For example, an application developer defines resources such as files and program objects for replication in the automated installation data area. The cluster middleware provider usually handles the task of replicating information.

Application developers also provide an exit program to restart applications on a backup system after a switchover. For example, the exit program can be used to ensure that the required CRGs are available before an application is restarted.

Ideally, an application is transferred from one iSeries server to another and the user is repositioned at precisely the same spot in a transaction as before the switchover occurs. In pursuit of this ideal, application developers can add checkpoint facilities to their applications. An application checkpoint is usually implemented using commitment control. In conjunction with an appropriate exit program, a checkpoint application has the ability to restart at the last complete transaction.

12

Problem determination for iSeries clustering

To effectively manage an iSeries cluster, the operator must first know where to look for error messages that deal specifically with clustering and understand how to resolve the errors. This chapter identifies problem determination tools and techniques that are useful when encountering a problem during the setup of clusters or when managing an iSeries cluster.

12.1 Monitoring for problems

As with any component of the iSeries and OS/400, error logs exist to help in problem determination and resolution. There are several system log entries and error messages unique to clustering:

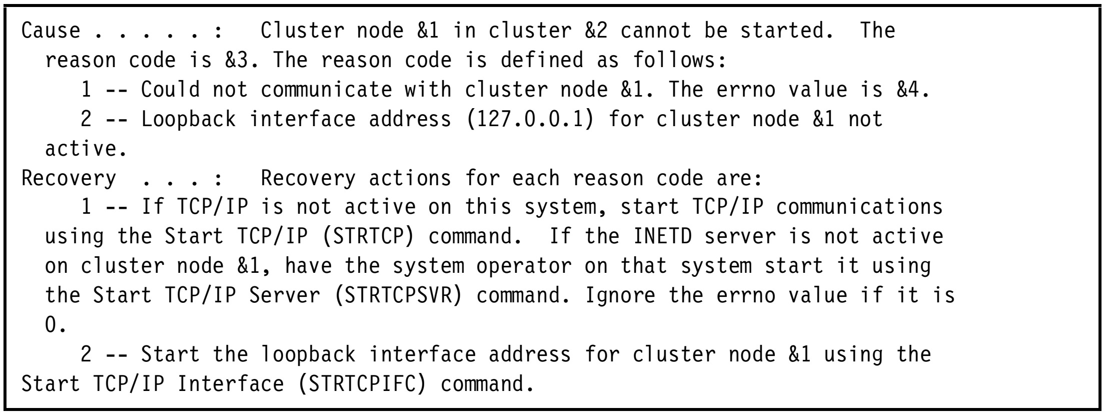
- Vertical Licensed Internal Code (VLIC) logs (VLOG) entries with a major code of 4800 indicate a clustering situation.
- Many of the messages related to clustering are in the range of CPFBB00 to CPFBBFF.

The occurrence of CPF messages can be monitored for. Alertable messages can be found in the QHST history log and in the QSYSOPR message queue. Use the Display Log (DSPLOG) command to display the history log, and use the Display Message (DSPMSG) command to see what is logged in QSYSOPR.

If following the suggested resolution in the message text does not resolve the problem, note the message ID and message text before calling IBM for service. For example, message ID CPFBB05 contains the help text:

```
Cluster node xx cannot be started.
```

The recovery action based on the cause code is also found within the message text. See Figure 12-1 for an illustration.



```
Cause . . . . . :   Cluster node &1 in cluster &2 cannot be started.  The
  reason code is &3. The reason code is defined as follows:
     1 -- Could not communicate with cluster node &1. The errno value is &4.
     2 -- Loopback interface address (127.0.0.1) for cluster node &1 not
  active.
Recovery  . . . :   Recovery actions for each reason code are:
     1 -- If TCP/IP is not active on this system, start TCP/IP communications
  using the Start TCP/IP (STRTCP) command.  If the INETD server is not active
  on cluster node &1, have the system operator on that system start it using
  the Start TCP/IP Server (STRTCPSVR) command. Ignore the errno value if it is
  0.
     2 -- Start the loopback interface address for cluster node &1 using the
Start TCP/IP Interface (STRTCPIFC) command.
```

Figure 12-1 Recovery action for clustering error message

Messages related to cluster configuration are logged to the QCSTCTL job log. Messages related to configuring a Cluster Resource Group (CRG) are logged to the QCSTCRGM job log. Messages related to managing a Cluster Resource Group are logged to the CRGs job log. All of these job logs are found in the QSYSWRK subsystem. Run the Work with Active Job (WRKACTJOB) command to display messages about these Cluster Resource Service jobs.

Use the WRKACTJOB command to display messages about the Cluster Resource Group jobs. Look for the name of the Cluster Resource Group job under the QSYSWRK subsystem. For example, if the name of the Cluster Resource Group is CRG1, look for the job named CRG1 in QSYSWRK.

To find an application Cluster Resource Group job, follow these steps:

1. Determine the subsystem where the application job is running. Check the Cluster Resource Group object to determine the user profile used when the application Cluster Resource Group job was submitted. Every user profile is associated with a job description and every job description is associated with a subsystem.

2. Check this subsystem for the application Cluster Resource Group job.

12.2 Problems with the status of a cluster

This section addresses commonly asked questions about the status of clustering jobs when working on a system with clusters. You can find more complete information in the Troubleshooting clusters topic in the Information Center at:
http://www.ibm.com/eserver/iseries/infocenter

To locate the problem determination section, click **Systems Management-> Clusters-> Troubleshooting clusters**.

12.2.1 Is my cluster up and running?

To determine if Cluster Resource Services (CRS) is active on the system, run the WRKACTJOB command from an OS/400 command line. Under the QSYSWRK subsystem, look for two jobs:

- QCSTCTL
- QCSTCRGM

If these jobs exist, Cluster Resource Services is active and the cluster is up and running.

QCSTCTL and QCSTCRGM jobs are cluster-critical jobs. Both jobs must be active for Cluster Resource Services to be active. Should one of these jobs end, clustering ends on that system (node).

CRG jobs are not cluster-critical jobs. Clustering can remain active when a CRG job ends.

12.2.2 Why won't my cluster start?

When a cluster is not started, check these areas:

- Make sure that TCP/IP is active:

  a. Enter the WRKACTJOB command.

  b. Locate the QSYSWRK subsystem.

  c. Look for a job named QTCPIP running in the QSYSWRK subsystem. If this job exists, TCP/IP is running on the system.

  Or follow these steps:

  a. Enter the network status command (NETSTAT).

  b. Select option **1** to see if TCP/IP is active. If TCP/IP is not active, run the Start TCP/IP (STRTCP) command from the OS/400 command line.

- Be sure that the *INETD server is started by performing one of the following tasks:

  – Under the QSYSWRK subsystem, look for a job named QTOGINTD. If this job exists, the *INETD server is started.

  – Use NETSTAT and select option **3** to see if the *INETD server is started.

    If *INETD is not started, run the Start TCP Server (STRTCPSVR *INETD) command from the OS/400 command line.

- Check the ALWADDCLU network attribute

  The ALWADDCLU network attribute specifies whether a node allows itself to be added to a cluster. This should be set to either *ANY or *RQSAUT depending on the environment. The default value is *NONE, which does not allow the system to be added to a cluster.

- Check that the IP addresses chosen to be used for clustering locally and on the target node show an *Active* status.

  The local and any remote nodes must be able to PING using the IP addresses to be used for clustering to ensure network routing is active.

- Check that the LOOPBACK address (127.0.0.1) locally and on the target node is also active.

12.2.3 Why is my CRG hung up?

The Cluster Resource Group job can appear to be in a hang (wait) state for a number of reasons. For example, a hang state can appear if:

- The exit program associated with the CRG is itself in a long wait state (status).

  The CRG hangs upon a return of control from the exit program.

- The CRG exit program job has not ended or returned to the CRG job.
- There is an inquiry message on the QSYSOPR message queue while the CRG job logs First Failure Data Capture (FFDC) messages, and the spooled file reaches the maximum number of records allowed.
- An exit program is submitted to a job queue in a subsystem that releases one job at a time.

  In this case, the CRG waits for the exit program to return a completion message. If the exit program does not complete, the CRG appears to be in a hang state.

Maximum number of jobs from job queue or in subsystem

The default job description submits jobs to the QBATCH job queue. This QBATCH is used for many user jobs.

Set the maximum jobs to release from QBATCH to a value that allows the exit program job to run in a timely fashion.

> **Tip:** Specify a unique job queue name in the job description that identifies the exit program associated with the CRG.

In addition to the control offered by the number of jobs allowed to run from the job queue at any time, the subsystem definition has a similar throttle. Be sure that the subsystem the Cluster Resource Group exit program runs in allows more than one job to run at a time.

Use the Change Subsystem Description (CHGSBSD) command and specify `*NOMAX`, or a number greater than 1, for the maximum jobs parameter. If it is not possible to change the maximum job value to *NOMAX, consider creating a separate subsystem for the CRG exit program job or other jobs.

12.2.4 I cannot use the IBM Cluster Management GUI

To access the IBM Simple Cluster Management GUI, the High Availability Switchable Resources option of OS/400 must be installed on the system. A valid license key for this option must exist.

Use the GO LICPGM CL command and select option **11** (Display Licensed Programs). Look for option 41 HA Switchable Resources and install it if necessary.

> **Note:** If the cluster entry does not appear in Operations Navigator when you expand Management Central, that is likely because “Logical Systems” was not selected when installing Operations Navigator. Logical Systems is not included in the default setup of Operations Navigator. Re-install Operations Navigator on the workstation. Be sure to select *Logical Systems* on the Component Selection window. This will solve the problem and clusters will appear under Management Central.

12.2.5 I cannot use any new release functions

An attempt to use a new release function that is not compatible with the current cluster version errors with a CPFBB70 error message. The text of CPFBB70 indicates that the API request is not compatible with the current cluster version. This happens when the current cluster version does not support the function requested by the API.

Note that the new release functions are not available until both of the following points are true:

- All cluster nodes have a potential cluster version that supports the new function.
- The current cluster version has been set to match this potential cluster version.

For example, consider the use of the Change Cluster Resource Services API (QcstChgClusterResourceServices). In a cluster setup consisting of two-nodes, Node A is at OS/400 V4R5 and Node B is at V5R1. An attempt is made on Node B to change the cluster performance tuning level to be more sensitive to communications failures. This function is available with the V5R1 QcstChgClusterResourceServices API.

Another example is the Add Device Domain Entry API (QcstAddDeviceDomainEntry).

Even though these APIs are supported on one of the nodes of the cluster, Cluster Resource Services (CRS) does not allow the operation on any node of the cluster, because the cluster, as a whole, does not support the V5R1 function. To resolve this conflict, upgrade all cluster nodes to the new release level. Then use the QcstAdjustClusterVersion API to adjust the current cluster version and try the request again.

You can learn more about cluster versioning in “Cluster versioning” on page 36.

12.2.6 How do I recover from a Cluster Resource Group job failure?

Failure of a Cluster Resource Group job is usually indicative of some other problem. Look in the job log associated with the failed job for messages that describe why it failed. Correct any error situations.

To recover from a failure of a Cluster Resource Group job, follow these steps:

1. End clustering on the node where the job failure occurred.
2. Restart clustering on the node.

12.2.7 Why do I have two clusters after fixing my cluster partition?

The most common reason for more than one cluster to exist in a partition is when the Start Cluster Node (QcstStartClusterNode) API runs in an inactive node. Run this API on an active node in the cluster to start Cluster Resources Services on the inactive node.

12.3 Recovering from a cluster partition state

A *cluster partition* happens when contact is lost between one or more nodes in the cluster and a failure of the lost nodes cannot be confirmed. A CPFBB20 error message is issued to indicate the cluster is in a partitioned cluster situation.

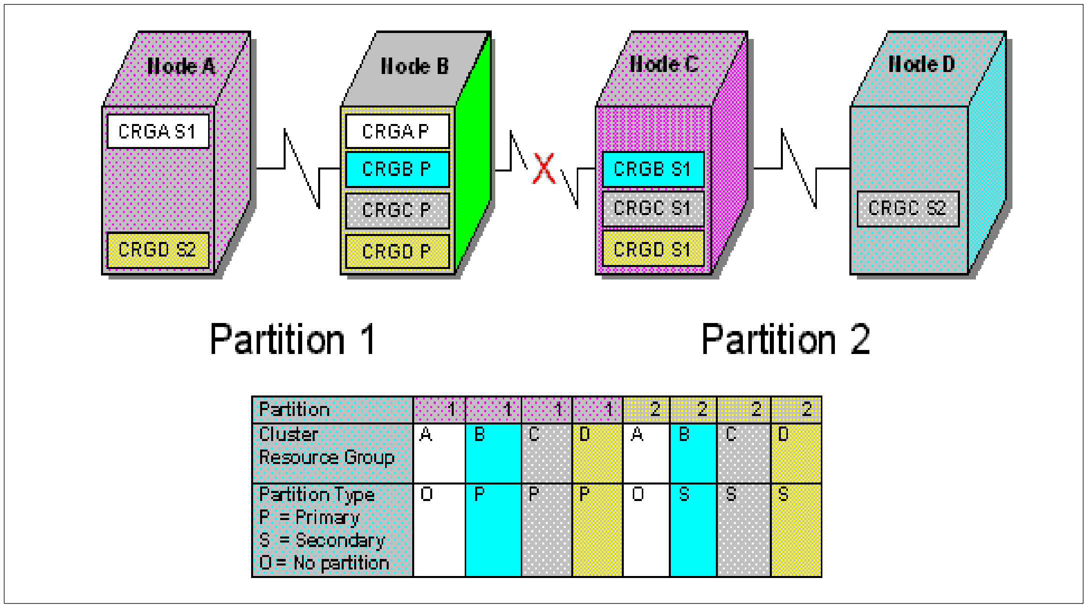
To recover, find the CPFBB20 error message in the QHST history log and in the QCSTCTL job log in the QSYSWRK subsystem. Follow the recovery action found in the message text.

12.3.1 Cluster partition error example

This section illustrates a cluster partition condition that involves a cluster made up of four nodes, known as A, B, C, and D. The Cluster Resource Groups associated with these nodes are named CRGA, CRGB, CRGC, and CRGD respectively.

For this example, a loss of communication occurs between cluster nodes B and C. The cluster then divides into two cluster partitions.

Figure 12-2 illustrates the recovery domain of each Cluster Resource Group.



Partition	1	1	1	1	2	2	2	2
Cluster Resource Group	A	B	C	D	A	B	C	D
Partition Type P = Primary S = Secondary O = No partition	O	P	P	P	O	S	S	S

Figure 12-2 Cluster partition

The types of Cluster Resource Group actions that can be taken within a cluster partition depend on whether the partition is a primary or a secondary cluster partition.

The cluster partition that contains the current primary node in the recovery domain of a Cluster Resource Group is considered the primary partition of the Cluster Resource Group. All other partitions are secondary partitions. The primary partitions may not be the same for all Cluster Resource Groups.

Table 12-1 and Table 12-2 identify the partition type in which each Cluster Control and Cluster Resource Group API is allowed to run. The action performed by the API takes affect only in the partition in which the API runs.

Table 12-1 Cluster Control API partition restrictions

Cluster Resource Group API	Partition the API is allowed to run in
Add Cluster Node Entry	Not allowed in any partition
Add Device Domain Entry	*
Adjust Cluster Version	Not allowed in any partition
Change Cluster Node Entry	Any partition
Change Cluster Resource Services	Any partition
Create Cluster	Not allowed in any partition
Delete Cluster	Any partition
End Cluster Node	Any partition**
List Cluster Information	Any partition
List Device Domain Information	Any partition
Remove Cluster Node Entry	Any partition
Remove Device Domain Entry	Any partition***
Retrieve Cluster Information	Any partition
Retrieve Cluster Resource Services Information	Any partition
Start Cluster Node	Any partition
* Allowed only for an existing device domain where all members are in the same partition. ** Allowed only in the same partition as the node being ended. *** All members must be in the same partition.	

Table 12-2 Cluster Resource Group API partition restrictions

Cluster Resource Group API	Partition the API is allowed to run in
Add CRG Device Entry	Primary*
Add Node to Recovery Domain	Primary
Change CRG	Primary
Change CRG Device Entry	Primary
Create CRG	Not allowed in any partition
Delete CRG	Any partition**
Distribute Information	Any partition**
End CRG	Primary
Initiate Switch Over	Primary
List CRGs	Any
List CRG Information	Any
Remove CRG Device Entry	Primary
Remove Node from Recovery Domain	Primary

Cluster Resource Group API	Partition the API is allowed to run in
Add CRG Device Entry	Primary*
Add Node to Recovery Domain	Primary
Change CRG	Primary
Change CRG Device Entry	Primary
Create CRG	Not allowed in any partition
Delete CRG	Any partition**
Start CRG	Primary
* All nodes in the CRGs recovery domain must be active in the primary partition. ** Affects only the partition running the API.	

By following these restrictions, Cluster Resource Groups can be resynchronized when the cluster is no longer partitioned. As nodes rejoin the cluster from a partitioned status, the version of the Cluster Resource Group in the primary partition is copied to nodes from a secondary partition.

When a partition is detected, neither the Add Cluster Node Entry or the Create Cluster API can be run in any of the partitions. All of the other Cluster Control APIs may be run in any partition. However, the action performed by the API takes affect only in the partition running the API.

Once the partitioned cluster situation is corrected, a CPFBB21 message is issued to indicate the cluster partition is recovered. The CPFBB21 message can be found in the QHST history log and in the QCSTCTL job log of the QCSTCTL job running in the QSYSWRK subsystem.

See B.3, “Cluster APIs and related QUSRTOOL commands” on page 286, for a listing and description of each cluster API.

12.3.2 Cluster partition tips

The rules for restricting operations within a partition are designed to make merging the partitions feasible. Without these restrictions, reconstructing the cluster can require extensive work.

This section offers tips for managing cluster partitions:

- If the nodes in the primary partition are destroyed, special processing can be necessary in a secondary partition.

  The most common scenario that causes this condition is the loss of the site that makes up the primary partition. Refer to Figure 12-2 on page 220 as an illustration.

  Assume that Partition 1 is destroyed. To locate the primary node for Cluster Resource Groups B, C, and D Partition 2, perform these operations:

  a. Delete Cluster Resource Groups B, C, and D in Partition 2.

  b. Remove Nodes A and B from the cluster in Partition 2. Partition 2 is now the primary cluster.

  c. Create Cluster Resource Groups B, C, and D in Partition 2. Specify Nodes C and D as the recovery domain.

  d. Establish any replication environments that are needed in the new cluster.

Since nodes are removed from the cluster definition in Partition 2, an attempt to merge Partition 1 and Partition 2 fails. To correct the mismatch in cluster definitions, run the Delete Cluster API on each node in Partition 1. Then add the nodes from Partition 1 to the cluster, and re-establish all the Cluster Resource Group definitions, recovery domains, and replication activity.

This process is difficult and prone to errors. Perform this procedure only in a site loss situation.

- A start node operation is processed differently depending on the status of the node being started:
  - If the node fails or an End Node operation ends the node:
    - Cluster Resource Services is started on the node that is being started.
    - A cluster definition is copied from an active node in the cluster to the node that is being started.
    - Any Cluster Resource Group that has the node being started in the recovery domain is copied from an active node in the cluster to the node being started. No Cluster Resource Groups are copied from the node that is being started to an active node in the cluster.
  - If the node is in a partitioned state:
    - The cluster definition of an active node is compared to the cluster definition of the node that is being started. If the definitions are the same, the start continues as a merge operation. If the definitions do not match, the merge stops. The user needs to intervene.
    - If the merge continues, the node that is being started is set to an active status.
    - Any Cluster Resource Group that has the node being started in the recovery domain is copied from the primary partition of the Cluster Resource Group to the secondary partition of the Cluster Resource Group. Cluster Resource Groups can be copied from the node that is being started to nodes that are already active in the cluster.

12.3.3 Merging a cluster partition

A merge operation is similar to a rejoin operation except that a merge occurs when a cluster has become partitioned. The partition can be a true partition in that Cluster Resource Services is still active on all nodes. However, some nodes cannot communicate with other nodes due to a communication line failure. Or, the problem may be that a node actually fails, but is not detected as such.

In the first case, the partitions are automatically merged back together once the communication problem is resolved. This happens when both partitions periodically try to communicate with the partitioned nodes and eventually re-establish contact with each other.

In the second case, Cluster Resource Services must be restarted on the failed node. Call the Start Cluster Node API from one of the nodes that is active in the cluster to start the CRS. If the Start Cluster Node API is called on the failed node, it becomes a one-node cluster and does not merge back into the rest of the cluster.

As shown in Figure 12-3, a merge operation can occur with one of the configurations that is present.

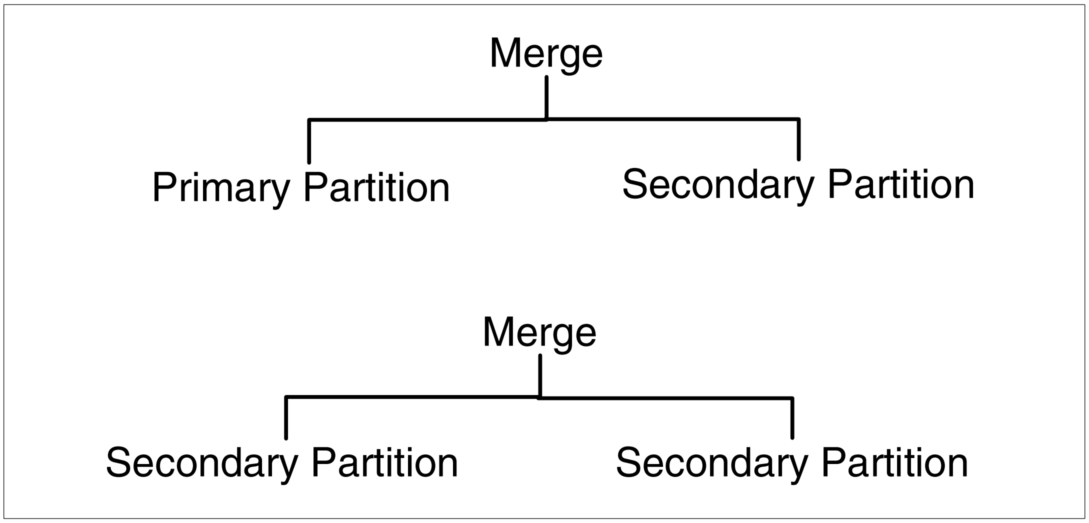


Figure 12-3 Possible merge operations

Primary and secondary partitions are unique to Cluster Resource Groups. For a CRG, a primary partition is defined as a partition that has the CRG's primary node active in it. A secondary partition is defined as a partition that does not have the primary node active in it.

For example, a cluster has two nodes: Node A and Node B. The cluster has two CRGs: CRG 1 and CRG 2. Node A is the primary node for CRG 1, and Node B is the backup node. Node B is the primary node for CRG 2, and Node A is the backup node. If a partition occurs, Node A is the primary partition for CRG 1 and the secondary partition for CRG 2. Node B is the primary partition for CRG 2 and the secondary partition for CRG 1.

This setup is illustrated in Figure 12-4.

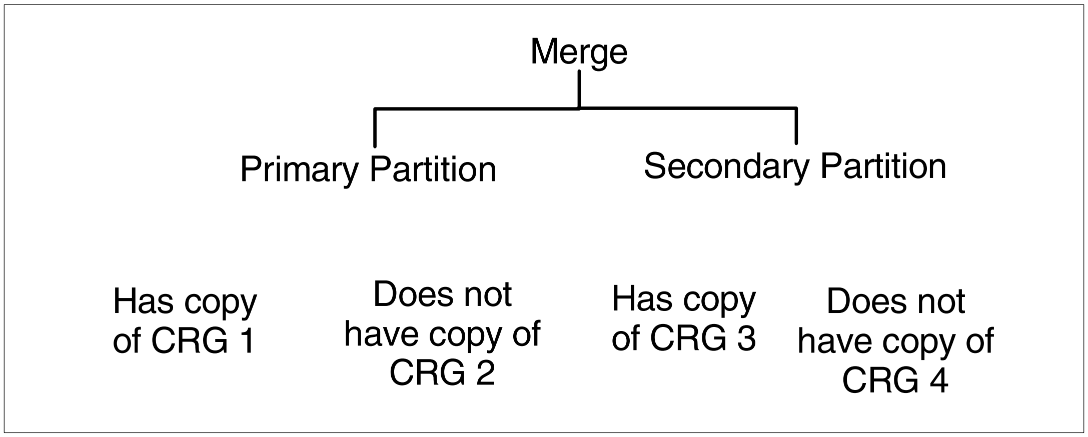


Figure 12-4 Primary-secondary merge operation

During a primary and secondary merge, as illustrated in Figure 12-4, these merges are possible:

- CRG 1 with CRG 3
- CRG 1 with CRG 4

A merge of CRG 2 and CRG 3 cannot happen since a primary partition has the primary node active and must have a copy of the CRG. Likewise, a merge of CRG 2 and CRG 4 cannot happen since a primary partition has the primary node active and must have a copy of the CRG.

Primary and secondary merge

In a primary and secondary merge situation, a copy of the CRG object is sent to all nodes in the secondary partition. The results vary.

As seen on the nodes of the secondary partition, these results are possible:

- No action since the secondary node is not in the CRG's recovery domain.
- A secondary node's copy of the CRG is updated with the data from the primary partition.
- The CRG object is deleted from a secondary node since the secondary node is no longer in the CRG's recovery domain.
- The CRG object is created on the secondary node since the object does not exist. However, the node is in the recovery domain of the CRG copy that is sent from the primary partition.

During a secondary-secondary merge as shown in Figure 12-5, these merge situations are possible:

- CRG 1 with CRG 3
- CRG 1 with CRG 4
- CRG 2 with CRG 3
- CRG 2 with CRG 4

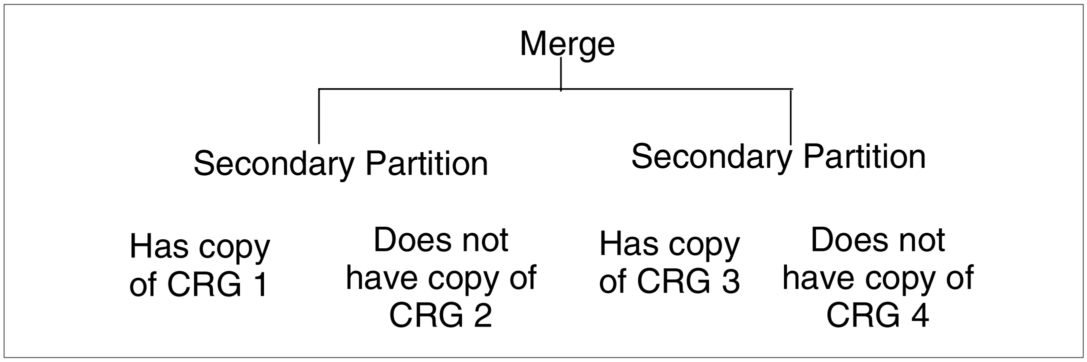


Figure 12-5 Secondary-secondary merge operation

Secondary and secondary merge: Situation 1

In a merge of two secondary partitions, one possible situation is that the node with the most recent change to the CRG is selected to send a copy of the CRG object to all nodes in the other partition. If multiple nodes are selected because they all appear to have the most recent change, the recovery domain order is used to select the node.

The resulting actions that can occur on the receiving partition nodes are:

- No action since the node is not the CRG's recovery domain.
- The CRG is created on the node since the node is in the recovery domain of the copy of the CRG object it receives.
- The CRG is deleted from the node since the node is not in the recovery domain of the copy of the CRG object is receives.

Secondary-secondary merge: Situation 2

In a merge of two secondary partitions, one possible situation is that a node from the partition that has a copy of the CRG object is selected to send the object data to all nodes in the other partition. The CRG object can be created on nodes in the receiving partition if the node is in the CRG's recovery domain.

Secondary-secondary merge: Situation 3

In a merge of two secondary partitions, one possible situation is that internal data is exchanged to ensure consistency throughout the cluster. A primary partition can subsequently be partitioned into a primary and secondary partition.

If the primary node fails, CRS detects it as a node failure. The primary partition becomes a secondary partition. The same result occurs if the primary node that uses the End Cluster Node API is ended. A secondary partition can become a primary partition if the primary node becomes active in the partition through either a rejoin or merge operation.

For a merge operation, the exit program is called on all nodes in the CRG's recovery domain, regardless of the partition in which the node is located. The same action code as rejoin is used.

No roles are changed as a result of the merge, but the status of the nodes in the CRG's recovery domain is changed from *partition* to *active*. Once all partitions merge together, the partition condition is cleared, and all CRG APIs can be used.

Part 3

Cluster middleware business partners

Cluster middleware business partners provide high availability solutions using the Cluster Resource Services support enabled in OS/400. Their customized solutions provide full function support for:

- Cluster management
- Data resiliency

Application resiliency can be delivered by exploiting the OS/400 cluster technology and the cluster management services. Part 3 highlights the cluster management utilities provided for iSeries customers by three cluster middleware business partners:

- DataMirror
- Lakeview Technology
- Vision Solutions

13

DataMirror iCluster

DataMirror’s iCluster is an easy to implement and high performance solution for ensuring the continuous availability of business-critical applications, such as e-business, ERP and customer-facing applications. Built on IBM’s cluster technology as implemented within OS/400 at V4R4 and later, iCluster provides customers with continuous availability during both planned and unplanned outages.

iCluster is for customers who require even greater levels of availability than can normally be attained through a non-clustered iSeries (AS/400) high availability environment.

With the involvement of iSeries Solution Developers (SDs), IBM’s ClusterProven program, and DataMirror iCluster software, iSeries shops now have the whole solution to continuous availability. DataMirror iCluster software further enhances the iSeries cluster solution by providing the easy-to-use cluster management and robust data resiliency required in today's 24 x 365 e-business world.

13.1 Introduction to iCluster

iCluster is built on the underlying engine of DataMirror’s award winning HA Suite product. All of the features and ease of use that customers have come to rely upon are included in iCluster. It includes such features as:

- XtremeCache technology that optimizes the flow of data into and out of a high speed software cache to help customers attain near zero latency for clustered iSeries high availability environments.
- A single product code base for ease of use, administration, and less complex switching.
- Match Merge technology to ensure data and object integrity.
- Real-time auto registration to minimize administration.
- Real-time Integrated File System replication for critical applications such as Domino, WebSphere and ERPs like J.D. Edwards OneWorld and SAP.

iCluster provides three interfaces for iSeries cluster management. All three can be used interchangeably and provide a consistent view of the cluster. The three iCluster interfaces are:

- A Java graphical user interface (GUI) client running on a PC or workstation
- An OS/400 green-screen menu interface
- A full set of OS/400 commands for cluster setup and management

In addition to cluster management and data and object replication, iCluster fully supports ClusterProven applications and traditional iSeries applications. In other words, a ClusterProven application is not needed to take advantage of iCluster’s features and Java interface. iCluster also supports type 3 Cluster Resource Groups (CRGs) (or IASPs or switched disk) as provided in OS/400 V5R1.

iCluster provides traditional high availability functions such as the ability to:

- Check whether objects and data are synchronized across two systems.
- Monitor replication processes, both current and historical, with alarms and alerts to allow for notification when user-defined latency thresholds are exceeded.
- Stop and start replication apply processes while continuing the replication journal scrape processes to allow for backups to be performed on the remote system.
- Define synchronization points in a replication process, with optionally specified user exits to be executed when a synchronization point is reached.
- Define user exits to be executed automatically before or after a group switchover or failover (failure of a group’s primary node).
- Define message queues where messages are placed by iCluster in the event of a failure of a group’s primary node.

The basic steps that are performed to set up and run a cluster using DataMirror iCluster are explained in the following sections. Remember that each step can be performed through either the Java GUI, via green-screen menu options or through commands.

13.2 Getting started with iCluster

Once the system administrator installs DataMirror iCluster on the nodes to form the cluster and the iCluster GUI interface (called the *iCluster Administrator*) is installed on the PC or workstation, a cluster can be setup.

If the iCluster Administrator is used on the PC or workstation, the user logs in with an iSeries user ID and password. The iCluster Administrator main window is presented, as shown in Figure 13-1.

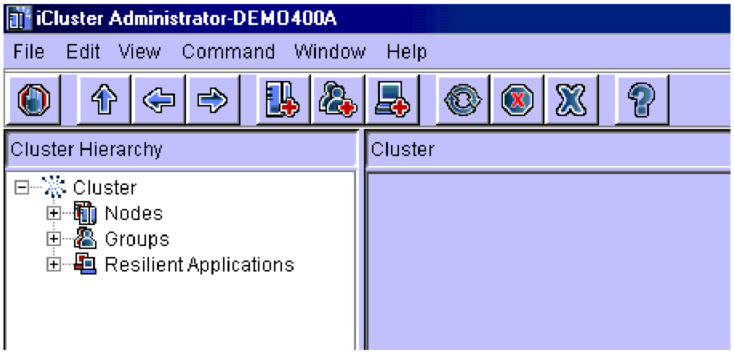

iCluster Administrator-DEMO400A
File Edit View Command Window Help
Cluster Hierarchy
Cluster
Cluster
Nodes
Groups
Resilient Applications

Figure 13-1 The DataMirror iCluster Administrator main window

13.3 Creating a cluster

After the first node in the cluster is defined, a cluster is created. iCluster automatically activates (starts) each node as it is added to the cluster. Nodes can be de-activated (ended) and re-activated (re-started) at any time through either the Java GUI or the commands, for example:

```
DMSTRNODE NODE(DEMO400A)
```

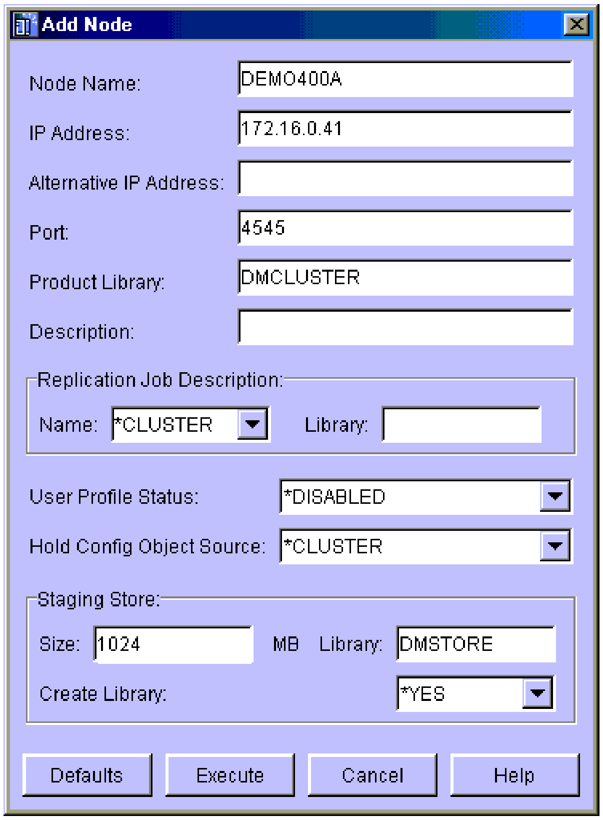
Once the node is started, its status is shown as *ACTIVE. When a node is ended, its status displays as *INACTIVE.

The cluster's first node must be defined as the iSeries server that is currently used. Other nodes in the cluster must be defined from a system that is already an active node in the cluster. To define a node from a system that is not a node in the cluster, create a new cluster with that system as its first node.

The first node defined in the cluster becomes its master node. The master node is responsible for maintaining the information that iCluster needs for data and object replication. This information has to be maintained on all the nodes in the cluster. That way, in the event of a failure or removal of the master node, any other node can automatically assume the role of the master node. For this reason, the master node, or any node that can potentially become the master node, must be directly accessible to all the nodes of the cluster via the TCP/IP interface given when each node is defined.

13.3.1 Adding a node to the cluster

A node can be defined using the Add option on the iCluster Work with Nodes display, the DataMirror iCluster Add Node (DMADDNODE) command, or the iCluster Administrator Add Node input dialog, as shown in Figure 13-2.



Node Name: DEMO400A

IP Address: 172.16.0.41

Alternative IP Address:

Port: 4545

Product Library: DMCLUSTER

Description:

Replication Job Description:

Name: *CLUSTER Library:

User Profile Status: *DISABLED

Hold Config Object Source: *CLUSTER

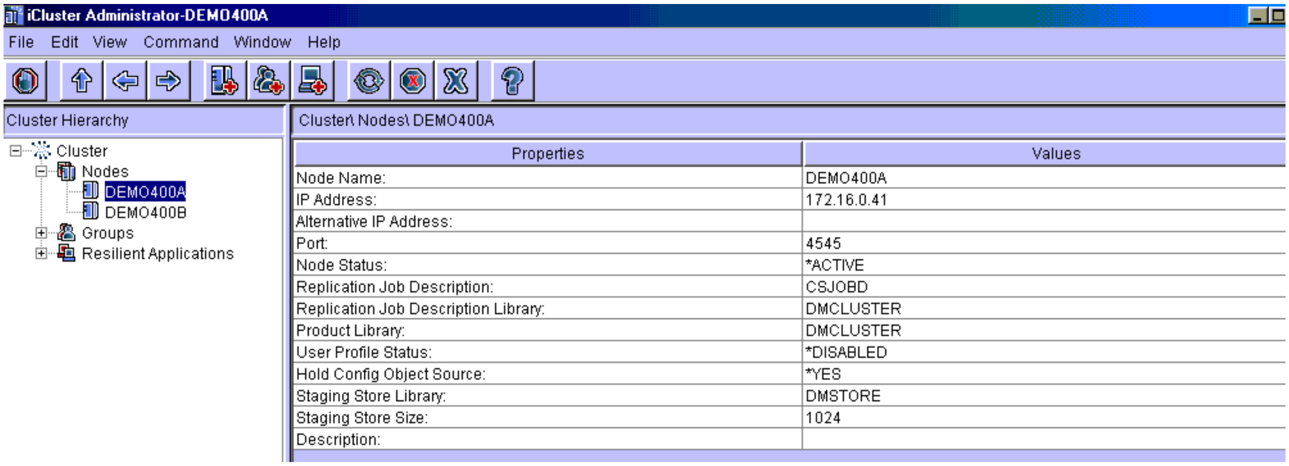
Staging Store:

Size: 1024 MB Library: DMSTORE

Create Library: *YES

Figure 13-2 The iCluster Administrator Add Node input dialog

Expand the node list in the iCluster Administrator main window to view the complete list of nodes in the cluster and their current status (Figure 13-3).



Cluster\ Nodes\ DEMO400A

Properties	Values
Node Name:	DEMO400A
IP Address:	172.16.0.41
Alternative IP Address:	
Port:	4545
Node Status:	*ACTIVE
Replication Job Description:	CSJOBD
Replication Job Description Library:	DMCLUSTER
Product Library:	DMCLUSTER
User Profile Status:	*DISABLED
Hold Config Object Source:	*YES
Staging Store Library:	DMSTORE
Staging Store Size:	1024
Description:	

Figure 13-3 The iCluster Administrator cluster nodes display

View the nodes with the Work with nodes option from the iCluster main menu on the iSeries server.

13.4 Creating and using Cluster Resource Groups

DataMirror iCluster allows creation of data CRGs, application CRGs, and device CRGs that have either one node (the primary) or two nodes (a primary and a backup) in their recovery domain.

Using two node groups, more complex cluster scenarios can be created. For example, a cluster consisting of a single primary node with two or more backup nodes can be setup. Simply create as many data CRGs as there are backup nodes, all with the same primary node, and select the same object specifiers to all the CRGs. See Figure 13-4.

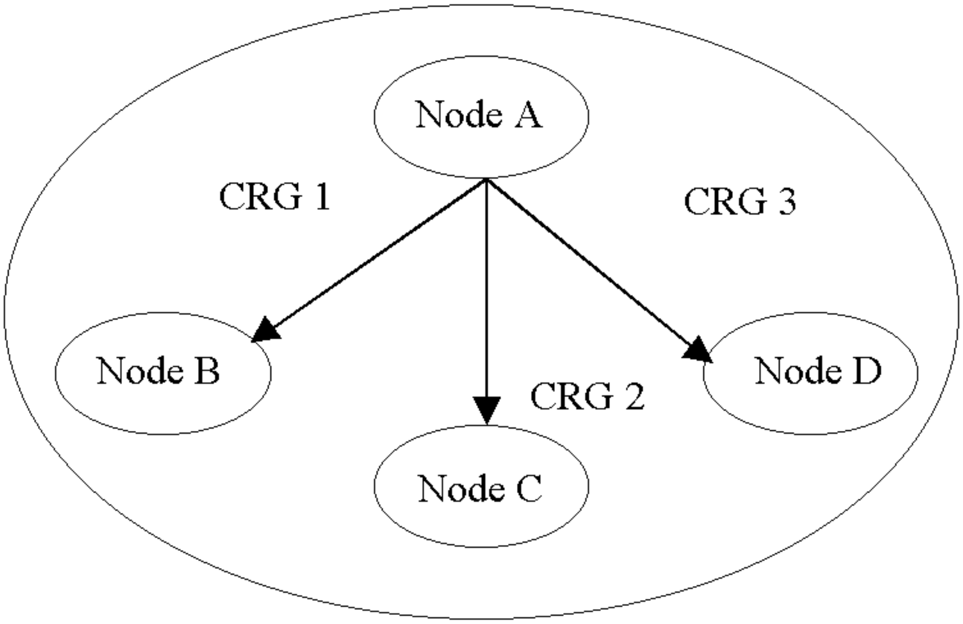


Figure 13-4 Cluster with a single primary node and three backup nodes

13.4.1 Creating data CRGs

A data CRG is created in one of two ways:

- Use the DataMirror iCluster Add Group (DMADDGRP) command, the Add option from the iCluster Work With Groups screen, or the iCluster Administrator Add Group input dialog. This creates a data CRG in the cluster.
- Use the DataMirror iCluster Add Group (DMADDAPP) command, the Add option from the iCluster Work With Resilient Applications screen, or the Add option from the iCluster Administrator Resilient Applications window. This sets up a resilient application that contains one or more data CRGs.

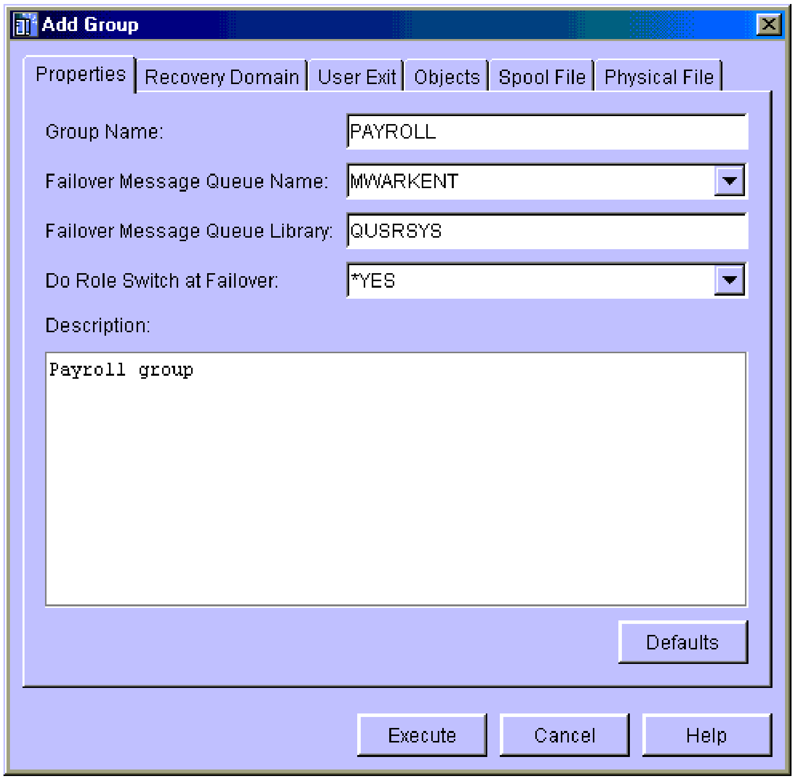
Use the first approach when there is specific, known high-availability requirements for data and objects. This approach allows a direct selection of the objects required for high availability.

The second approach is primarily intended for setting up a ClusterProven resilient application (or an application that has been converted to be “cluster aware”) on the cluster.

Note: ClusterProven applications can be used with any high availability business partner’s cluster software.

DataMirror consultants can help convert regular applications into "cluster aware" applications by creating the QCSTHAAPPI data area and the switching user exit required for iCluster. The remainder of this section deals with data CRGs created using the first approach, that is, as individual groups not associated with any resilient application.

Figure 13-5 shows the input required to create a data CRG with the iCluster Administrator Add Group input dialog.



Group Name: PAYROLL

Failover Message Queue Name: MWARKENT

Failover Message Queue Library: QUSRSYS

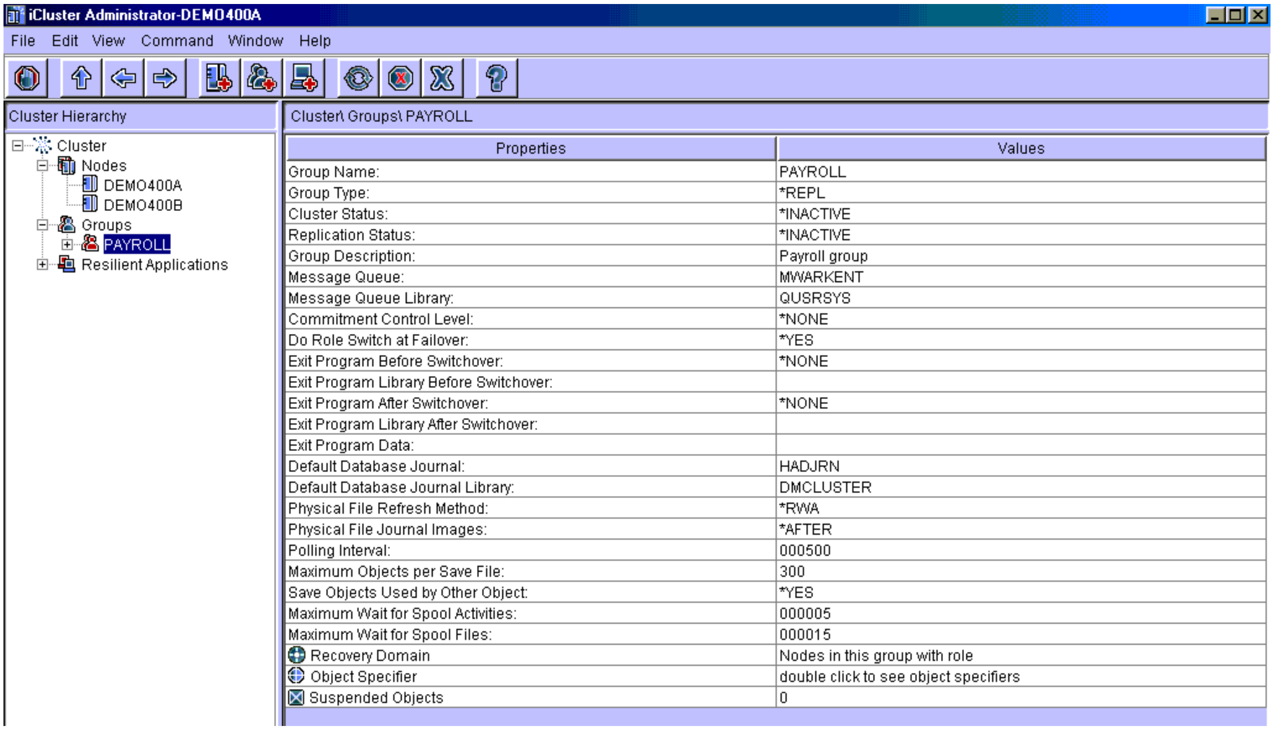
Do Role Switch at Failover: *YES

Description:

Payroll group

Figure 13-5 The iCluster Administrator Add Group input dialog

Note that an alternate message queue name can be specified to receive failover messages (in this case MWARKENT in QUSRSYS). Select other tabs to specify the recovery domain, user exits to be executed before and after a role swap, user-defined parameters to control replication of spooled files, and default database journals. These options are displayed in the iCluster Administrator Groups window as shown in Figure 13-6.



Properties	Values
Group Name:	PAYROLL
Group Type:	*REPL
Cluster Status:	*INACTIVE
Replication Status:	*INACTIVE
Group Description:	Payroll group
Message Queue:	MWARKENT
Message Queue Library:	QUSRSYS
Commitment Control Level:	*NONE
Do Role Switch at Failover:	*YES
Exit Program Before Switchover:	*NONE
Exit Program Library Before Switchover:	
Exit Program After Switchover:	*NONE
Exit Program Library After Switchover:	
Exit Program Data:	
Default Database Journal:	HADJRN
Default Database Journal Library:	DMCLUSTER
Physical File Refresh Method:	*RWA
Physical File Journal Images:	*AFTER
Polling Interval:	000500
Maximum Objects per Save File:	300
Save Objects Used by Other Object:	*YES
Maximum Wait for Spool Activities:	000005
Maximum Wait for Spool Files:	000015
Recovery Domain	Nodes in this group with role
Object Specifier	double click to see object specifiers
Suspended Objects	0

Figure 13-6 The iCluster Administrator Group window

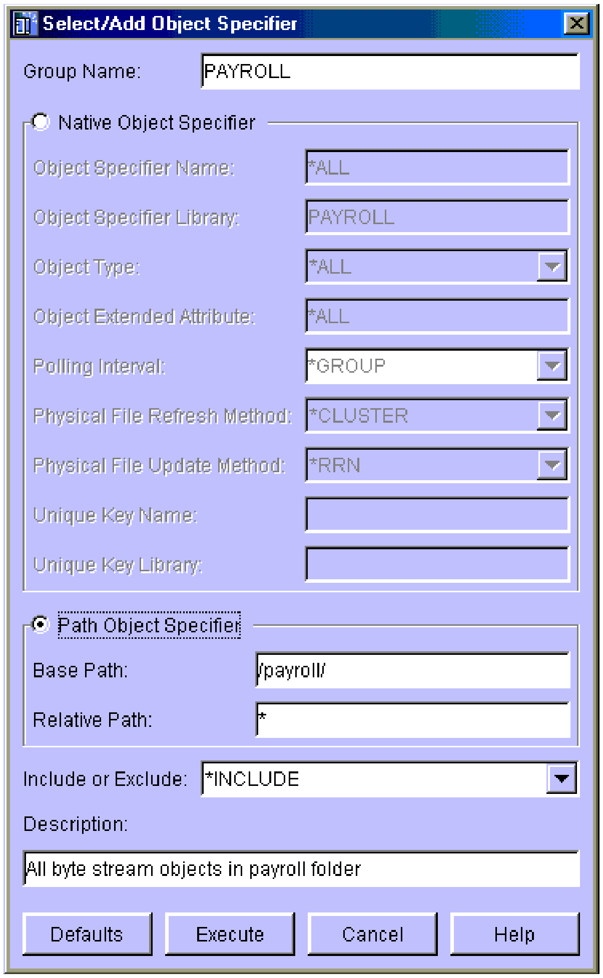
View the groups in the cluster and their current status with either the iCluster Work With Groups screen or the iCluster Administrator Groups window. Note that CRGs are not automatically activated (started) when they are created. A group remains in *INACTIVE status until it is activated. Also data CRGs (group type *REPL) have a second status value – the replication status. The replication status indicates whether replication processes are active for the group. Normally, if a data CRG is in *ACTIVE status, its replication status should also be *ACTIVE.

13.4.2 Selecting objects for a data CRG for high availability

After defining a data CRG, select the objects required for high availability with either the DataMirror iCluster:

- Select Object (DMSELOBJ) command
- Select option on the Work with Object Specifiers by Group screen from the iCluster Work with Groups screen
- Administrator Select/Add Object Specifier input dialog

The iCluster Administrator Select/Add Object Specifier input dialog is shown in Figure 13-7.


Select/Add Object Specifier

Group Name: PAYROLL

○ Native Object Specifier

Object Specifier Name: *ALL

Object Specifier Library: PAYROLL

Object Type: *ALL

Object Extended Attribute: *ALL

Polling Interval: *GROUP

Physical File Refresh Method: *CLUSTER

Physical File Update Method: *RRN

Unique Key Name:

Unique Key Library:

● Path Object Specifier

Base Path: /payroll/

Relative Path: *

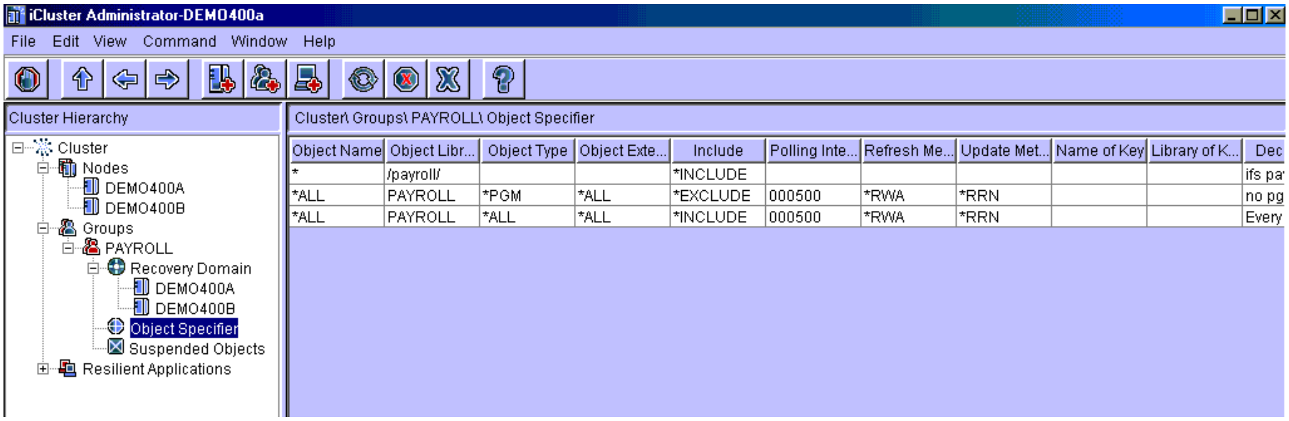
Include or Exclude: *INCLUDE

Description: All byte stream objects in payroll folder

Figure 13-7 The iCluster Administrator Select/Add Object Specifier input dialog

As illustrated in the dialog under the Path Object Specifier radio button, all byte stream files in the payroll folder are selected. Prior to this selection under the Native Object Specifier radio button, all objects in library PAYROLL of all object types had also been selected. Note that exclusions and generics can be used to limit the objects that should be selected to the PAYROLL group. In this case, we selected all objects but excluded *PGM object types.

The objects that match the object specifiers selected to the group are replicated from the group's primary node to its backup node when the group is activated (started). See which object specifiers are selected to a particular group by using the Show Selected function (F16) on the Work with Object Specifiers By Group screen or through the GUI Administrator. See Figure 13-8.



Object Name	Object Libr...	Object Type	Object Exte...	Include	Polling Inte...	Refresh Me...	Update Met...	Name of Key	Library of K...	Dec
*	/payroll/			*INCLUDE						ifs pa
*ALL	PAYROLL	*PGM	*ALL	*EXCLUDE	000500	*RWA	*RRN			no pg
*ALL	PAYROLL	*ALL	*ALL	*INCLUDE	000500	*RWA	*RRN			Every

Figure 13-8 iCluster GUI Administrator Object Specifiers in PAYROLL group

Note that the objects to be replicated do not need to exist when the object specifier is selected to the group. iCluster's real-time auto-registration technology can detect when an object that matches the specifier is created and begins replicating the object as soon as it is created.

Remove (de-select) object specifiers from a data CRG with any of these tools:

- DMDSELOBJ command
- De-select option on the iCluster Work with Object Specifiers by Group screen
- iCluster Administrator Deselect Object input dialog

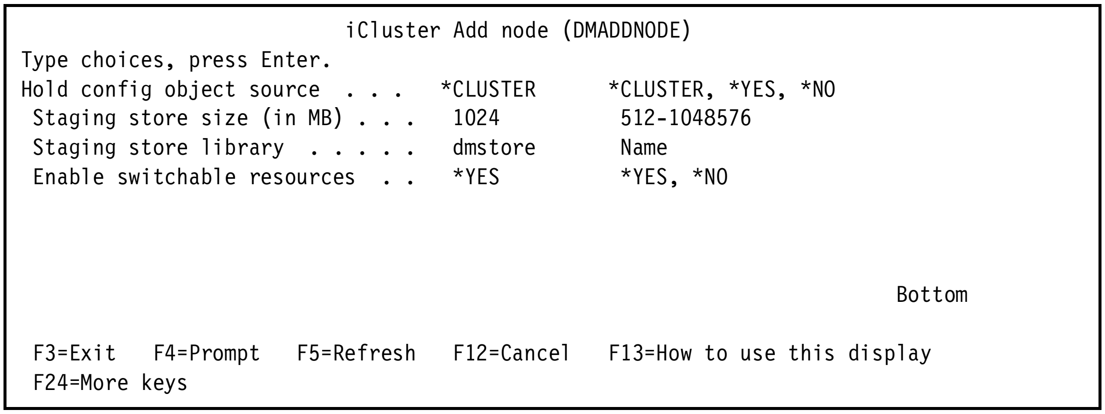
Note: Object specifiers cannot be directly selected or de-selected from CRGs that are part of a resilient application. See 13.4, "Creating and using Cluster Resource Groups" on page 233, for more information.

13.4.3 Creating application CRGs

Application CRGs are created according to the specification in the QCSTHAAPPI automated installation data area architected for application resiliency by IBM. See 13.4, "Creating and using Cluster Resource Groups" on page 233, for details.

13.4.4 Creating device CRGs

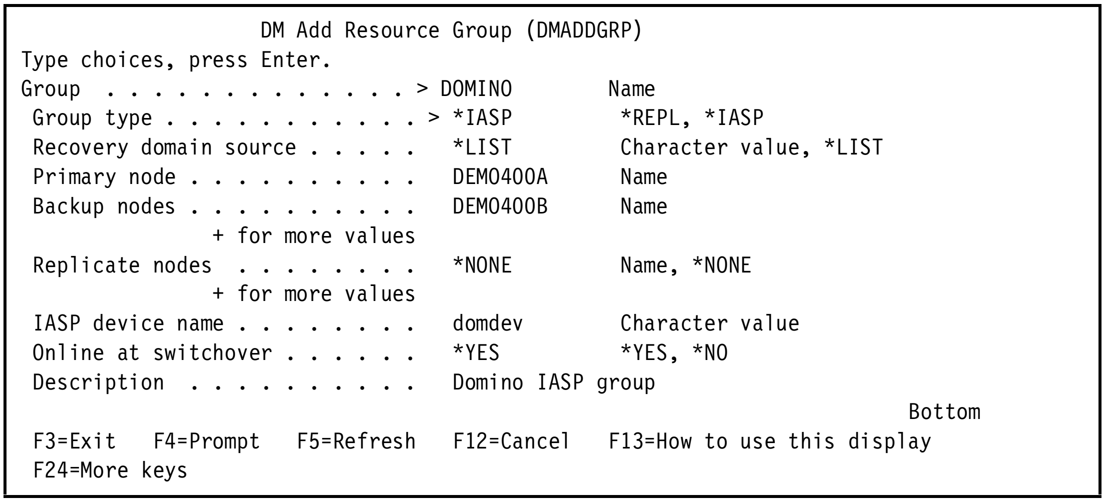
A type 3 CRG can be created in iCluster by first enabling each node in the cluster for switchable resources as shown in Figure 13-9. This is done through either the Add option on the iCluster Work with Nodes screen, the DataMirror iCluster Add Node (DMADDNODE) command, or the iCluster Administrator Add Node input dialog by changing the Enable switchable resources parameter to *YES.



```
                           iCluster Add node (DMADDNODE)
Type choices, press Enter.
Hold config object source  . . .   *CLUSTER      *CLUSTER, *YES, *NO
 Staging store size (in MB) . . .   1024          512-1048576
 Staging store library  . . . . .   dmstore       Name
 Enable switchable resources  . .   *YES          *YES, *NO




                                                                         Bottom

 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 13-9 iCluster Add node display

Once this is done, create an IASP group. This can be done with either the DataMirror iCluster Add Resource Group (DMADDGRP) command, the Add option from the iCluster Work with Groups screen, or the iCluster Administrator Add Group input dialog. Figure 13-10 shows the Add option from the Work With Groups display.



```
                    DM Add Resource Group (DMADDGRP)
Type choices, press Enter.
Group  . . . . . . . . . . . . . > DOMINO        Name
 Group type . . . . . . . . . . . > *IASP         *REPL, *IASP
 Recovery domain source . . . . .   *LIST         Character value, *LIST
 Primary node . . . . . . . . . .   DEMO400A      Name
 Backup nodes . . . . . . . . . .   DEMO400B      Name
               + for more values
 Replicate nodes  . . . . . . . .   *NONE         Name, *NONE
               + for more values
 IASP device name . . . . . . . .   domdev        Character value
 Online at switchover . . . . . .   *YES          *YES, *NO
 Description  . . . . . . . . . .   Domino IASP group
                                                                          Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

Figure 13-10 DM Add Resource Group display

In this example, a new group called *Domino* of type **IASP* is created to connect to a storage device called *domdev*.

13.4.5 Changing a CRG recovery domain

With DataMirror iCluster, a CRG can be initially defined with either one node (the primary) or two nodes (primary and backup) in its recovery domain. If the CRG is defined with a primary node, a backup node can be added later before activating the group. Add and remove backup nodes as necessary when the group is inactive. Select option 5 (Display) on the iCluster Work With Groups screen or select Groups and Display details in the DataMirror iCluster Administrator window to view a group's current recovery domain. Figure 13-8 shows that the recovery domain for the PAYROLL group consists of DEMO400A and DEMO400B.

If a CRG has only one node in its recovery domain (by definition the primary node), add a backup node with the DataMirror iCluster Add backup Node (DMADDBACK) command, the Add Backup option on the Work With Groups screen, or the iCluster Administrator Add Backup input dialog.

If the CRG has two nodes in its recovery, the backup node can be changed by removing it and adding another node as the backup. Use either the DMRMVBACK command, the Remove Backup option on the iCluster Work With Groups screen, or the iCluster Administrator's Remove Backup dialog to remove the existing backup.

Note that a CRG's primary node cannot directly be changed. To change the primary node, perform a switchover on the group so that the current backup node becomes the primary node. (See 13.4.8, "Switching over a data CRG" on page 239.) Or, re-define the group with a different primary node.

13.4.6 Activating or starting a data CRG

A data CRG that is not part of a resilient application can be activated (started) with the DataMirror iCluster:

- Start Group (DMSTRGRP) command
- Start option on the iCluster Work With Groups screen
- iCluster Administrator Start Group input dialog

Note: The application CRGs and data CRGs that are part of a resilient application are activated by activating the resilient application with which they are associated. See 13.5.5, "Activating or starting a resilient application" on page 246.

Once the CRG is activated, its status changes to *ACTIVE. If the CRG has objects or data selected to it, replication of the objects or data from the primary node to the backup node begins and the group's replication status changes to *ACTIVE. Replication activation typically takes longer than CRG activation due to the number of jobs that have to be started.

13.4.7 De-activating or ending a data CRG

Use either the DataMirror iCluster End Group (DMENDGRP) command, the End option on the iCluster Work With Groups display, or the iCluster Administrator End Group input dialog to de-activate or end a data CRG that is not part of a resilient application.

Note: An application CRG and data CRG that are part of a resilient application are de-activated by de-activating the resilient application with which they are associated. See 13.5.6, "De-activating or ending a resilient application" on page 246, for more information

13.4.8 Switching over a data CRG

Switchover is the process of interchanging the primary and backup roles of a CRG's recovery domain and changing the direction of object and data replication in a data CRG. Switchover an active data CRG that is not part of a resilient application with the DataMirror iCluster Start Switch Over (DMSTRSWO) command, the Start Switch Over option on the iCluster Work With Groups screen, or the iCluster Administrator Switch Over Group input dialog.

Switching over a group may not happen instantaneously, particularly if large amounts of objects and data are replicated by the group. Other factors that can increase the time required for switchover to complete are:

- Latency in the apply processes on the backup node
- Switchover user exit processing
- Starting journaling of database files on the new primary node, particularly if many files need to be journaled
- Setting up trigger programs and enabling database file constraints on the new primary node

While a group is undergoing the switchover process, the group's status is displayed as *SWO_PENDING. When switchover is complete and the group starts replicating in the opposite direction, the group's status reverts to *ACTIVE.

Note: Switchover application CRGs and data CRGs that are part of a resilient application by switching over the resilient application with which they are associated. See 13.5.7, "Switching over a resilient application" on page 246.

13.4.9 Problem solving

iCluster's GUI administrator includes capabilities to assist in problem solving. Two such tools are:

- **Eventlogs**: A physical file that contains status messages and error conditions.
- **Status Monitor**: Displays latency and journal information. It provides alarms and alerts on critical latency thresholds or error messages.

Combine these instruments with additional external message queues, and monitor with third-party packages to further assist in problem solving.

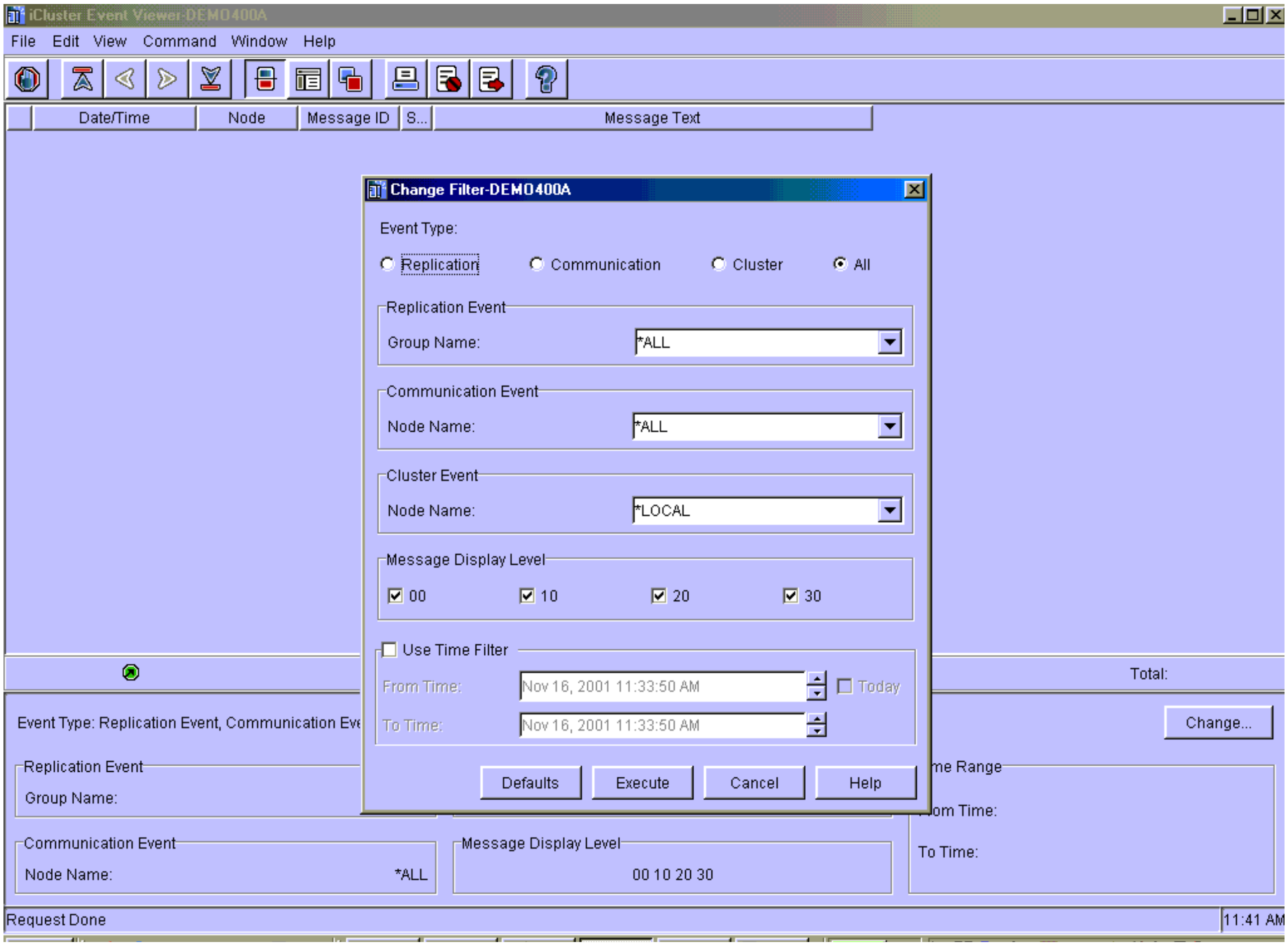
One of the nicest features of the iCluster Administrator is the GUI display of the event viewer. From the GUI Administrator, select Window New Event Viewer and optionally filter which messages to display on the PC or workstation.

Figure 13-11 shows the Change Filter dialog.

Several message filtering options are available. You can choose to display:

- Only replication messages
- Communications messages
- Only clustering messages
- All messages

Selective filtering by group or node is possible within each of these categories. Messages can be filtered based on severity level or even by time.



Change Filter-DEMO400A

Event Type:

○ Replication ○ Communication ○ Cluster ● All

Replication Event
Group Name: *ALL

Communication Event
Node Name: *ALL

Cluster Event
Node Name: *LOCAL

Message Display Level
☑ 00 ☑ 10 ☑ 20 ☑ 30

□ Use Time Filter
From Time: Nov 16, 2001 11:33:50 AM □ Today
To Time: Nov 16, 2001 11:33:50 AM

Defaults Execute Cancel Help

Event Type: Replication Event, Communication Eve

Replication Event
Group Name:

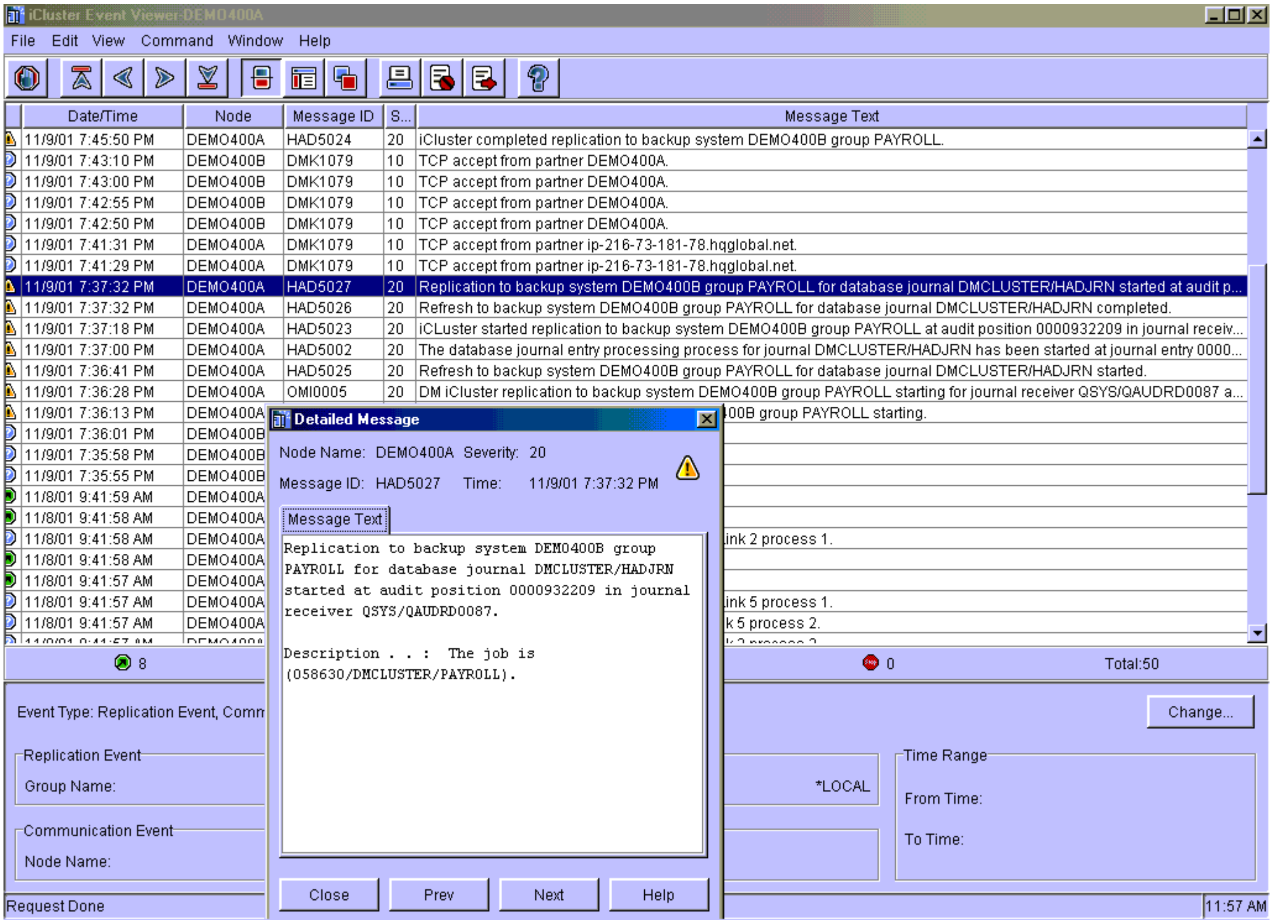
Communication Event
Node Name: *ALL

Message Display Level
00 10 20 30

me Range
om Time:
To Time:

Figure 13-11 iCluster GUI Event Viewer Change Filter display

Once filtering is selected, all of the eventlog messages from the iSeries server are brought down in buffers to the PC or Workstation. Double-click the message to see second level help text. See Figure 13-12 for an illustration.


iCluster Event Viewer-DEMO400A
File Edit View Command Window Help

	Date/Time	Node	Message ID	S...	Message Text
⚠	11/9/01 7:45:50 PM	DEMO400A	HAD5024	20	iCluster completed replication to backup system DEMO400B group PAYROLL.
	11/9/01 7:43:10 PM	DEMO400B	DMK1079	10	TCP accept from partner DEMO400A.
	11/9/01 7:43:00 PM	DEMO400B	DMK1079	10	TCP accept from partner DEMO400A.
	11/9/01 7:42:55 PM	DEMO400B	DMK1079	10	TCP accept from partner DEMO400A.
	11/9/01 7:42:50 PM	DEMO400B	DMK1079	10	TCP accept from partner DEMO400A.
	11/9/01 7:41:31 PM	DEMO400A	DMK1079	10	TCP accept from partner ip-216-73-181-78.hqglobal.net.
	11/9/01 7:41:29 PM	DEMO400A	DMK1079	10	TCP accept from partner ip-216-73-181-78.hqglobal.net.
⚠	11/9/01 7:37:32 PM	DEMO400A	HAD5027	20	Replication to backup system DEMO400B group PAYROLL for database journal DMCLUSTER/HADJRN started at audit p...
⚠	11/9/01 7:37:32 PM	DEMO400A	HAD5026	20	Refresh to backup system DEMO400B group PAYROLL for database journal DMCLUSTER/HADJRN completed.
⚠	11/9/01 7:37:18 PM	DEMO400A	HAD5023	20	iCLuster started replication to backup system DEMO400B group PAYROLL at audit position 0000932209 in journal receiv...
⚠	11/9/01 7:37:00 PM	DEMO400A	HAD5002	20	The database journal entry processing process for journal DMCLUSTER/HADJRN has been started at journal entry 0000...
⚠	11/9/01 7:36:41 PM	DEMO400A	HAD5025	20	Refresh to backup system DEMO400B group PAYROLL for database journal DMCLUSTER/HADJRN started.
⚠	11/9/01 7:36:28 PM	DEMO400A	OMI0005	20	DM iCluster replication to backup system DEMO400B group PAYROLL starting for journal receiver QSYS/QAUDRD0087 a...
⚠	11/9/01 7:36:13 PM	DEMO400A			00B group PAYROLL starting.
	11/9/01 7:36:01 PM	DEMO400B			
	11/9/01 7:35:58 PM	DEMO400B			
	11/9/01 7:35:55 PM	DEMO400B			
	11/8/01 9:41:59 AM	DEMO400A			
	11/8/01 9:41:58 AM	DEMO400A			
	11/8/01 9:41:58 AM	DEMO400A			ink 2 process 1.
	11/8/01 9:41:58 AM	DEMO400A			
	11/8/01 9:41:57 AM	DEMO400A			
	11/8/01 9:41:57 AM	DEMO400A			ink 5 process 1.
	11/8/01 9:41:57 AM	DEMO400A			k 5 process 2.

8 | 0 | Total:50

Event Type: Replication Event, Comm

Replication Event
Group Name:

Communication Event
Node Name:

*LOCAL

Time Range
From Time:
To Time:

Request Done | 11:57 AM

Detailed Message

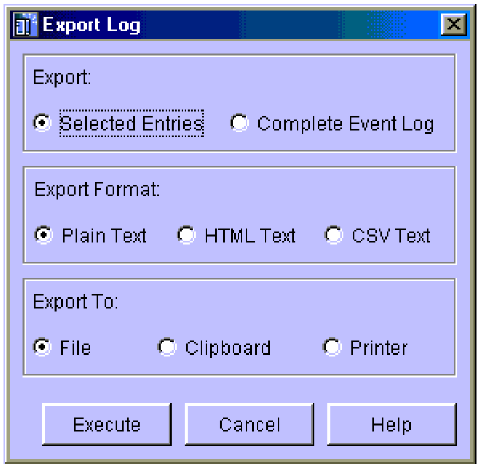
Node Name: DEMO400A Severity: 20
Message ID: HAD5027 Time: 11/9/01 7:37:32 PM

Message Text

```
Replication to backup system DEM0400B group
PAYROLL for database journal DMCLUSTER/HADJRN
started at audit position 0000932209 in journal
receiver QSYS/QAUDRD0087.

Description . . :   The job is
(058630/DMCLUSTER/PAYROLL).
```

Figure 13-12 iCluster GUI Event Viewer Detailed Message display

Once the message is displayed, right-click the message to print or export either the selected message, or messages, or all messages. Export the messages as plain text, HTML, or CSV format, or it can be brought into a file, the clipboard, or directly to the printer. See Figure 13-13 for an illustration.


Export Log

Export:
● Selected Entries ○ Complete Event Log

Export Format:
● Plain Text ○ HTML Text ○ CSV Text

Export To:
● File ○ Clipboard ○ Printer

Figure 13-13 iCluster GUI Event Viewer Export Log dialogue

13.5 DataMirror ClusterProven applications

iCluster provides a simple interface to implement ClusterProven applications in a cluster. Resilient applications can be setup, started, ended, switched over, changed, updated, and removed with DataMirror iCluster. Once the application is setup, a CRG cannot be added or removed from a resilient application. Cluster operations on the groups that comprise the resilient application cannot be performed individually, but can as part of the resilient application.

Refer to Chapter 8, “ClusterProven for iSeries applications” on page 161, to further understand the iSeries ClusterProven identify.

13.5.1 Setting up a resilient application

A resilient application can be setup with either one node (the primary) or two nodes (a primary and a backup) in its recovery domain with DataMirror iCluster. Considerations to set up a resilient application include:

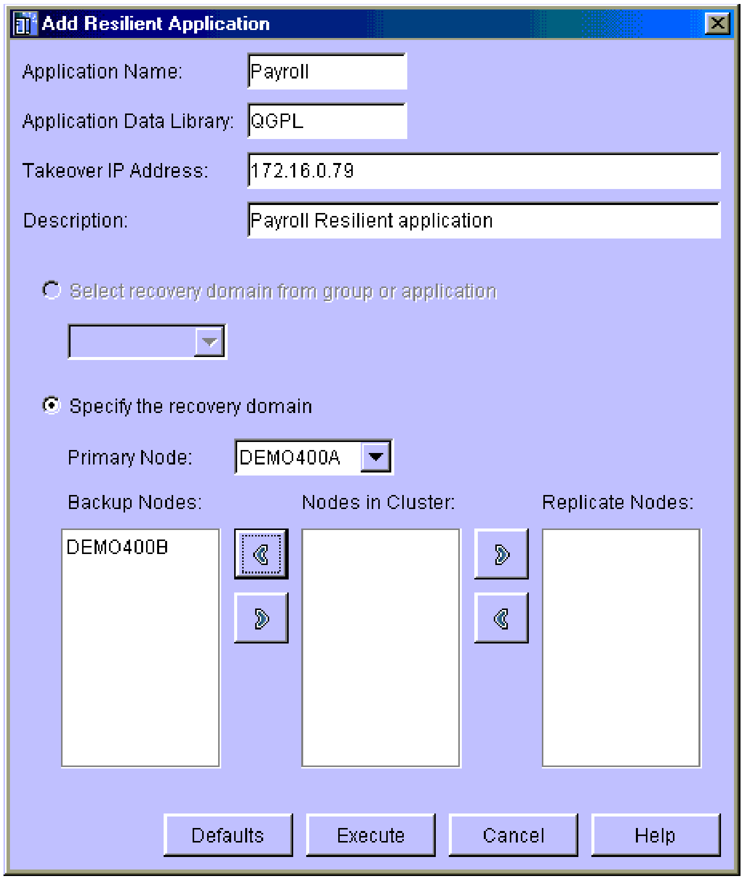
- The takeover IP address of the resilient application.
- The name of the ClusterProven application’s installation library on the systems.

  The library used to install the application must exist on all nodes of the application’s recovery domain and must contain the QCSTHAAPPI data area. This data area defines what CRGs are to be created for the resilient application and what object specifiers are to be selected to the application’s data CRGs. QCSTHAAPPI is provided by the application vendor.

DataMirror provides these options to set up a resilient application:

- The iCluster Add Application (DMADDAPP) command
- The Add option on the iCluster Work With Resilient Applications display
- The iCluster Administrator Add Resilient Application input dialog

Figure 13-14 shows the iCluster Administrator Add Resilient Application input dialog.



Application Name: Payroll

Application Data Library: QGPL

Takeover IP Address: 172.16.0.79

Description: Payroll Resilient application

○ Select recovery domain from group or application

● Specify the recovery domain

Primary Node: DEMO400A

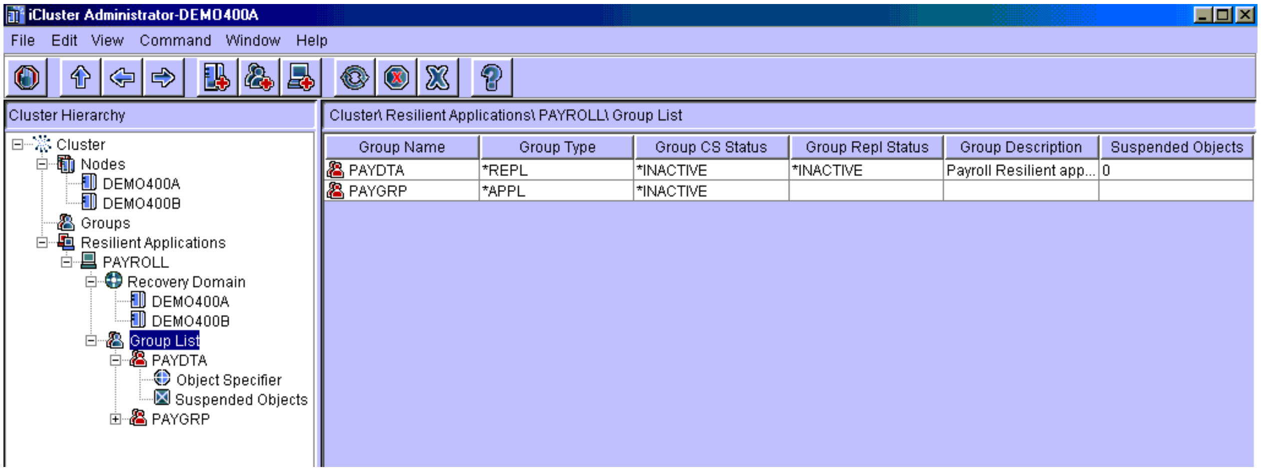
Backup Nodes: DEMO400B

Nodes in Cluster:

Replicate Nodes:

Figure 13-14 iCluster GUI Add Resilient Application input dialogue

After a resilient application is created, it appears on the iCluster Work With Resilient Applications screen or the iCluster Administrator Resilient Applications window, as illustrated in Figure 13-15.



```
Cluster
  Nodes
    DEMO400A
    DEMO400B
  Groups
  Resilient Applications
    PAYROLL
      Recovery Domain
        DEMO400A
        DEMO400B
      Group List
        PAYDTA
          Object Specifier
          Suspended Objects
        PAYGRP
```

Cluster\ Resilient Applications\ PAYROLL\ Group List

Group Name	Group Type	Group CS Status	Group Repl Status	Group Description	Suspended Objects
PAYDTA	*REPL	*INACTIVE	*INACTIVE	Payroll Resilient app...	0
PAYGRP	*APPL	*INACTIVE			

Figure 13-15 The iCluster Resilient Applications window

Choose the Work With Groups option on the iCluster Work With Resilient Applications screen to view the list of groups that are associated with an application. Figure 13-15 shows that there are two groups associated with the Payroll resilient application, PAYDTA (a data CRG) and PAYGRP (an application CRG).

The display as shown in Figure 13-15 also displays the status of the groups associated with the resilient application. Note that the replication status field of group PAYGRP is blank. This indicates that PAYGRP is an application CRG (type *APPL), not a data CRG (type *REPL) like the PAYDTA group.

13.5.2 Selecting objects to a resilient application

The object specifiers required for a resilient application are listed in a file that is named in the QCSTHAAPPI data area for the application. iCluster reads this file when defining the resilient application and automatically selects the object specifiers to the appropriate data CRGs that are associated with the resilient application.

De-selecting object specifiers from a resilient application or a group that is associated with a resilient application is done automatically by iCluster when the application is updated or removed.

13.5.3 Changing or updating a resilient application

A resilient application's takeover IP address and its description directly can be changed with DataMirror's iCluster:

- Change Application (DMCHGAPP) command
- Change option on the Work with Resilient Applications screen
- Administrator Change Resilient Application input dialog

A resilient application's recovery domain can also be changed. See 13.5.4, "Changing a resilient application's recovery domain" on page 245.

However, no other parts of a resilient application's definition can be changed directly. To change any other aspect of a resilient application's definition (for example, the object specifiers selected for replication or the number of groups associated with the application), the application must be updated. The update process removes the groups currently associated with the application and reads the application's QCSTHAAPPI data area to re-define the groups and re-select the object specifiers required for the application.

Use the DataMirror iCluster Update Application (DMUPDAPP) command, the Update option on the iCluster Work With Resilient Applications screen, or the iCluster Administrator Update Resilient Application input dialog to update a resilient application.

When the ClusterProven application is upgraded with a new version supplied by the application vendor, the upgrade may also include some changes to the resilient application's definition. The application vendor provides a new QCSTHAAPPI data area to take account of these changes. In this situation, update the resilient application using the method described in the previous paragraph.

13.5.4 Changing a resilient application's recovery domain

Using iCluster, a resilient application can be defined with either one (the primary) or two nodes (a primary and a backup) in its recovery domain. Backup nodes can be added and removed as necessary when the resilient application is inactive.

Add a backup node with either the DMADDBACK command, the Add Backup option on the iCluster Work With Resilient Applications screen, or the iCluster Administrator Add Backup Node input dialog if a resilient application has only a primary node in its recovery domain.

Remove the backup node with either the DMRMVBACK command, the Remove Backup option on the iCluster Work With Resilient Applications screen, or the iCluster Administrator Remove Backup Node input dialog, if a resilient application has two nodes in its recovery domain.

A resilient application's primary node cannot be changed. To change the primary node, perform a switchover on the resilient application so that the current backup node becomes the primary node. Or re-define the resilient application with a different primary node.

13.5.5 Activating or starting a resilient application

Use the DataMirror iCluster Start Application (DMSTRAPP) command, the Start option on the iCluster Work With Resilient Applications screen, or the iCluster Administrator Start Resilient Application input dialog to activate or start a resilient application.

If the resilient application has data CRGs with objects selected to them, replication is also activated for those CRGs.

13.5.6 De-activating or ending a resilient application

Use the DataMirror iCluster End Application (DMENDAPP) command, the End option on the iCluster Work With Resilient Applications screen, or the iCluster Administrator End Resilient Application input dialog to de-activate or end a resilient application.

13.5.7 Switching over a resilient application

Switchover an active resilient application with either the DataMirror iCluster Switch Over Application (DMSWOAPP) command, the Start Switch over option on the iCluster Work With Resilient Applications screen, or the iCluster Administrator Switchover Resilient Application input dialog.

13.6 Removing the cluster and its components

This section describes how to handle some of the on-going cluster management tasks. Once the cluster is up and running, some of its components or even the cluster itself may need to be removed. This is how to carry out these tasks.

13.6.1 Removing a resilient application

Remove a resilient application with either the DataMirror iCluster Remove Application (DMRMVAPP) command, the Remove option on the iCluster Work With Resilient Applications screen, or the iCluster Administrator Remove Resilient Application input dialog.

When a resilient application is removed, the object specifiers selected to the application are also removed. However, the objects that were replicated by the application's CRGs are not affected on either the primary or backup node of the application.

13.6.2 Removing a data CRG

Remove a data CRG that is not associated with a resilient application with either the DataMirror iCluster Remove Group (DMRMVGRP) command, the Remove option on the iCluster Work With Groups screen, or the iCluster Administrator Remove Group input dialog.

When a data CRG is removed, the object specifiers that were selected to the group are also removed, but the objects that were replicated by the group are not affected on either the primary or backup node of the group.

13.6.3 Removing a node from the cluster

A node can be removed from the cluster at any time. Remove a node when it is active to ensure a complete cleanup of cluster data from the node. If any cluster data is left over after the node is removed from the cluster, it can lead to difficulties if the node is added to a new cluster or the same cluster at a later time.

Remove a node from the cluster with the DataMirror iCluster Remove Node (DMRMVNODE) command, the Remove option on the iCluster Work With Nodes display, or the iCluster Administrator Remove Node input dialog.

13.6.4 Removing the entire cluster

Remove all resilient applications, data CRGs and nodes from the cluster with the DataMirror iCluster Delete Cluster (DMDLTCLSTR) command. This command can be invoked on the command line or as an option in the DataMirror iCluster Commands (DMCMDS) menu, which is accessible from any iCluster menu or screen on the iSeries server.

Note that removing the entire cluster only means that the cluster is de-activated, and nodes and other cluster components are removed. The objects that were replicated by the cluster's data CRGs are not affected. The iCluster product itself is still accessible as it was before the cluster was created.

If the cluster is not partitioned and all the nodes in the cluster are active, call the DataMirror iCluster Delete Cluster (DMDLTCLSTR) command on one node to remove the entire cluster. However, if the cluster is partitioned, call the DMDLTCLSTR command once in the primary partition and once on each node in the secondary partitions. Similarly, if any nodes in the cluster are inactive, call this command on each inactive node of the cluster and in the active part of the cluster.

The DMDLTCLSTR command can be used to delete any cluster.

13.7 Using iCluster commands to access Cluster Services operations

Most iCluster commands correspond directly to an OS/400 Cluster Services operation or API. Use the iCluster commands to access the OS/400 Cluster Services operations, for example, when recovering from a partition or node failure.

Chapter 12, "Problem determination for iSeries clustering" on page 215, describes the cluster operations that are allowed in a cluster partition situation and shows how to recover from a node failure with Cluster Services operations.

Recovery from a cluster partition or node failure can be performed with the iCluster commands that map to the Cluster Services operations.

Table 13-1 lists the mapping between the Cluster Services APIs and the DataMirror iCluster commands.

Table 13-1 Mapping Cluster Services operations to iCluster commands

Cluster Services operation	iCluster command
Add a node to the cluster	DMADDNODE
Change a cluster node	DMCHGNODE
Remove a node from the cluster	DMRMVNODE
Start a cluster node	DMSTRNODE
End a cluster node	DMENDNODE
Delete the cluster	DMDLTCLSTR

Table 13-2 lists the mapping between Cluster Resource Group operations and iCluster commands.

Table 13-2 Mapping Cluster Resource Group operations to iCluster commands

Cluster Resource Group operation	iCluster commands
Create a Cluster Resource Group	DMADDGRP, DMADDAPP
Change a Cluster Resource Group	DMCHGGRP, DMCHGAPP
Delete a Cluster Resource Group	DMRMVGRP, DMRMVAPP
Add node to recovery domain	DMADDBACK
Remove node from recovery domain	DMRMVBACK
Start a Cluster Resource Group	DMSTRGRP, DMSTRAPP
End a Cluster Resource Group	DMENDGRP, DMENDAPP
Initiate switchover	DMSTRSWO, DMSWOAPP

13.8 For more information

For the latest information on DataMirror and iCluster, visit the DataMirror site on the Web at: `http://www.datamirror.com`

14

Lakeview Technology MIMIX

Lakeview Technology is an IBM Business Partner specializing in availability management for the IBM @server iSeries server.

Lakeview's wide array of solution offerings includes the MIMIX® suite of products:

- **MIMIX Replicator™**: For near real-time replication of data and objects
- **MIMIX Monitor™**: For automated system management and switching in non-clustered environments
- **MIMIX Promoter™**: Supports continuous operations during planned downtime events

For clustered iSeries environments, Lakeview offers *MIMIX Cluster Server*™ as an integrated part of MIMIX Replicator™. Implementing MIMIX Cluster Server provides a robust clustering environment for data and application resiliency.

Lakeview also offers *MIMIX FastPath*™, an exclusive Lakeview Technology Solution Services offering designed to enable applications to work in clustered iSeries environments with little or no modification of the application's original code. This service is especially useful for Independent Software Vendors (ISVs), in-house application development shops, or customers who want to use clustering for their entire high availability (HA) environment.

14.1 MIMIX Cluster Server

MIMIX Cluster Server, from Lakeview Technology, offers a new and completely integrated clustering solution for the availability of applications and data; centralized cluster management; and worldwide, single-point-of-contact support 24 hours a day, 7 days a week. MIMIX Cluster Server includes a Java-based GUI Cluster Manager, high performance MIMIX Optimizer, and MIMIX Replicator high availability software. When this software is coupled with MIMIX Fastpath and Lakeview's Solution Services for clustering end users, the result is a totally integrated environment that delivers a robust high availability solution capable of rapid, coordinated planned switchovers, and unplanned failovers.

Cluster Server's industry-leading functionality is designed in direct response to customer requirements and reflects Lakeview's proven track record of over eleven years in the high availability marketplace. Some examples of the extensive range of functionality found in Cluster Server include:

- Automated configuration of MIMIX for ClusterProven or a cluster-enabled application, providing faster deployment of the solution and thereby further improving Return on Investment (ROI).
- Automated creation of the resilient objects for ClusterReady or ClusterProven applications. Creating resilient objects from a fully functional, switch-tested MIMIX configuration eliminates any doubt that the application can fully participate in the clustering environment. Not only does this apply to the initial creation of resilient objects for an application, it also allows resilient objects to be quickly re-generated and updated to keep pace with changes to the deployed, in-production clustering solution.
- Total integration of the clustering environment including all applications and their associated objects as well as all the other objects, system settings, and values that are necessary to provide a complete HA solution. The clustering architecture addresses a good portion of the total solution, but does not stipulate how these other objects and environmental issues should be managed. Cluster Server addresses the totality of the HA environment.
- Complete customizing of the application exit program or programs created by the FastPath Service. In the application world, one size does not fit all and customizing is the norm. Cluster Server recognizes and embraces this reality by allowing end-users the flexibility to "have it their way". For example, a ClusterProven core application can be surrounded by a number of other off-the-shelf or in-house developed applications. Customizing allows the core application exit program to also coordinate and manage the restart and recovery sequencing of any ancillary applications. In addition to the application exits, the MIMIX Cluster Server's data exit programs can be fully customized.
- Automated distribution of resilient objects for clustering to all nodes in a recovery domain. This simplifies the initial deployment of the clustering solution as well as the management of subsequent solution updates.
- Automated search for resilient objects for clustering – no need for the end-user to specify the application libraries where these objects reside.
- Flexibility to manage specific components of an application suite or the entire application suite as an entity. It is not an "all-or-nothing" proposition with MIMIX Cluster Server. There are choices and options.
- End-user specification of the data relationships between the various nodes within a recovery domain. MIMIX Cluster Server's data provider configuration option is especially useful in managing the data relationships after a switching event occurs. This capability provides greater flexibility than the clustering architecture behaviors would otherwise dictate.
- Request Start from any participating node.

  Normally, a node can only be started by a request originating from an active node. Cluster Server determines which nodes in the cluster are active and starts the requesting node on the correct system automatically. This means a start request is successful from any node participating in the cluster.
- Switchover/Failover intervention allows customers better control over the clustering solution behavior – beyond what is provided by the architecture. This is especially useful when, for example, you simply want to take the primary system down without switching to the backup – perhaps the primary is simply being IPLed and is to be back up quickly.

- MIMIX Optimizer is a generalized automation tool that can be used to programmatically automate virtually any iSeries function or event – independently or in collaboration with clustering.

14.1.1 Implementing clustering with the MIMIX Cluster Server

Using MIMIX Cluster Server simplifies the process of establishing and maintaining a total clustering environment. Figure 14-1 presents an overview of how to enable a clustering solution using MIMIX.

Once MIMIX Replicator is operational, new and separate system definitions, transfer definitions, journal definitions, or data group definitions are not necessary to enable clustering. Clustering uses the existing definitions that are used with MIMIX Replicator. Some existing definitions may need to be modified to designate the systems and communications to be used by the cluster.

OS/400 Cluster Resource Services are used to enable cluster functions and provide programming interfaces that are used to create and manage a cluster. MIMIX uses the Start MIMIX Manager (STRMMXMGR) command to start the OS/400 Cluster Resource Services.

Once the systems are identified that are to participate in the cluster, and how they are to communicate with each other, create an application group. An application group defines application and data resiliency for an application to be included in the cluster environment.

With existing cluster-enabled applications, use the Load Application Groups (LODAG) command to save time. The LODAG command searches the libraries on the local system for cluster-enabled applications. When the LODAG command encounters these applications, it creates application groups and data CRG entries for each unique cluster-enabled application it finds. The LODAG command also attempts to add a node entry for the *LOCAL system.

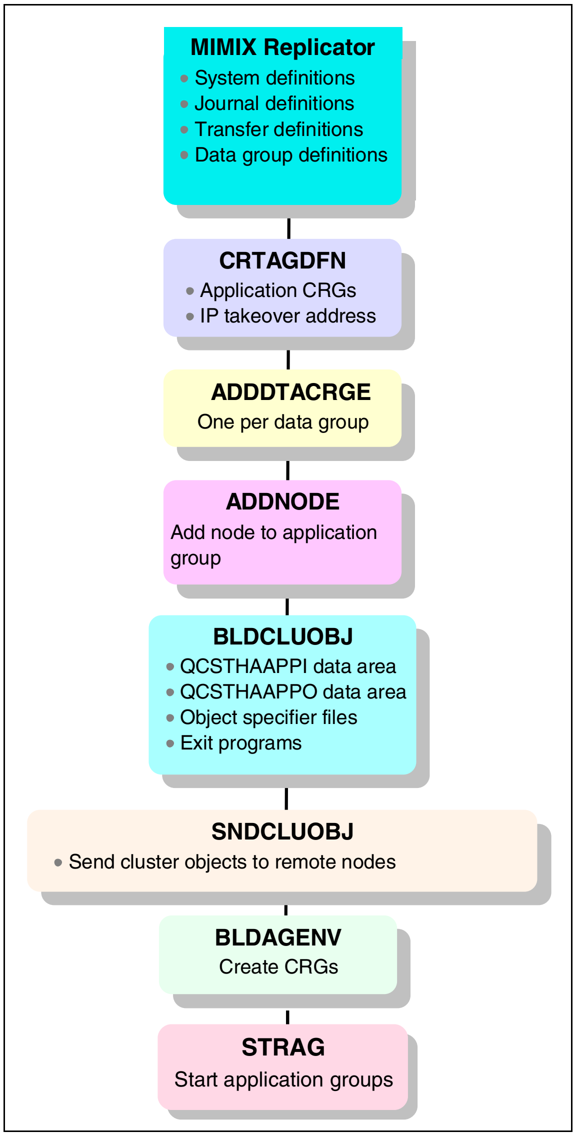


Figure 14-1 Cluster enable with MIMIX

Next run the Create MIMIX Definitions (CRTMMXDFN) command to populate the MIMIX environment with the data group definitions, journal definitions, and other entries associated with the data groups that are created. These entries are derived from the Object Specifier Files (OSF) associated with the data CRG entry.

If there are no cluster-enabled applications, create a data CRG for each MIMIX Replicator data group to define the data to be made resilient. MIMIX assumes that the first part of the name of the data group represents one set of data regardless of how many data groups are defined. For example, with a data group named ACCOUNTING SYS1 SYS2 and another named ACCOUNTING SYS2 SYS3, MIMIX assumes that these data groups contain the same set of data and represents that data with only one data CRG.

Once the applications and data to be included in the cluster are identified, designate which nodes (systems) are to be used by the application group. MIMIX uses the Add Node Entry (ADDNODE) command to enable assignment of a system to the cluster and to an application group, and to designate what role (primary, backup, or replicate) the system is to play in the application group.

MIMIX uses the Build Cluster Objects (BLDCLUOBJ) command to cluster-enable a MIMIX configured clustering application. This command creates the resilient objects for clustering that are required for cluster enabling an application, based on the MIMIX configuration.

Once a cluster-enabled application is created, distribute it to the other systems in the cluster. MIMIX uses the Send Cluster Objects (SNDCLUOBJ) command for this purpose. The SNDCLUOBJ command uses the functionality in the MIMIX Replicator Send Network Object (SNDNETOBJ) command to distribute the cluster objects to the other systems in the cluster.

When the cluster is operational, these actions are possible:

- Add additional nodes to the cluster
- Change the role of the nodes associated with an application group
- End an application group
- Perform other cluster management tasks

When the application group definition is complete, use the Build Application Environment (BLDAGENV) command to create the application CRG and data CRGs based on the application group specified. This command builds all the child application groups defined for a parent application group. Note that this command does not create the cluster. Run the Create Mimix Cluster (CRTMMXCLU) command if the cluster does not exist.

14.1.2 Technical support

In a multi-vendor clustered environment, it can be difficult to isolate the cause of a problem so that the appropriate vendor can be called for assistance. With MIMIX Cluster Server, Lakeview Technology and its worldwide network of partners provide “Follow the Sun” Support for the entire cluster through a worldwide single point-of-contact.

For more information about Lakeview Technology, MIMIX, or iSeries Clustering services, visit the Lakeview Technology Web site at: http://www.lakeviewtech.com

14.2 MIMIX FastPath

MIMIX FastPath services prepare applications to take advantage of clustering, a process referred to as “cluster enabling”, with little or no modification to the application’s original code. Applications enabled through MIMIX FastPath are prepared for continuous availability solutions. For ISVs, MIMIX FastPath is the service that prepares applications to achieve IBM ClusterProven branding.

14.2.1 Resilient objects for clustering

To take advantage of iSeries clustering, three elements must be in place:

- OS/400 V4R4 (or higher)
- High availability replication software, such as MIMIX Replicator
- Cluster enabled applications

To support clustering, applications undergo a detailed evaluation and analysis to determine which of their many program components are critical for re-establishing operations on a backup node after a switch. Lakeview's customized object specifier file is used to allow the iCluster middleware product and OS/400 to work in concert with the application to provide the ultimate in continuous availability.

When considering the effort to cluster enable applications, the critical issues for developers are the development time, the resource investment, and the high availability expertise required when modifying applications to take advantage of the Cluster Resource Services, as well as the ongoing support of those modifications. Many ISVs and in-house developers find these challenges too great to readily adopt any clustering technology. MIMIX FastPath provides an optimal solution by providing services that cluster enable applications with a minimal amount of effort and investment.

Note: The IBM ClusterProven process is described in Chapter 8, "ClusterProven for iSeries applications" on page 161.

14.2.2 MIMIX FastPath services

MIMIX FastPath for iSeries is an exclusive Lakeview Solution Services offering. This offering consists of tools, services, support, Managed Availability Services, and clustering expertise designed to fully enable applications to work in an iSeries clustering environment with little or no modification of the application code.

Lakeview Technology Solution Services consultants begin the MIMIX FastPath process by determining the level of availability required, assessing the application, and creating the ROCs necessary to cluster-enable the application and achieve ClusterReady status. The process is completed with professional skills training, extensive testing in a fully operational clustered iSeries environment, and ClusterReady certification.

It also includes a report that details any additional work that may be required to achieve IBM ClusterProven or Advanced ClusterProven branding for the application.

14.2.3 MIMIX FastPath support

Maintaining the clustering modifications throughout the life of the application is a significant challenge. MIMIX FastPath provides a single point of worldwide support for ongoing modifications, updates, and upgrades to the MIMIX FastPath work through the life of an agreement. This assures compliance with new releases and fixes of the operating system, application version changes, and other required updates, while freeing critical ISV resources to focus on the development of core product functionality.

14.2.4 MIMIX FastPath Professional Services

As part of its ongoing support as application developers and Cluster Server implementations, Lakeview Technology offers additional professional services.

Advanced ClusterProven Service

The Advanced ClusterProven Service is designed to assist the ISV or in-house developer in bringing applications to a higher level of integration with iSeries clustering. Through professional consultation and collaboration, the developer's in-depth application knowledge and the Lakeview Technology consultant's iSeries clustering expertise are leveraged to develop upgraded application design models and processes that meet the more demanding requirements of IBM's Advanced Cluster Proven branding.

See 8.2.1, "The Advanced ClusterProven for iSeries Program" on page 164, for a discussion of the IBM Advanced ClusterProven program.

Application Clustering Audit

Over time, cluster-enabled applications can change. During an Application Clustering Audit, Lakeview Technology consultants work with the developers to ensure that, after redevelopment or upgrade to a new point release, applications remain fully compliant with current IBM clustering requirements.

Cluster Server Integration

Based on the market-proven MIMIX FourPack service, Cluster Server Integration covers all aspects of implementing the MIMIX suite of products in a clustered iSeries environment. This integration includes cluster and non-cluster enabled iSeries applications.

The full functionality of an HA environment is only partially addressed by the clustering architecture. Other non-application objects, environmental settings, and values must also be replicated or managed appropriately to achieve a complete, totally integrated, and optimized HA solution.

Clustering education

Different modes of education are available depending on the required level of clustering knowledge. Lakeview clustering courses range from a half-day overview seminar that reviews the IBM clustering initiatives, to a full five-day course that gives iSeries technical professionals an in-depth understanding of the details of clustering technology.

Note: IBM's education and service offerings are listed in Appendix C, "iSeries cluster resources" on page 291.

15

Vision Solutions Vision Suite

With OMS/400 Cluster Manager, recovery from unplanned failovers or planned switchovers can now be both seamless and rapid. Building upon the Vision Suite of middleware high availability software products, OMS/400 Cluster Manager extends the ability to create highly available and resilient data, application, and user environments.

> **Note:** Section 15.5, "Enhancements for OS/400 V5R1 functionality" on page 271, summarizes the latest enhancements to Vision Solutions (OMS 8.0), which includes support for the journaling and clustering features available in OS/400 V5R1. For the latest enhancements, and complete documentation on Vision Suite 8.0 and its clustering support, refer to the Vision Solutions Web Site at: `http://www.visionsolutions.com`
>
> Follow the links to **Vision Clustering Support**.

15.1 Vision Solutions OMS/400 Cluster Manager

In addition to non-clustering related product features, such as mirroring database files, data areas, and data queues in real-time using IBM journaling abilities, OMS/400 Cluster Manager provides object mirroring support for data Cluster Resource Groups (CRGs) and ClusterProven applications.

OMS/400 Cluster Manager provides these capabilities through new menus and screens presented to the user in two ways:

- The traditional OS/400 green-screen interface
- A client-server Java application with a graphical user interface running on a PC or workstation

Both of these interfaces are fully integrated with OMS/400. They allow the user to define sets of objects and, using bi-directional communication paths, to create and maintain additional sets of synchronized data.

15.1.1 Implementation goals

In addition to supporting clustered environments, the Vision Solutions objectives in implementing OMS/400 Cluster Manager include to:

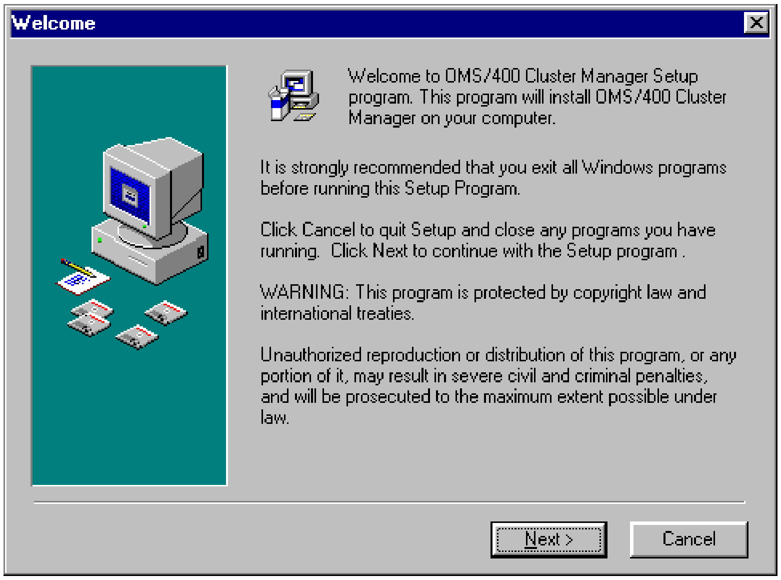
- Build upon the OMS/400 high level of data integrity to increase data resiliency on all clustered systems
- Work with ISVs to build highly resilient application environments
- Assist ISVs in the process of obtaining IBM ClusterProven status

15.2 Getting started with OMS/400 Cluster Manager

Before you install the client, set up the iSeries servers for clustering. Ensure that all managed systems are on the same operating system level and enabled for TCP/IP and clustering. OMS/400 R6.3 (or higher) must be installed.

15.2.1 Installing the client

In a Windows implementation of Vision Solutions OMS/400 Cluster Manager, there are five installation panels. The first panel is shown in Figure 15-1.



Welcome

Welcome to OMS/400 Cluster Manager Setup program. This program will install OMS/400 Cluster Manager on your computer.

It is strongly recommended that you exit all Windows programs before running this Setup Program.

Click Cancel to quit Setup and close any programs you have running. Click Next to continue with the Setup program .

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

Figure 15-1 OMS Welcome page

15.2.2 Starting the product

Upon first use of the GUI, OMS/400 Cluster Manager asks for the hostname of the cluster node to which to initially connect. Either the node's hostname or TCP/IP address of at least one cluster-enabled node is required to login and begin managing clusters.

15.2.3 Defining host systems

To determine which iSeries servers in a network are currently cluster-enabled, use the Client Server Configuration Wizard. Built into OMS/400 Cluster Manager, the Wizard automatically detects and reports on the cluster-enabled status and operating system level of all nodes reachable from the client computer.

When at least one cluster-enabled node is configured to OMS/400 Cluster Manager, management of the clustered environment can begin. To work with pre-defined clusters (for example, clusters built with the green-screen version of OMS/400), send a request for cluster information to any node in an existing cluster. Even if that request finds that the node is inactive, OMS/400 Cluster Manager attempts to forward that request to other nodes.

To configure new clusters using the GUI interface, log on to any iSeries server. Use OMS/400 Cluster Manager's built-in wizards to define clusters, CRGs, and recovery domains.

15.2.4 Auto-detecting clustered nodes

As requests are sent to various systems, those iSeries servers are automatically added to the list of configured host machines. OMS/400 Cluster Manager can then forward requests directly to the additional cluster nodes.

To illustrate this concept, refer to Figure 15-2. The client has only Node 1 currently defined in its list of host systems. When cluster information is requested from "Node 1", the response tells the client that there are actually three nodes in the cluster and stores in memory the additional nodes' host configuration information. When the client application is closed, that information is stored on the client computer for retrieval next time the application is started.

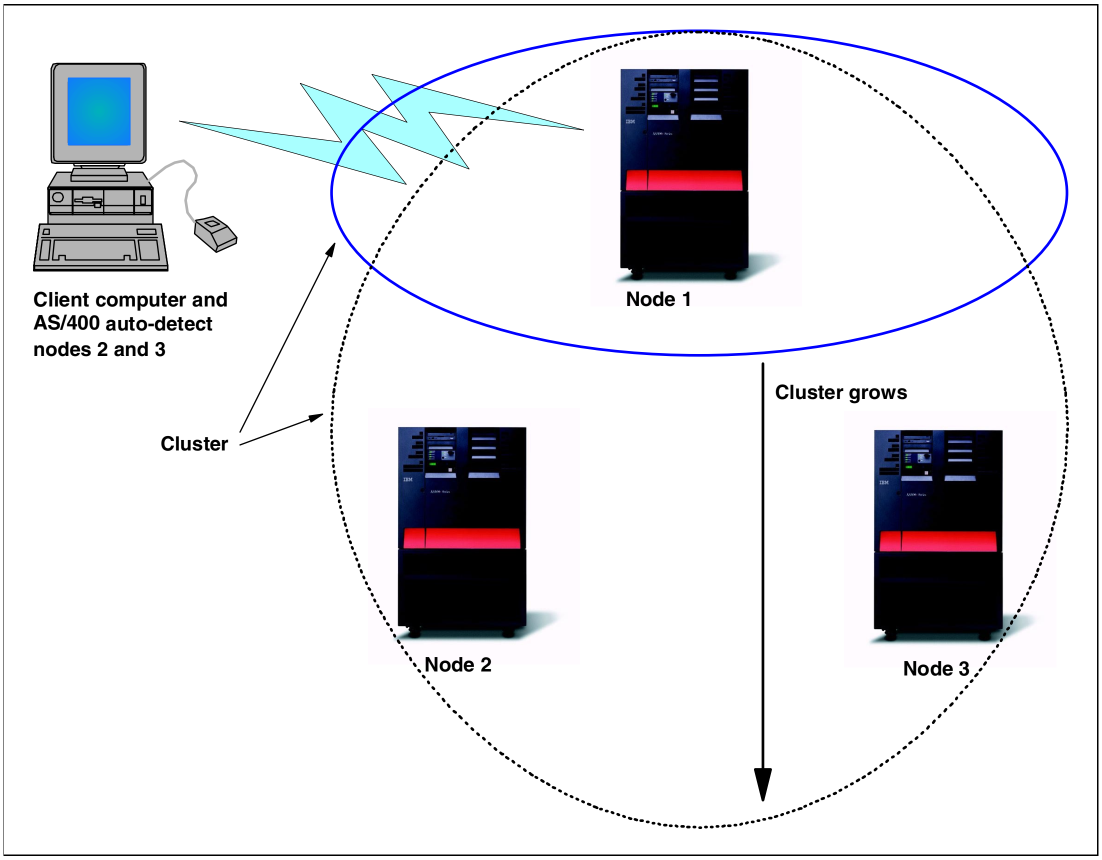


Figure 15-2 Auto-detecting nodes

Similarly, if new nodes have been added to an existing cluster since last using OMS/400 Cluster Manager, the client recognizes those hosts as “new” and adds them to the list of currently defined host systems.

This ability to auto-forward requests to any iSeries server reachable via TCP/IP allows organizations to rapidly configure and implement clustering environments.

15.2.5 IP interface selection

An additional feature allows the selection of an IP interface. When adding and configuring systems for clusters, you can view all IP interfaces that can interconnect to the nodes in a clustered environment. This feature allows organizations to define specific routing paths for cluster-enabled networks and reduce IP traffic on other, non-clustered networks.

15.2.6 Working with ClusterProven applications

As part of Vision Solutions’ continuing support for application integrity, OMS/400 Cluster Manager works with mixed data and application CRG environments for seamless switchovers. To prevent the loss of in-flight data transactions, OMS/400 Cluster Manager, working with ClusterProven applications, waits until the application completes its activities before notifying the data CRG that switchover or failover can commence.

The interaction between the application CRG and the data CRG varies, depending on the specific resiliency requirements of the application. For example, OMS/400 ensures the data associated with a ClusterProven application is in-sync. The term “in-sync” in this example means the recovery domain and switchover or failover information is the same (such as the current roles of the primary node and first and second backups are the same for both application and data CRGs).

If a commitment control scheme is used to increase application resilience, OMS/400 Cluster Manager only begins a data switchover when the following conditions are met:

- All users are removed and disconnected from the application, ensuring no more transactions are created.
- Transactions that are not committed are rolled back.
- No more transactions are coming into the journals.

Only then does the application CRG notify the data CRG that a data switchover can begin.

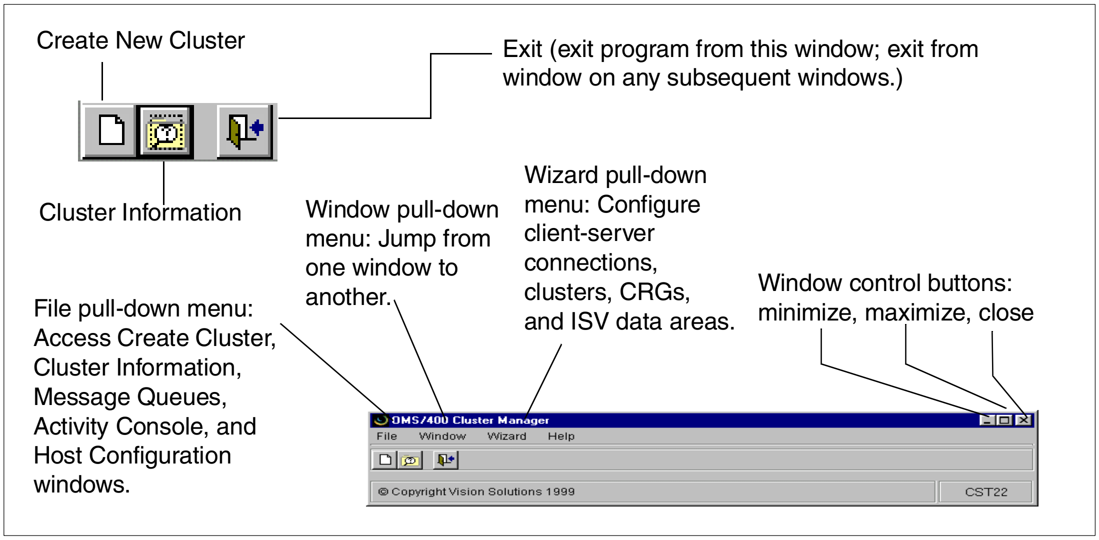
Similarly, when the switchover or failover is completed (assuming the former primary node is now a backup node, and the node that was the first backup is now the primary node), the application can restart. This allows users to log back into the application on the new primary system and begin working again. The data CRG then notifies OMS/400 to begin picking up new transactions and send them to the new backup system.

15.3 OMS/400 Cluster Manager sample displays

The following figures illustrate how to perform various clustering tasks using the GUI version of OMS/400 Cluster Manager.

15.3.1 Working with clusters and CRGs

Figure 15-3 shows the OMS Cluster Manager window and contains selection buttons to create a cluster and gather cluster information. From the File pull-down menu, a range of other cluster-related activities is available.

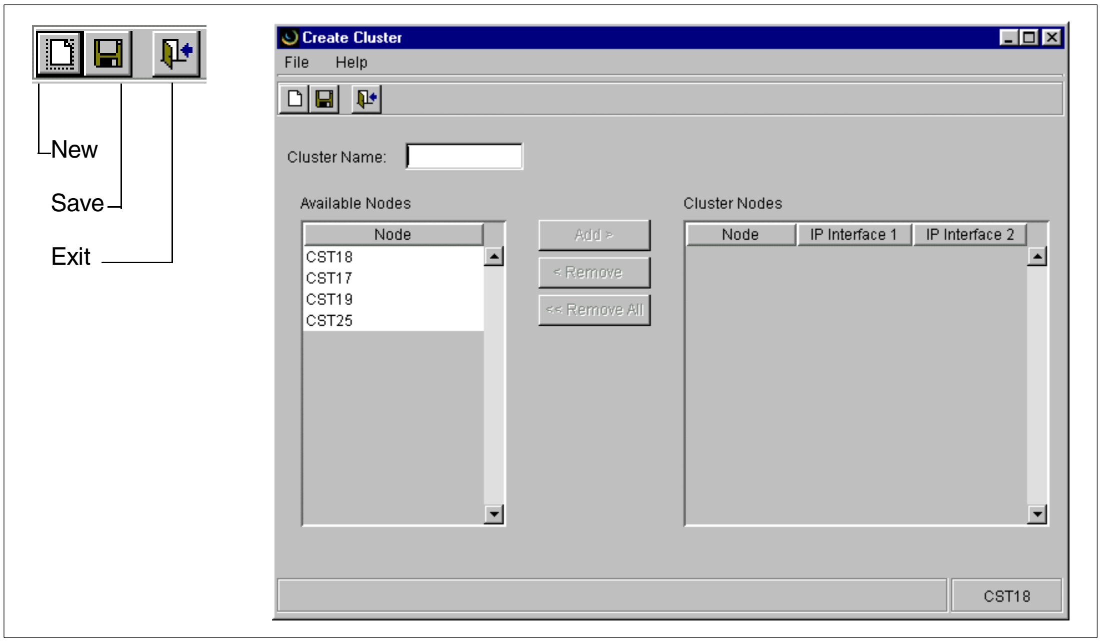


Create New Cluster
Exit (exit program from this window; exit from window on any subsequent windows.)
Cluster Information
Window pull-down menu: Jump from one window to another.
Wizard pull-down menu: Configure client-server connections, clusters, CRGs, and ISV data areas.
Window control buttons: minimize, maximize, close
File pull-down menu: Access Create Cluster, Cluster Information, Message Queues, Activity Console, and Host Configuration windows.
OMS/400 Cluster Manager
File
Window
Wizard
Help
© Copyright Vision Solutions 1999
CST22

Figure 15-3 OMS Cluster Manager

15.3.2 Creating new clusters

When creating a new cluster, a window appears as shown in Figure 15-4.



Cluster Name:

Available Nodes

Node
CST18
CST17
CST19
CST25

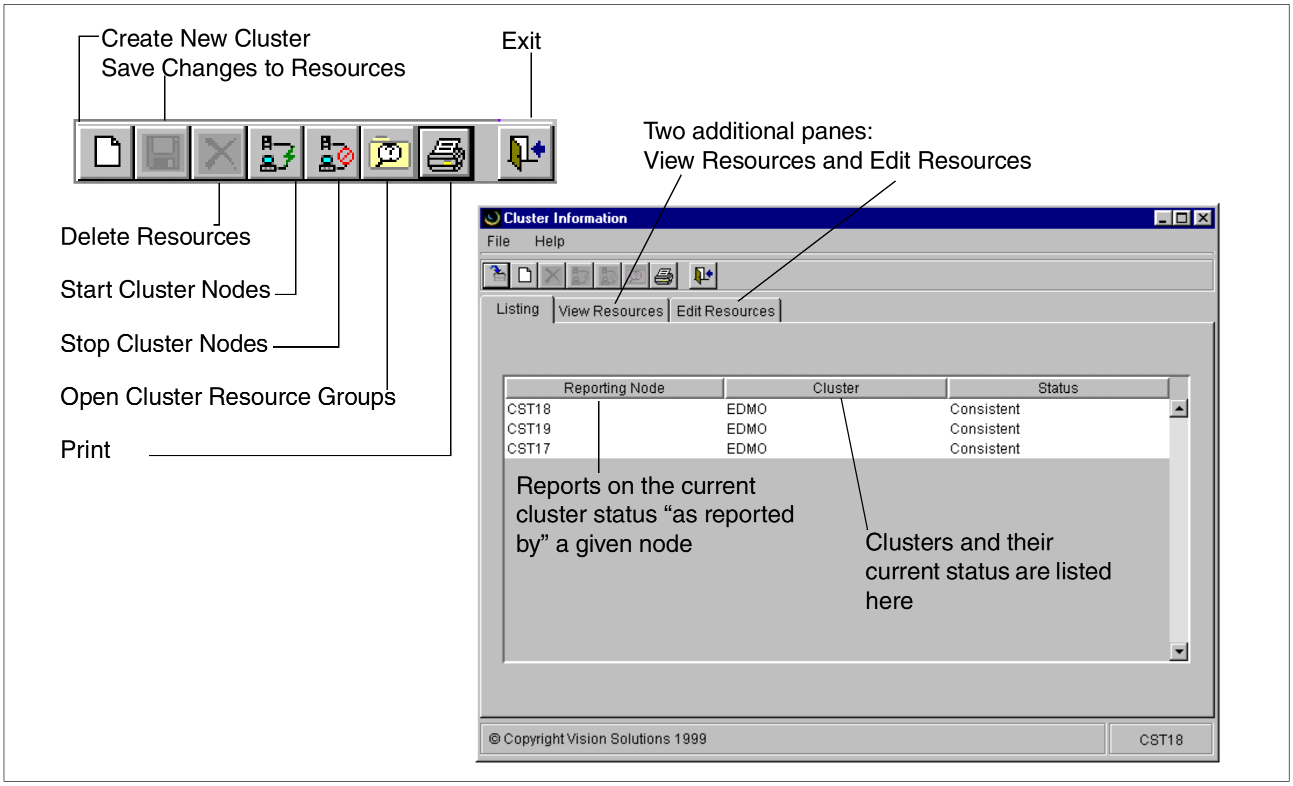
Cluster Nodes

Node	IP Interface 1	IP Interface 2

Figure 15-4 Creating a cluster window

15.3.3 Viewing cluster information

Once the cluster is created, cluster information and resources can be viewed from the cluster information window. Two additional panels are available: View Resources and Edit Resources. See Figure 15-5 for an illustration.



Create New Cluster

Save Changes to Resources

Exit

Delete Resources

Start Cluster Nodes

Stop Cluster Nodes

Open Cluster Resource Groups

Print

Two additional panes:
View Resources and Edit Resources

Reporting Node	Cluster	Status
CST18	EDMO	Consistent
CST19	EDMO	Consistent
CST17	EDMO	Consistent

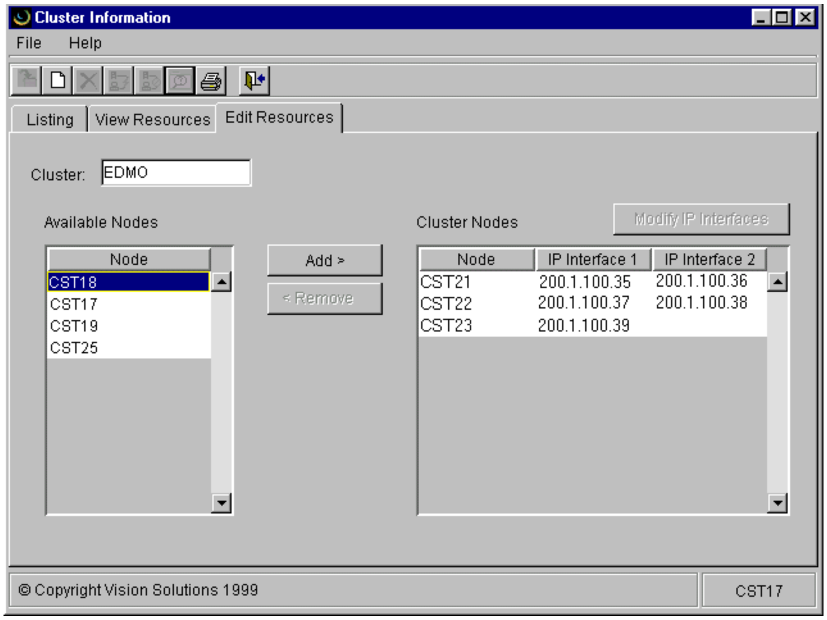
Reports on the current cluster status “as reported by” a given node

Clusters and their current status are listed here

Figure 15-5 Cluster information windows

15.3.4 Adding a node to the cluster

Add a node to the cluster by selecting the node from a OMS Cluster Manager standard window list. See Figure 15-6 for an illustration.



Cluster: EDMO

Available Nodes

Node
CST18
CST17
CST19
CST25

Cluster Nodes

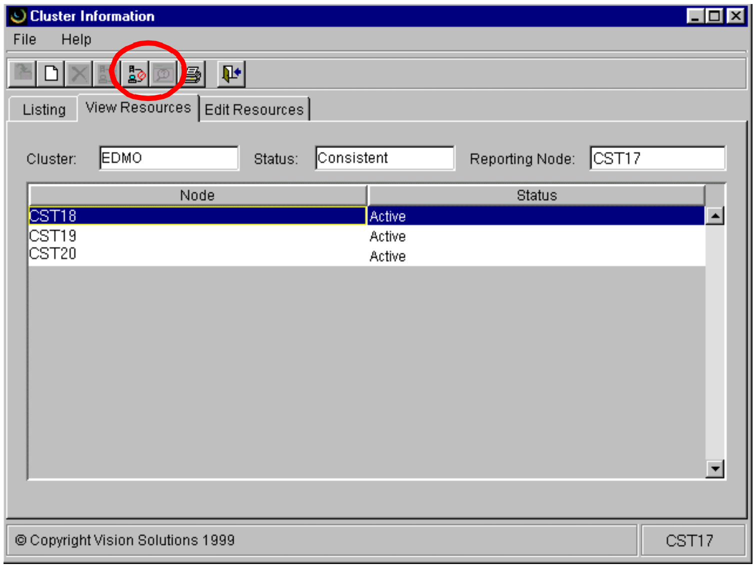
Node	IP Interface 1	IP Interface 2
CST21	200.1.100.35	200.1.100.36
CST22	200.1.100.37	200.1.100.38
CST23	200.1.100.39	

Figure 15-6 Adding a node

15.3.5 Activating and de-activating nodes in the cluster

Once the initial setup tasks are completed, the cluster can be activated. When it is running, it can be de-activated. The nodes displayed can be selected and processed by clicking the End or Start buttons.

Select a node and then click the **End** or **Start Cluster Node** icon (circled in Figure 15-7) to deactivate or activate the node.



Cluster: EDMO Status: Consistent Reporting Node: CST17

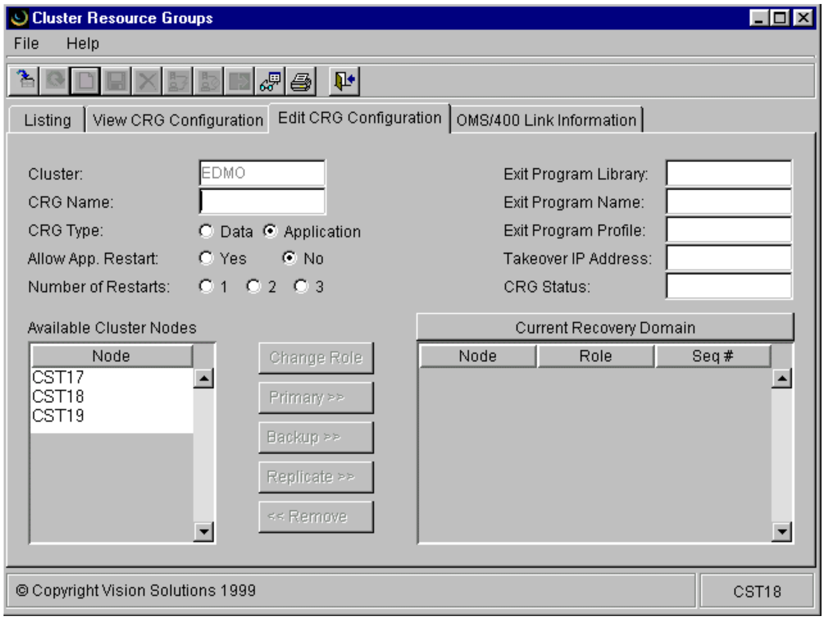
Node	Status
CST18	Active
CST19	Active
CST20	Active

© Copyright Vision Solutions 1999 CST17

Figure 15-7 Cluster activation

15.3.6 Creating and using Cluster Resource Groups

To create resilient objects in the Cluster Resource Groups window, select the **Edit CRG Configuration** tab (Figure 15-8). This panel allows the creation of both data and application CRGs. Depending on the type of CRG, not all input boxes are required.



Cluster: EDMO

CRG Name:

CRG Type: ○ Data ● Application

Allow App. Restart: ○ Yes ● No

Number of Restarts: ○ 1 ○ 2 ○ 3

Exit Program Library:

Exit Program Name:

Exit Program Profile:

Takeover IP Address:

CRG Status:

Available Cluster Nodes

Node
CST17
CST18
CST19

Current Recovery Domain

Node	Role	Seq #

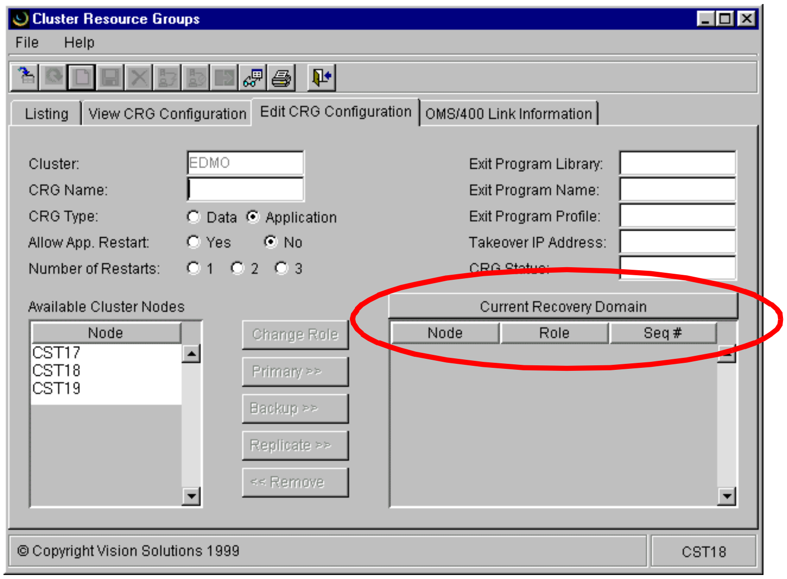
© Copyright Vision Solutions 1999 CST18

Figure 15-8 Creating CRGs with iCluster

15.3.7 Changing a CRG recovery domain

Changing the recovery domain and altering the role of a node is one of the tasks performed by the operations group when managing the cluster.

Toggle between the current and preferred recovery domain roles. See Figure 15-9 for an illustration.



Cluster: EDMO
CRG Name:
CRG Type: ○ Data ● Application
Allow App. Restart: ○ Yes ● No
Number of Restarts: ○ 1 ○ 2 ○ 3

Exit Program Library:
Exit Program Name:
Exit Program Profile:
Takeover IP Address:
CRG Status:

Available Cluster Nodes

Node
CST17
CST18
CST19

Current Recovery Domain

Node	Role	Seq #

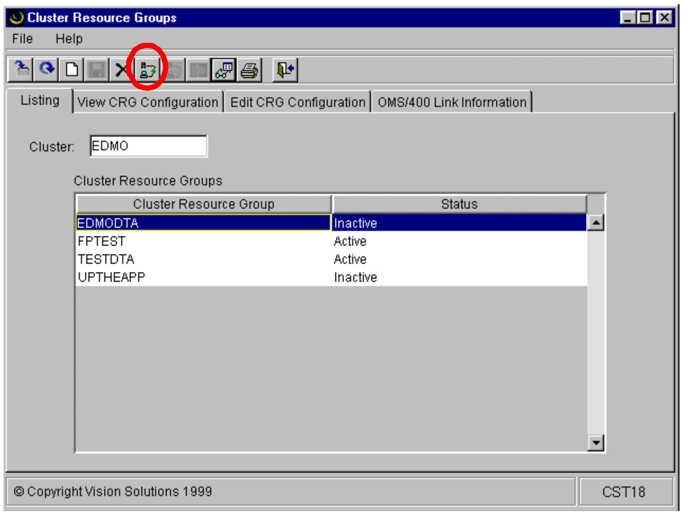
© Copyright Vision Solutions 1999   CST18

Figure 15-9 Changing a recovery domain

15.3.8 Activating or starting a data or application CRG

Once an application or data CRGs is created, it can be selected for activation.

Select an inactive CRG, and click the **Start Resource Group Services** button (circled in Figure 15-10).



Cluster: EDMO

Cluster Resource Groups

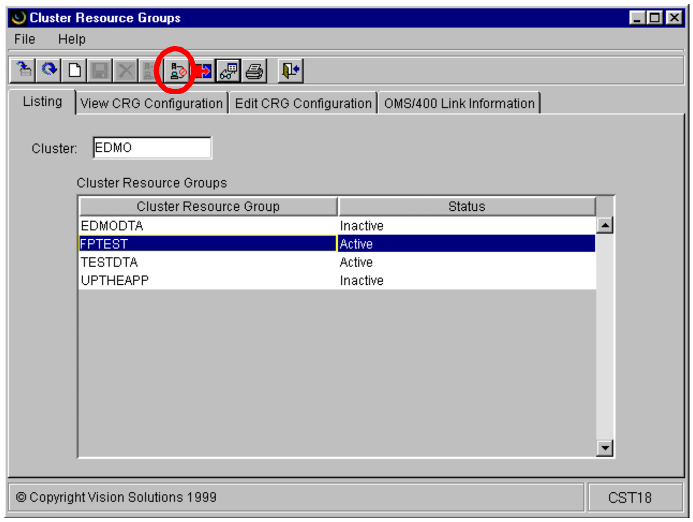
Cluster Resource Group	Status
EDMODTA	Inactive
FPTEST	Active
TESTDTA	Active
UPTHEAPP	Inactive

© Copyright Vision Solutions 1999   CST18

Figure 15-10 Activating CRG

15.3.9 De-activating or ending a data or application CRG

To end a data or application CRG, first highlight the active CRG. Then click the **Stop Resource Group Services** button (circled in Figure 15-11).



Cluster: EDMO

Cluster Resource Groups

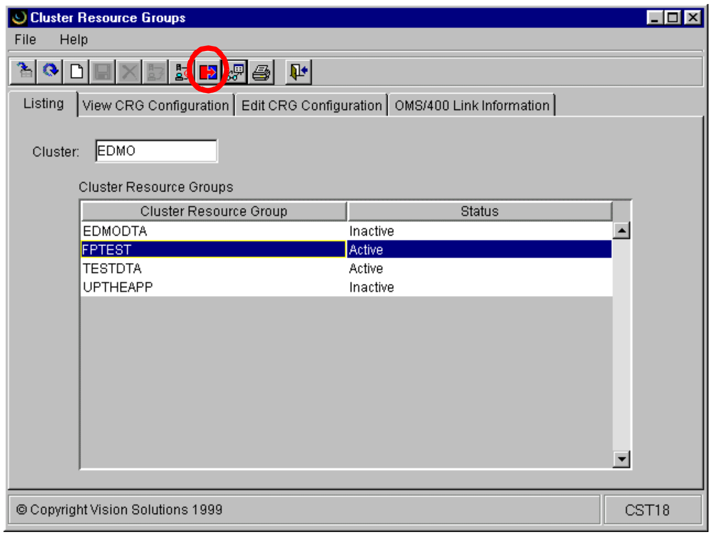
Cluster Resource Group	Status
EDMODTA	Inactive
FPTEST	Active
TESTDTA	Active
UPTHEAPP	Inactive

© Copyright Vision Solutions 1999 CST18

Figure 15-11 Stopping Resource Group Services

The Cluster Resource Groups Listing view can be used to perform a data switchover or application CRG.

Select an active CRG, and click the **Initiate Switchover** icon (circled in Figure 15-12).



Cluster: EDMO

Cluster Resource Groups

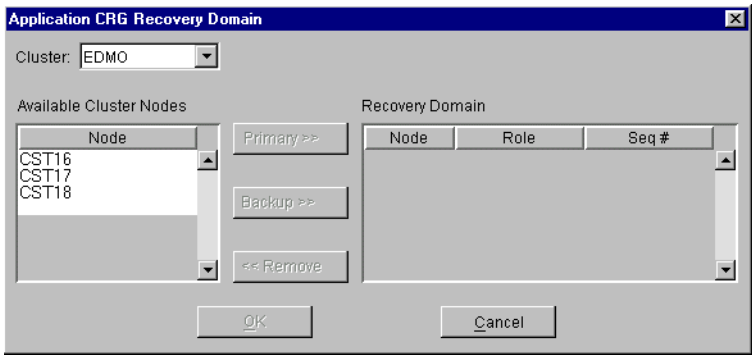
Cluster Resource Group	Status
EDMODTA	Inactive
FPTEST	Active
TESTDTA	Active
UPTHEAPP	Inactive

© Copyright Vision Solutions 1999 CST18

Figure 15-12 Switching over CRG

15.3.10 Creating an application CRG recovery domain

When creating an application CRG recovery domain, you must specify a takeover IP address See Figure 15-13 for an illustration.



Cluster: EDMO

Available Cluster Nodes

Node
CST16
CST17
CST18

Recovery Domain

Node	Role	Seq #

Figure 15-13 Creating an application CRG

The takeover IP address must not be active on any of the nodes. See Figure 15-14.

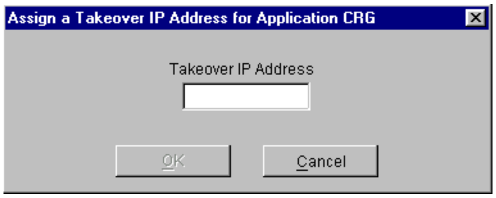
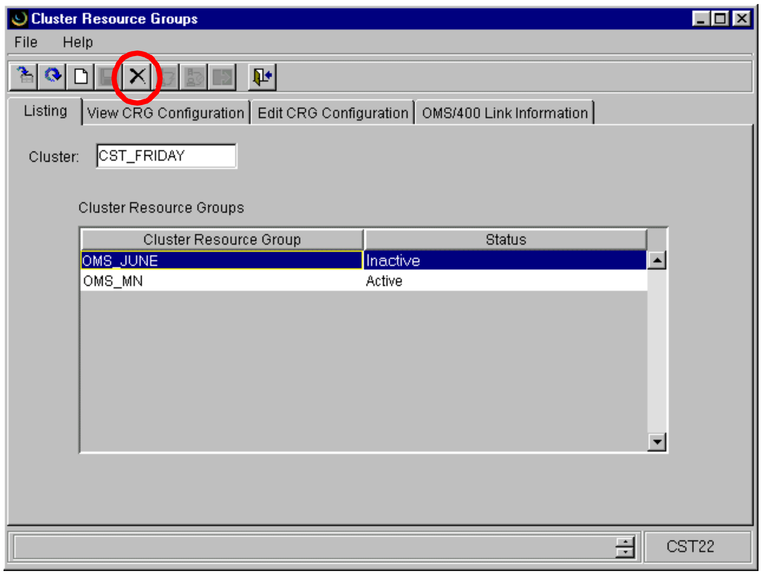


Figure 15-14 Takeover IP address

15.3.11 Removing a data or application CRG

Removing a data or application CRG is a standard management function.

Select an inactive CRG and click the **Delete Cluster Resource Group** icon. Figure 15-15 shows an example of this function.



Cluster: CST_FRIDAY

Cluster Resource Groups

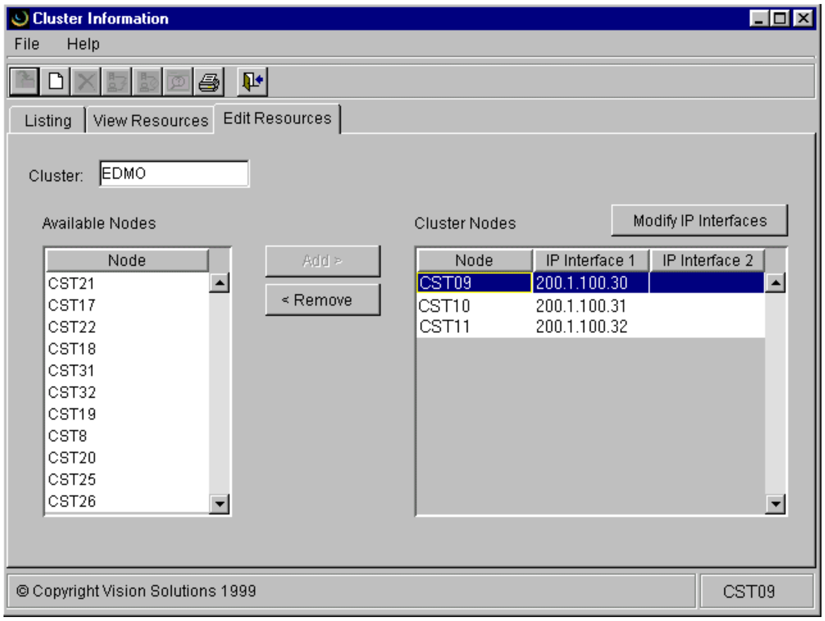
Cluster Resource Group	Status
OMS_JUNE	Inactive
OMS_MN	Active

CST22

Figure 15-15 Removing CRGs

15.3.12 Removing a node from the cluster

Back at the Cluster information window, select the **Edit Resources** tab (Figure 15-16). The panel that appears enables a node to be selected and removed from the cluster.



Cluster: EDMO

Node
CST21
CST17
CST22
CST18
CST31
CST32
CST19
CST8
CST20
CST25
CST26

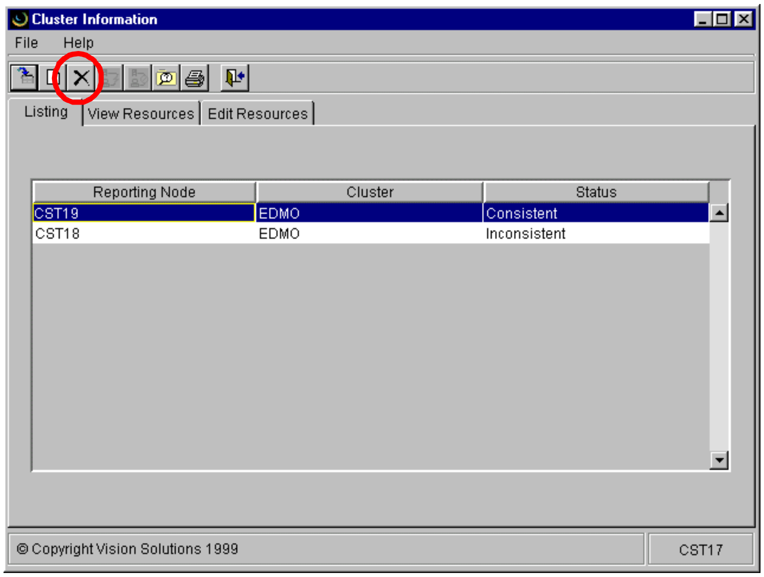
Node	IP Interface 1	IP Interface 2
CST09	200.1.100.30	
CST10	200.1.100.31	
CST11	200.1.100.32	

Figure 15-16 Removing a node from a cluster

15.3.13 Removing the entire cluster

In certain cases, the entire cluster must be removed.

At the Cluster Information window, select the **Listing** tab. In the view, select the cluster and then click the **Delete** cluster button (see Figure 15-17).



Reporting Node	Cluster	Status
CST19	EDMO	Consistent
CST18	EDMO	Inconsistent

Figure 15-17 Removing the cluster

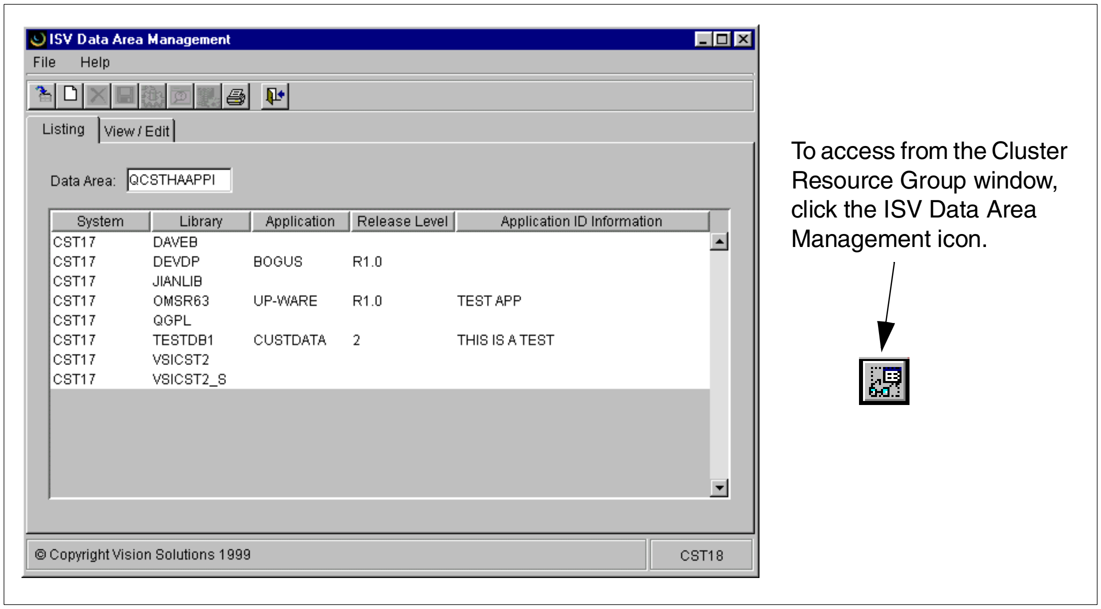
15.4 Working with applications

If the application is to be cluster aware, edit the ISV data area QCSTHAPPI.

QCSTHAPPI will be changed if the application is developed by an ISV. If the application is developed in-house, change this data area to make the application cluster aware.

15.4.1 ISV data area contents

The QCSTHAPPI data area can be modified from the ISV Data Area Management window. To access from the Cluster Resource Group window, select the **ISV Data Area Management** icon (Figure 15-18).



Data Area: QCSTHAAPPI

System	Library	Application	Release Level	Application ID Information
CST17	DAVEB			
CST17	DEVDP	BOGUS	R1.0	
CST17	JIANLIB			
CST17	OMSR63	UP-WARE	R1.0	TEST APP
CST17	QGPL			
CST17	TESTDB1	CUSTDATA	2	THIS IS A TEST
CST17	VSICST2			
CST17	VSICST2_S			

© Copyright Vision Solutions 1999

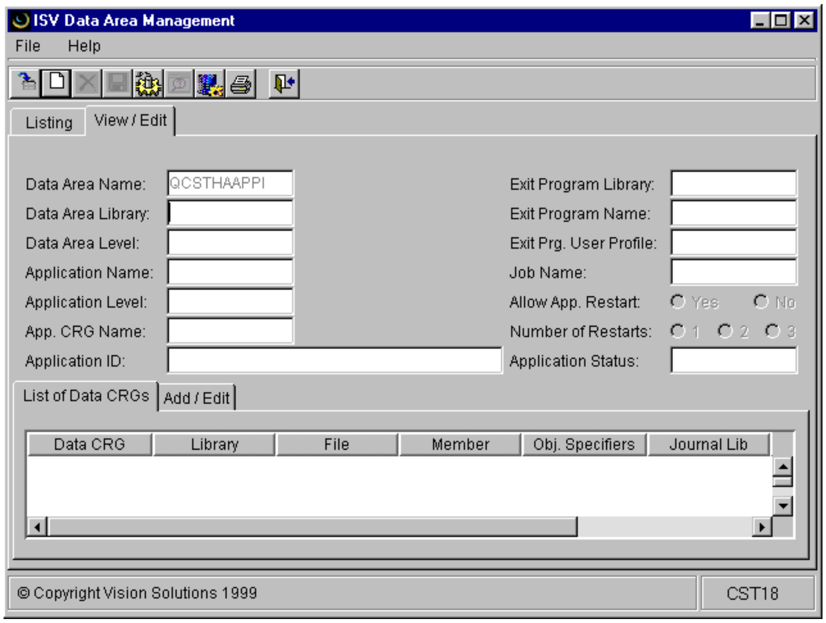
CST18

To access from the Cluster Resource Group window, click the ISV Data Area Management icon.

Figure 15-18 QCSTHAPPI contents

15.4.2 Creating ISV data areas for application CRGs

When creating ISV data areas for application CRGs, select the **View/Edit** tab (Figure 15-19) for the data area input fields.



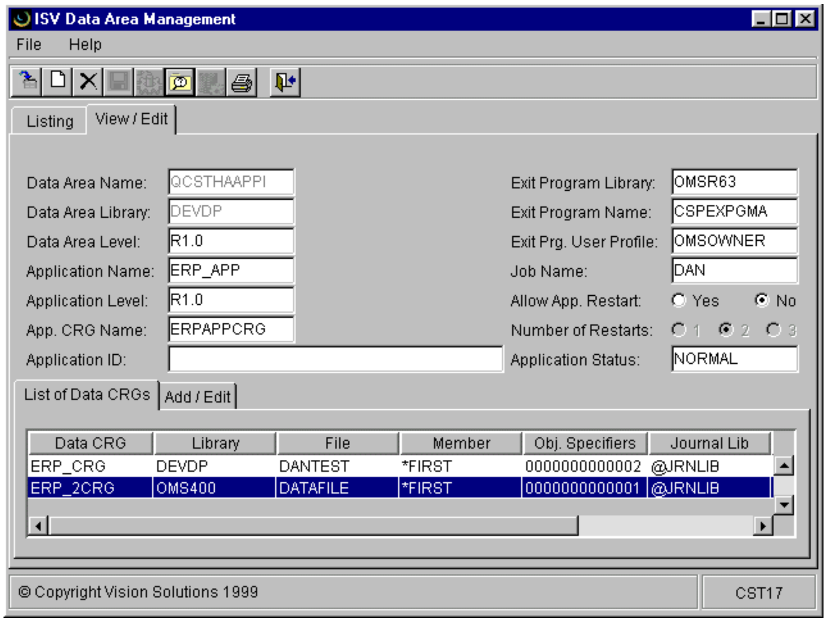
Data Area Name: QCSTHAAPPI
Data Area Library:
Data Area Level:
Application Name:
Application Level:
App. CRG Name:
Application ID:
Exit Program Library:
Exit Program Name:
Exit Prg. User Profile:
Job Name:
Allow App. Restart: ○ Yes ○ No
Number of Restarts: ○ 1 ○ 2 ○ 3
Application Status:

Data CRG	Library	File	Member	Obj. Specifiers	Journal Lib

Figure 15-19 Creating QCSTHAPPI

15.4.3 Changing or updating data areas

To change or update a data area, select the **View/Edit** tab from the ISV Data Area Management window. Then, select the CRG to be changed in the List of Data CRGs panel (Figure 15-20).



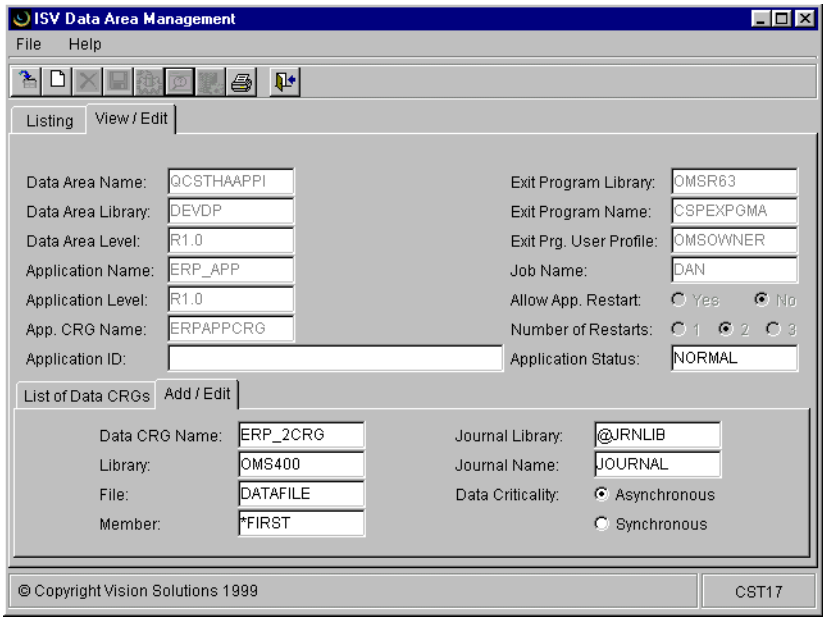
Data Area Name: QCSTHAAPPI
Data Area Library: DEVDP
Data Area Level: R1.0
Application Name: ERP_APP
Application Level: R1.0
App. CRG Name: ERPAPPCRG
Application ID:
Exit Program Library: OMSR63
Exit Program Name: CSPEXPGMA
Exit Prg. User Profile: OMSOWNER
Job Name: DAN
Allow App. Restart: ○ Yes ● No
Number of Restarts: ○ 1 ● 2 ○ 3
Application Status: NORMAL

Data CRG	Library	File	Member	Obj. Specifiers	Journal Lib
ERP_CRG	DEVDP	DANTEST	*FIRST	0000000000002	@JRNLIB
ERP_2CRG	OMS400	DATAFILE	*FIRST	0000000000001	@JRNLIB

Figure 15-20 Changing QCSTHAPPI

15.4.4 Changing a resilient application's data area contents

The data area contents are displayed and available for update in the Add/Edit panel (Figure 15-21).



Data Area Name:	QCSTHAAPPI	Exit Program Library:	OMSR63
Data Area Library:	DEVDP	Exit Program Name:	CSPEXPGMA
Data Area Level:	R1.0	Exit Prg. User Profile:	OMSOWNER
Application Name:	ERP_APP	Job Name:	DAN
Application Level:	R1.0	Allow App. Restart:	○ Yes ● No
App. CRG Name:	ERPAPPCRG	Number of Restarts:	○ 1 ● 2 ○ 3
Application ID:		Application Status:	NORMAL

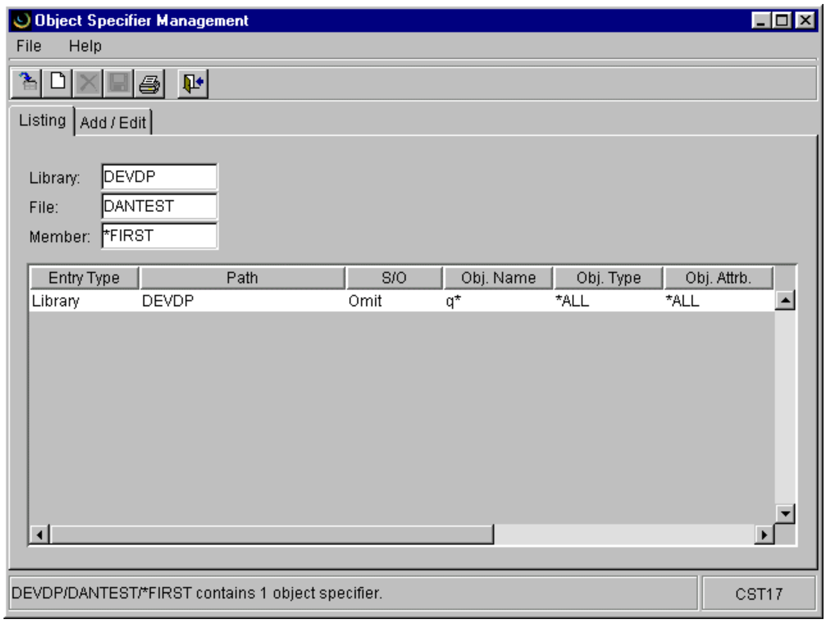
Data CRG Name:	ERP_2CRG	Journal Library:	@JRNLIB
Library:	OMS400	Journal Name:	JOURNAL
File:	DATAFILE	Data Criticality:	● Asynchronous
Member:	*FIRST		○ Synchronous

© Copyright Vision Solutions 1999 CST17

Figure 15-21 Updating QCSTHAPPI contents

15.4.5 Working with object specifiers

Object specifiers are the files that contain the resilient information associated with a particular application CRG. The Object Specifier Management window, shown in Figure 15-22, allows the management of these object specifier files.



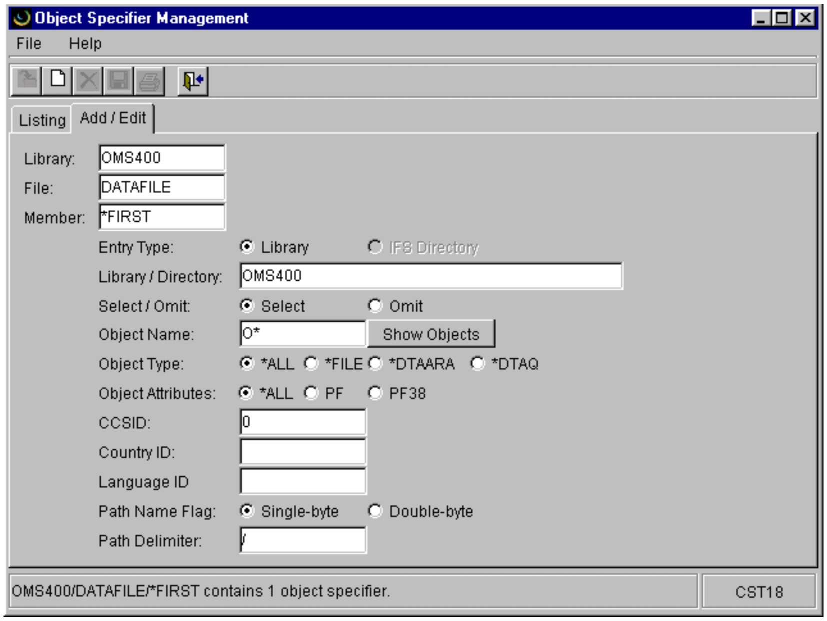
Library:	DEVDP
File:	DANTEST
Member:	*FIRST

Entry Type	Path	S/O	Obj. Name	Obj. Type	Obj. Attrb.
Library	DEVDP	Omit	q*	*ALL	*ALL

DEVDP/DANTEST/*FIRST contains 1 object specifier. CST17

Figure 15-22 Object specifier list

To work with object specifiers, select the **Add/Edit** tab. The object specifier details are then displayed (Figure 15-23).



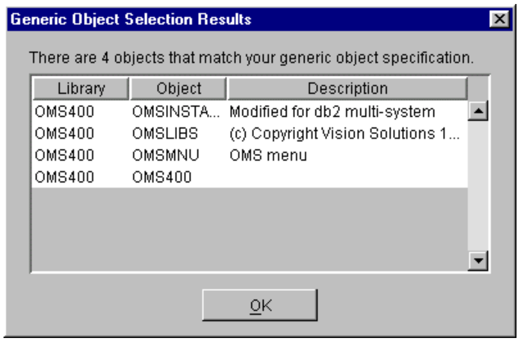
Library: OMS400
File: DATAFILE
Member: *FIRST
Entry Type: ● Library ○ IFS Directory
Library / Directory: OMS400
Select / Omit: ● Select ○ Omit
Object Name: O*
Object Type: ● *ALL ○ *FILE ○ *DTAARA ○ *DTAQ
Object Attributes: ● *ALL ○ PF ○ PF38
CCSID: 0
Country ID:
Language ID
Path Name Flag: ● Single-byte ○ Double-byte
Path Delimiter: /

OMS400/DATAFILE/*FIRST contains 1 object specifier.
CST18

Figure 15-23 Working with object specifiers

15.4.6 Object selection results

The Object Selection Results panel (Figure 15-24) displays objects that are found within the library or directory that is selected.



There are 4 objects that match your generic object specification.

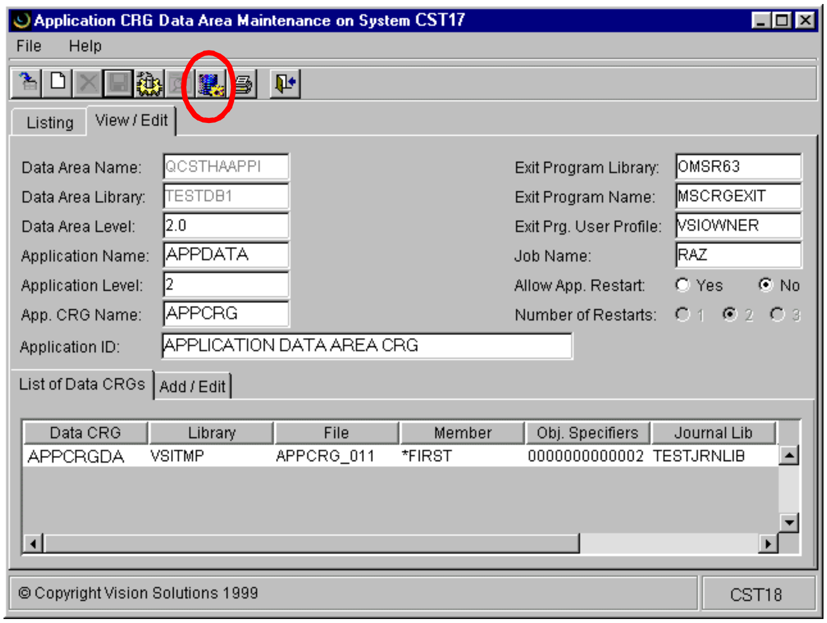
Library	Object	Description
OMS400	OMSINSTA...	Modified for db2 multi-system
OMS400	OMSLIBS	(c) Copyright Vision Solutions 1...
OMS400	OMSMNU	OMS menu
OMS400	OMS400	

Figure 15-24 Object selection

15.4.7 Creating a list of objects for high availability

The wizard for ISV Data Management enables easy selection of objects for resiliency.

From the Wizard for ISV Data Area Management, select a data area, and then click the **View/Edit** tab. Click the **Generate OMS/400 Object List** icon (circled in Figure 15-25).



Data Area Name:	QCSTHAAPPI
Data Area Library:	TESTDB1
Data Area Level:	2.0
Application Name:	APPDATA
Application Level:	2
App. CRG Name:	APPCRG
Application ID:	APPLICATION DATA AREA CRG
Exit Program Library:	OMSR63
Exit Program Name:	MSCRGEXIT
Exit Prg. User Profile:	VSIOWNER
Job Name:	RAZ
Allow App. Restart:	○ Yes ● No
Number of Restarts:	○ 1 ● 2 ○ 3

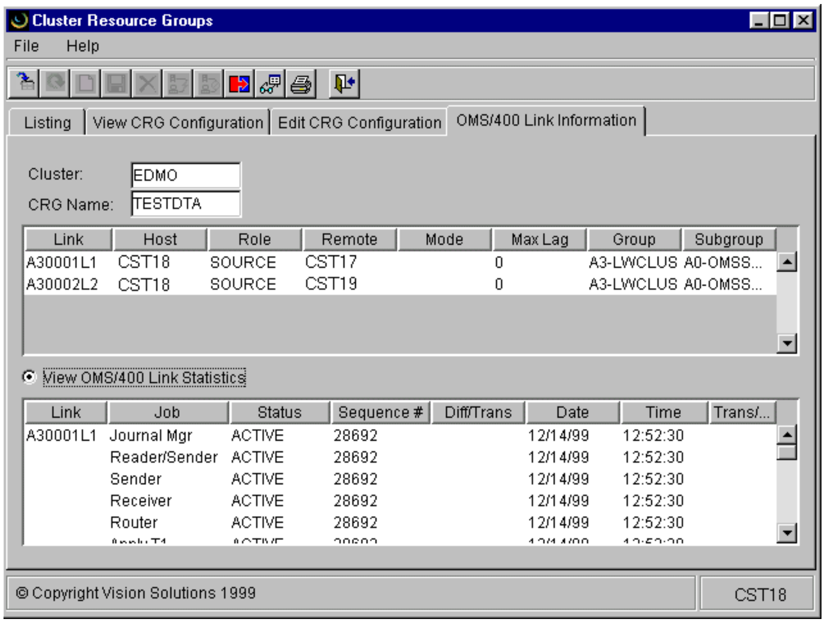
Data CRG	Library	File	Member	Obj. Specifiers	Journal Lib
APPCRGDA	VSITMP	APPCRG_011	*FIRST	0000000000002	TESTJRNLIB

© Copyright Vision Solutions 1999 CST18

Figure 15-25 Creating a resilient object list

15.4.8 Viewing OMS/400 links and statistics

From the Cluster Resource Groups main window, select the **OMS/400 Link Information** tab. On this panel (Figure 15-26), the resilient resources status is displayed.



Cluster: EDMO

CRG Name: TESTDTA

Link	Host	Role	Remote	Mode	Max Lag	Group	Subgroup
A30001L1	CST18	SOURCE	CST17		0	A3-LWCLUS	A0-OMSS...
A30002L2	CST18	SOURCE	CST19		0	A3-LWCLUS	A0-OMSS...

● View OMS/400 Link Statistics

Link	Job	Status	Sequence #	Diff/Trans	Date	Time	Trans/...
A30001L1	Journal Mgr	ACTIVE	28692		12/14/99	12:52:30	
	Reader/Sender	ACTIVE	28692		12/14/99	12:52:30	
	Sender	ACTIVE	28692		12/14/99	12:52:30	
	Receiver	ACTIVE	28692		12/14/99	12:52:30	
	Router	ACTIVE	28692		12/14/99	12:52:30	

© Copyright Vision Solutions 1999 CST18

Figure 15-26 Viewing OMS/400 links and statistics

15.5 Enhancements for OS/400 V5R1 functionality

Vision Suite supports the clustering functions available with OS/400 V5R1. This section provides an overview of the enhanced OMS/400 features:

- **Clustering technology**

  Level 2 cluster services provided for in OS/400 V5R1 are supported and managed through OMS/400. This includes the *device CRG,* which adds resilient hardware to the clustering environment.

- **Switched disk configuration**

  OMS/400 supports the creation of an independent ASP, defined as a device CRG to allow switching of Integrated File System objects. Switchable towers (for switchable standalone hardware) and switchable DASD IOP (for switching between LPAR partitions) are supported.

- **Vision Command Center Management**

  A Java-based Command Center (VCC) is offered. VCC represents a new fully Java-based architecture that extends the graphical central point management capabilities offered in earlier Vision solutions.

  > **Note:** The screens in this chapter document the Vision Clustering and Java-based Cluster Management Support available prior to OS/400 V5R1.

- **Cluster Tuning**

  Clustering Version 2 (V5R1 and higher) includes a performance tuning feature to monitor and change the cluster services heartbeat and messaging characteristics to better match a network and response time requirements.

- **Data area and data queue journaling**

  Clustering Version 2 supports the journaling enhancements available in V5R1 for data areas and data queues. OMS/400 now supports sending data queues and restoring the queue contents.

- **Large Object (LOB)**

  OMS/400 fully supports the configuration and mirroring of large object fields in database files.

- **Minimized journal entry data**

  OMS/400 fully supports a mirroring of the V5R1 minimized entry specific journal data feature.

- **Byte stream file journaling**

  OMS/400 includes new features to support the replication of changes, creations, and deletions of files stored in the Integrated File System (IFS) on V5R1 systems. While IFS byte stream file replication is enabled in ODS/400, a change to an IFS object requires that the entire object be replicated to the target system. With the V5R1 support of byte stream journaling, only the changed bytes are replicated rather than the entire object.

- **Held Object Analysis**

  Held Object Analysis (HOA) is a utility application that can help troubleshoot the cause of an object on hold status.

- **Support for APYJRNCHGX**

  OMS/400 supports elements of the extended journaling features of this option.

For full information on the latest clustering support from Vision Solutions, refer to the Vision Solutions Web site at: http://www.visionsolutions.com

Part 4

Appendices

Part 4 includes the appendices that complement the material presented in this redbook. The appendices contain information on the iSeries software and hardware functions that are not unique to cluster support, but vital for a highly available solution.

A

Advanced clusters explained

For the more advanced (or more curious) reader, the technologies and architectural features available to support clustering from OS/400 V4R4 onward are described in the following sections. This appendix provides more detailed information than Chapter 4, “iSeries clusters explained” on page 31.

A.1 Underlying technologies

This section describes the design and recovery of iSeries clusters.

A.2 Peer cluster node design

Many cluster implementations follow the paradigm of having a *leader* for various clustering protocols. A leader may be established as a result of configuration (for example, the primary node is the leader), or it may be determined through an internal algorithm (for example, based on an IP address).

iSeries cluster architecture is leaderless architecture. It involves a peer relationship among the cluster nodes. Each active node has all of the information needed to understand the total configuration and operational characteristics of the cluster. As such, a request for a cluster action can be initiated from any active node in the cluster. Furthermore, any node (not necessarily the requesting node) can assume the role as the coordinator for a particular protocol.

The iSeries peer design helps to ensure that a single outage, or even an outage of several cluster nodes, seldom constitutes a cluster failure.

A.2.1 Heartbeat and cluster communication

Heartbeat monitoring determines whether each cluster node is active. When the heartbeat processing for a cluster node fails, the condition is reported, and the resilient resources can automatically fail over to a backup node.

A heartbeat failure is more complex than just one missed signal. A heartbeat message is sent every three seconds from every node in the cluster to its upstream neighbor. In return, each node expects an acknowledgment of the heartbeat signal. In effect, this presents a two-way liveness mechanism. If a node fails or a break occurs in the network, or when a heartbeat or its acknowledgment is not received, a failure is not immediately reported. Heartbeating continues every three seconds to try to re-establish communications.

If a node misses four consecutive heartbeats, a heartbeat failure is signaled. After this failure is confirmed, the failover process causes access to the cluster resources to be switched over to the designated first backup node. Using a low-level message function, the heartbeat service within cluster topology services ensures low system overhead during normal operations. Heartbeating on remote subnets may be four times the overhead of local heartbeating.

Other components of Cluster Resource Services (CRS) rely on cluster topology services to determine when a node is unreachable. In some circumstances, heartbeat failure does not translate into a node failure, in which case a failover may not occur.

If the cluster consists of multiple physical networks, the heartbeat process is more complex. Routers and relay nodes are used to tie the physical networks together as though it were one logical network.

A router can be another iSeries server or a router box that directs communication to another router. Every local network is assigned a relay node. The relay node is determined to be the cluster node with the lowest node ID in the network.

For example, if two networks are involved, a logical network containing the two relay nodes is created. The relay nodes can then send heartbeats to each other. By using routers and relay nodes, the cluster nodes in these two networks can monitor each other and signal any node failures. See Figure A-1 for an illustration.

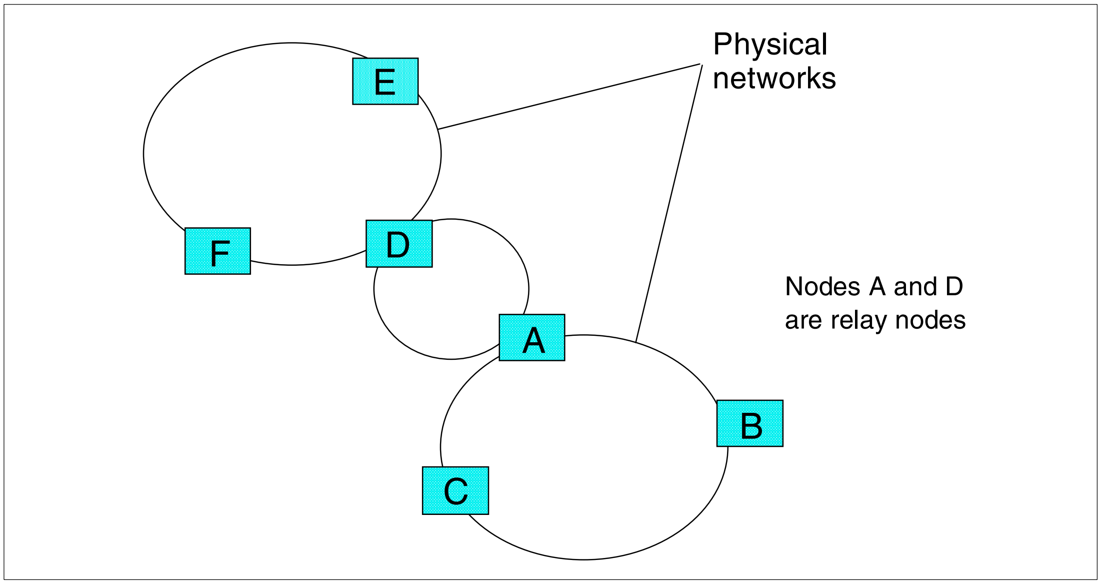


Figure A-1 Relay nodes in heartbeat protocols

iSeries Cluster Resource Services makes no assumption about throughput, latency, topology, or stability of the network. The heartbeat algorithms are expected to work over any supported network configurations.

A.2.2 Distributed activities

Most cluster actions are distributed activities resulting from a user request or system detected event. The synchronization of actions across the nodes of a cluster, or across a subset of the nodes, is accomplished through a distributed activity.

All of the cluster nodes affected by the action need to be involved to ensure that the results are consistently reflected across the cluster. The cluster engine and cluster communications provide the underlying services for building what is referred to as *distributed activity groups*. The Cluster Engine’s Group membership services are used by cluster control and the Cluster Resource Group manager to defined distributed activity groups.

For cluster control, a distributed activity group is used for the distributed activities associated with the definition and administration of the cluster. Each node in the cluster is a member in this distributed activity group. There are multiple distributed activity groups associated with the Cluster Resource Group manager. One set, called the *Cluster Resource Group manager distributed activity group*, is a distributed activity group. It is defined across the entire cluster and is used to handle the creation of new CRGs on each cluster node in the recovery domain and other similar global activities. A distributed activity group is defined for each CRG to handle processing specific to that CRG.

Using distributed activities, cluster control and Cluster Resource Group manager can synchronize their services across all affected nodes within the cluster. Any change to internal information or external cluster objects on one cluster node is simultaneously reflected on all nodes in the cluster. Complex protocol flows may be needed to accomplish this processing or to back out changes in the event that an error condition is detected. There are no assumptions made regarding the guaranteed low latency for the services of the underlying network. The reliance is solely on asynchronous distributed agreement solutions.

A.2.3 Job structure for Cluster Resource Services

The use of the cluster engine's group services is apparent by looking at the Cluster Resource Services (CRS) job structure. When a cluster is started on a cluster node, a set of system services is started. Each of these services is designed to be highly available (resilient to errors).

These services are represented by multi-threaded jobs running in the QSYSWRK subsystem. Anytime a cluster node is active, the following jobs are active in that subsystem:

- A cluster control job called QCSTCTL
- A Cluster Resource Group manager job called QCSTCRGM
- Additional jobs are started for handling the Cluster Resource Groups. One job exists for each CRG defined in the cluster. The job name is the same as the CRG name.

Figure A-2 shows an example job structure with just one CRG defined (CRG a). The figure also shows the related jobs, including:

- User jobs to initiate the cluster request (normally in the subsystem for the cluster management processing)
- Exit program job that is called to handle CRG specific processing
- Application subsystem for a highly available application

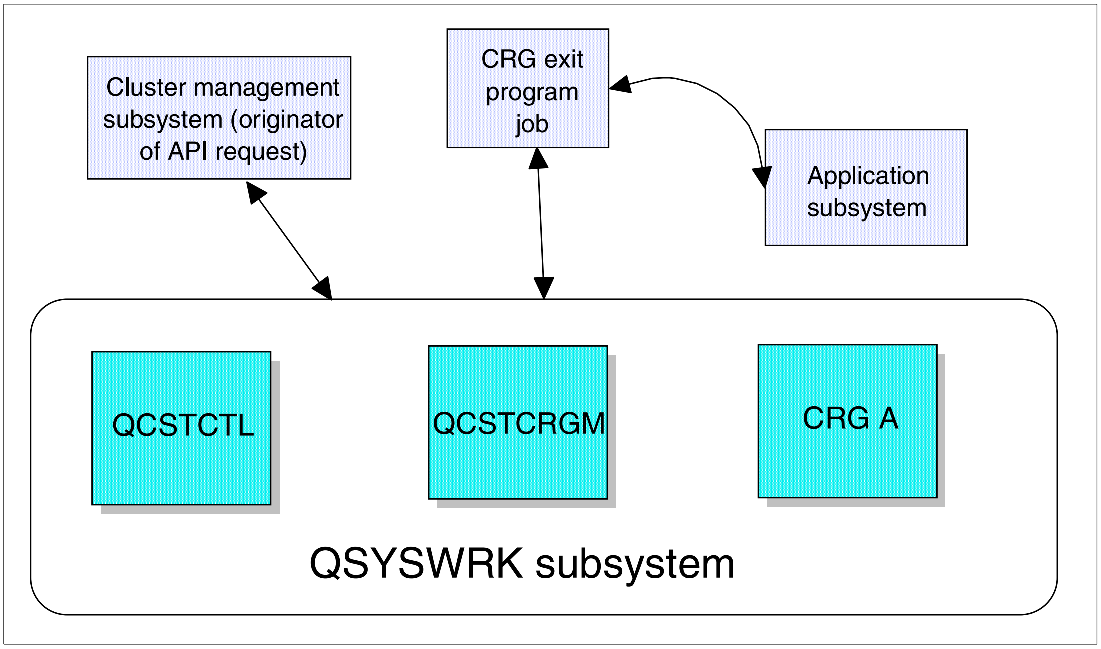


Figure A-2 Example job structure

In addition to the system jobs, consider the user job that originates the request for a cluster service. The request normally consists of a call to a clustering API. After the API is validated, the request passes to the appropriate cluster job in the QSYSWRK subsystem. The cluster job then handles the distributed processing of the request. Through a distributed activity group technology, the request is distributed to other members of the group on the other nodes of the cluster. The request is appropriately processed, and the results are returned to the node that initiated the request. Once responses are received from all participating members of the activity, the results are returned to the results information queue.

Finally, Cluster Resource Services initiates the exit program associated with the CRG on all active nodes in the recovery domain. These exit program jobs run in a user-specified subsystem, which can be the same as the application subsystem.

Typically, the exit program jobs are transitory and exist only for the duration of the API request. An exception is the CRG exit program that starts the resilient application, which runs only on the primary system. This job remains active and serves as a daemon job between the application and Cluster Resource Services.

A.2.4 Cluster engine services

Cluster Engine Group Membership Services provide the ability to define and modify distributed activity group membership definitions. Each live group member is notified of any change made to the definition or to the state of a member. Notification is via a special message called a *membership change message*.

The cluster engine ensures that cluster membership changes are handled consistently across affected groups for both administrative changes and changes as a result of a failure. Therefore, a consistent view of the membership is guaranteed across members of the same distributed activity group, as well as across related groups.

The messaging services provided to group members by the cluster engine include a variety of reliability and ordering guarantees over group messaging. These include:

- Non-reliable, FIFO ordered messaging

  FIFO messaging means that group messages sent by the same node are received in the same order by all the group members.

- Reliable, FIFO ordered messaging

  Reliable messaging is a variation of virtually synchronous messages. Members that appear in two consecutive membership change notifications receive the same set of messages between these notifications.

- Reliable, totally ordered messaging

  Totally ordered messaging implies that group members who receive the same set of messages receive them in the same order.

These guarantees are defined per group. A cluster engine provides the ability to send non-reliable messages to the group or to a subset of the group.

A.2.5 Cluster partition state

A cluster is in a partition state when the cluster cannot communicate with one or more nodes and no certain failure is identified. Do not confuse a *cluster partition* with a logical partition. A cluster partition is not good.

The typical cause of a cluster partition occurs when there is a communications link failure and a redundant path for the clusters is not established. Neither node has failed. But neither node knows the status of its counterpart. See Figure A-3 for an illustration.

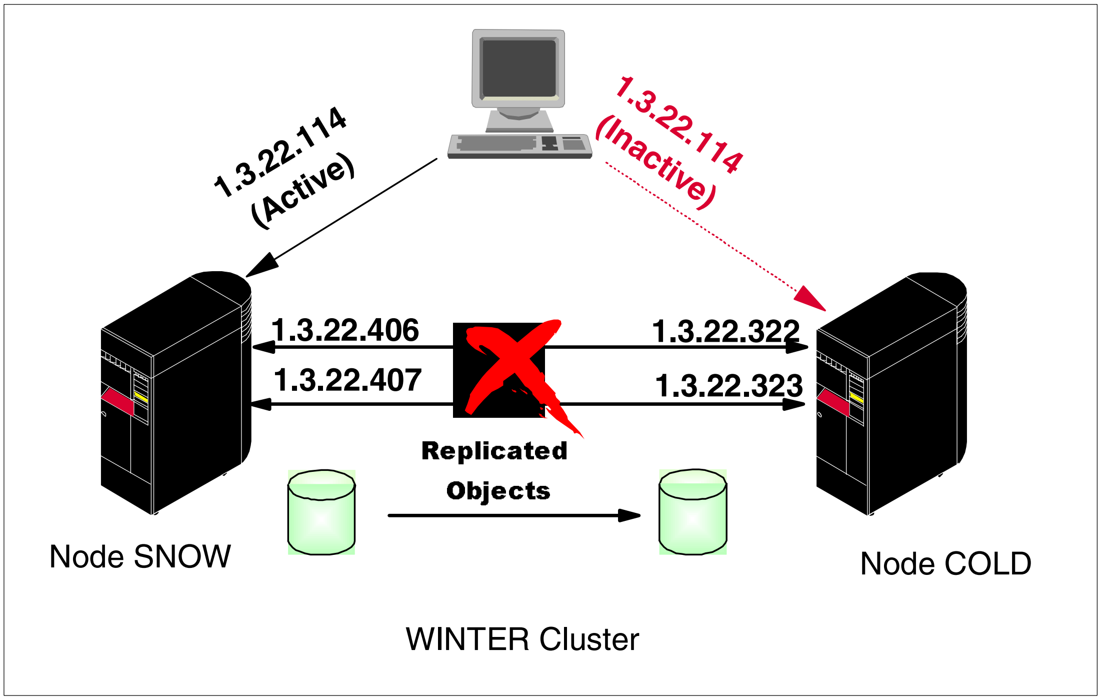


Figure A-3 Cluster partitions

What has failed is the communication between the SNOW and COLD nodes. When communication between nodes in the cluster is lost, and the status of the “missing” nodes is unknown, this is a *cluster partition*.

When a cluster partition occurs, failover should not be done because one node is still active. For example, in Figure A-3, the node named SNOW is still active. The IP address is still started on SNOW. Users can still access the application and the data on SNOW.

The nodes operate as independent until communication is restored. When communication between the nodes is restored, OS/400 cluster support merges the partitions to their original state.

When the cluster is partitioned, CRS takes overt action to ensure that:

- As many operations as possible can continue normally
- Operations that would cause inconsistencies between partitions are not allowed

The recovery goal is to ensure that the partitions can be joined back together (merged). If configuration and operational changes are allowed to be made independently in two or more partitions, there can be no guarantee that the merging of the partitions would be successful.

iSeries design of cluster partition recovery

The first problem to solve in a partitioned cluster scenario is to determine which cluster node is to assume the role of primary node for each of the CRGs.

There are numerous methods in the industry to determine where the primary role should be, including quorum protocols and complex voting algorithms. The design of the iSeries cluster architecture allows for the initial set of CRG types to cause the least amount of disruption to current operations. Therefore, the primary location for a group of resources is not moved when a partition occurs. The partition that includes the node with the primary role is called the primary partition for that CRG. Any other partition is known as a secondary partition. A node in the secondary partition cannot assume the role of primary.

The second problem to solve is to ensure that the partitions do not make inconsistent changes. An inconsistent change is one that would adversely affect the ability to bring the partitions back together (a *merge* of partitions). Operations on the cluster configuration and individual CRG that would prevent partition merging (for example, adding a node to the cluster in one partition) are not allowed.

Table A-1 summarizes the CRG operations allowed when clusters are in a partition state.

Table A-1 CRG operations allowed in LPAR partition

Cluster action	Primary partition	Secondary partition
Add node to recovery domain	Y	N
Change CRG	Y	N
Create CRG	N	N
Delete CRG	Y[1]	Y[1]
End CRG	Y	N
Initiate switchover	Y	N
List CRGs	Y	N
List CRG information	Y[2]	Y[2]
Remove node from recovery domain	Y	N
Start CRG	Y	N
1) After a merge, the CRG is also deleted from the secondary partition 2) The CRG must exist in the partition		

See 12.2, “Problems with the status of a cluster” on page 217, for a further discussion of cluster partitions.

A.2.6 Cluster versions

Cluster versions (or *cluster versioning*) are supported on the iSeries servers to enable a cluster node to recognize and interoperate with other cluster nodes that are at different release levels. Multiple releases of OS/400 can coexist in a single cluster. Therefore, an individual node can be upgraded to the next release of OS/400 without taking the cluster offline (and thereby minimizing the amount of planned unavailability for system maintenance).

To support this nondisruptive clustering environment, Cluster Resource Services implements levels of versioning beyond what is supported by existing iSeries server capabilities.

One level of versioning is in the objects used by CRS. Any internal change to an object causes the version information to change. When information is exchanged between nodes, the system services accounts for different object versions.

The second level of versioning is in the messages passed between nodes and between cluster components. Enhanced messaging, and therefore additional services, can be introduced without hindering the ability to communicate with nodes at the previous release level.

B

Referenced lists

This appendix contains lists and tables referred to from other locations in this redbook. They have been moved to this appendix to improve the readability of the main content of each chapter.

B.1 Clustering data area layout

> **Note:** This section is referenced from 9.2.2, “Input data area” on page 171, and 9.2.3, “Output data area” on page 172.

The layout of the QCSTHAAPPI input data area is shown in Table B-1.

Table B-1 QCSTHAAPPI data area

Offset	Type	Field
0	CHAR(10)	Data area level information
10	CHAR(10)	Application name
20	CHAR(6)	Application Release level
26	CHAR(50)	Application identification information
76	CHAR(34)	Reserved
110	CHAR(10)	Application CRG name (or *NONE or *LIST)
120	CHAR(20)	Qualified exit program name formatted as: ▸ CHAR(10) -- exit program name ▸ CHAR(10) -- library name
140	CHAR(10)	User profile name
150	CHAR(256)	Exit program data
406	CHAR(10)	Job name
416	BIN(4)	Application restart indicator
420	BIN(4)	Number of restarts
424	CHAR(1)	Application status
425	CHAR(35)	Reserved
460	BIN(4)	List type
464	BIN(4)	Number of entries (max of 20)
*	ARRAY(*) of CHAR(76)	List of entries array
These fields repeated for each entry in the list	CHAR(30)	Qualified name of file or data area formatted as: ▸ CHAR(10) -- File or Data Area name ▸ CHAR(10) -- Library name ▸ CHAR(10) -- Member name
	BIN(4)	Number of object specifiers
	CHAR(20)	Qualified default journal name or IASP name
	CHAR(1)	Criticality of data indicator
	CHAR(10)	Preferred CRG name
	CHAR(1)	Data Resilience Mechanism
	CHAR(10:	Reserved

The layout of the QCSTHAAPPO output data area is represented in Table B-2.

Table B-2 QCSTHAAPPO data area

Offset	Type	Description
0	CHAR(10)	Data area level information
10	BIN(4)	Success indicator
14	CHAR(10)	Cluster name
24	CHAR(10)	Application CRT name or *LIST
34	CHAR(16)	Takeover IP Address
50	CHAR(1)	Data resilience status
51	CHAR(19)	Reserved
70	IBIN(4)	Number of entries (max of 20)
*	ARRAY(*) of CHAR(40)	Name list array
These fields repeated for each name in the list	CHAR(10)	Object name (CRG or Data Area)
	CHAR(20)	Qualified journal name or IASP name
	CHAR(1)	CRG data status
	CHAR(1)	CRG Type
	CHAR(8)	Reserved

B.2 Unpacking and creating QUSRTOOL APIs and commands

Note: This section is referenced from Chapter 5, “Implementing and managing clusters with IBM solutions” on page 59.

Follow these steps to use the commands provided in the QUSRTOOL library:

1. Programs are available to change save files to source physical files and to change source physical files to save files (UNPACKAGE and PACKAGE respectively). Before any tools can be compiled and run, unpackage the appropriate save files.

   The write up of each tool identifies the members that make up the tool and the file in which they reside.

   Tip: To unpackage *all* save files in the QUSRTOOL library, run the following command:

   ```
   CALL QUSRTOOL/UNPACKAGE (‘*ALL          ‘ 1)
   ```

   To create the *install* program (TCSTCRT), enter:

   ```
   CRTCLPGM PGM(userlib/TCSTCRT) SRCFILE(QUSRTOOL/QATTCL)
   ```

   Here *userlib* is the name of the existing user library in which the install program and Cluster Resource Service commands is to reside.

2. Run the *install* program (TCSTCRT) to generate the cluster command objects. Enter the command:

```
CALL userlib/TCSTCRT userlib
```

Here *userlib* is the same as the *userlib* specified in step 1 on page 285. This creates all the objects necessary to use the cluster commands. The objects are created into library *userlib*.

B.3 Cluster APIs and related QUSRTOOL commands

Note: This section is referenced from:

- Chapter 5, “Implementing and managing clusters with IBM solutions” on page 59
- Section 5.3, “Using QUSRTOOL CL commands and OS/400 APIs to implement an iSeries cluster” on page 87
- Sample 10.1.1, “Creating the sample clustering environment” on page 185
- Section 12.3, “Recovering from a cluster partition state” on page 220

B.3.1 Cluster Control APIs and QUSRTOOL commands

Table B-3 lists the V5R1 Cluster Control APIs, with a brief description of what the API is used for, and the equivalent command available in the QUSRTOOL library.

Table B-3 Cluster Control API and QUSRTOOL command descriptions

Cluster Control API name	Description	QUSRTOOL command name
Add Cluster Node Entry (QcstAddClusterNodeEntry)	Adds a node to the membership list of an existing cluster. Also assigns the IP interface addresses to be used by cluster communications.	ADDCLUNODE
Add Device Domain Entry (QcstAddDeviceDomainEntry)	Adds a node to a device domain membership list so that it can participate in recovery actions for resilient devices. The addition of the first node to a device domain has the effect of creating that device domain.	ADDDEVDMNE
Adjust Cluster Version (QcstAdjustClusterVersion)	Adjusts the current cluster version to the next level, for example, so that new function can be used within the cluster.	CHGCLUVER
Change Cluster Node Entry (QcstChangeClusterNodeEntry)	Changes the fields in the cluster node entry. For example, the IP interface addresses used for cluster communications can be changed.	CHGCLUNODE
Change Cluster Resource Services (QcstChgClusterResourceServices)	Adjusts cluster performance and configuration tuning parameters to match the communications environment of the network used for cluster communications.	CHGCRS
Create Cluster (QcstCreateCluster)	Creates a new cluster of one or more nodes.	CRTCLU
Delete Cluster (QcstDeleteCluster)	Deletes an existing cluster. Cluster resource services is ended on all active cluster nodes and they are removed from the cluster.	DLTCLU

End Cluster Node (QcstEndClusterNode)	Ends Cluster Resource Services on one or all nodes in the membership list of an existing cluster. The node becomes unavailable to the cluster until it is restarted using the Start Cluster Node API.	ENDCLUNOD
List Cluster Information (QcstListClusterInfo)	Retrieves information about a cluster. For example, the complete cluster membership list can be returned.	PRTCLUINF
List Device Domain Information (QcstListDeviceDomainInfo)	Lists device domain information of a cluster. For example, the list of currently defined device domains can be returned.	PRTDEVDMNI
Remove Cluster Node Entry (QcstRemoveClusterNodeEntry)	Removes a node from the membership list of a cluster. The node is removed from any recovery domains, cluster operations are ended on the node, and all Cluster Resource Services objects are deleted from the node.	RMVCLUNODE
Remove Device Domain Entry (QcstRemoveDeviceDomainEntry)	Removes a node from a device domain membership list. If this is the last node in the device domain, this also has the effect of deleting the device domain from the cluster.	RMVDEVDMNE
Retrieve Cluster Information (QcstRetrieveClusterInfo)	Retrieves information about a cluster. For example, the cluster name and current cluster version are returned.	PRTCLUNAM
Retrieve Cluster Resource Services Information (QcstRetrieveCRSInfo)	Retrieves information about the Cluster Resource Services performance tuning and configuration parameters.	PRTCRSINF
Start Cluster Node (QcstStartClusterNode)	Starts Cluster Resource Services on a node that is part of a cluster but is currently not active. This API must be called on a node that is currently active in the cluster.	STRCLUNOD

You can find a description of how to create QUSRTOOL commands in B.2, “Unpacking and creating QUSRTOOL APIs and commands” on page 285.

B.3.2 Cluster Resource Group APIs and QUSRTOOL commands

Table B-4 lists the V5R1 Cluster Resource Group APIs, with a brief description of what the API is used for, and the equivalent command available in the QUSRTOOL library.

Table B-4 CRG Group API and command descriptions

Cluster Resource Group API name	**Description**	**QUSRTOOL command name**
Add CRG Device Entry (QcstAddClusterResourceGroupDev)	Adds a new device entry to a CRG. The device becomes a member of the group of switchable devices.	ADDCRGDEVE
Add Node to Recovery Domain (QcstAddNodeToRcvyDomain)	Adds a new node to the recovery domain of an existing CRG. A node can be added as a primary node (if the CRG is inactive), as a backup node, or as a replicate node.	ADDCRGNOD
Change CRG (QcstChangeClusterResourceGroup)	Changes attributes of a CRG. For example, the takeover IP address for an application CRG can be modified.	CHGCRG
Change CRG Device Entry (QcstChgClusterResourceGroupDev)	Changes a device entry in a CRG. For example, the option to vary the configuration object online at switchover or failover can be modified.	CHGCRGDEVE

Create CRG (QcstCreateClusterResourceGroup)	Creates a CRG object. The CRG object identifies a recovery domain, which is a set of nodes in the cluster that will play a role in recovery.	CRTCRG
Delete CRG (QcstDeleteClusterResourceGroup)	Deletes a CRG from the cluster. The CRG object will be deleted from all active systems in the recovery domain.	DLTCRGACT
Distribute Information (QcstDistributeInformation)	Delivers information from a node in the recovery domain of a CRG to other nodes in that CRG's recovery domain.	SNDCRGMSG
End CRG (QcstEndClusterResourceGroup)	Disables resiliency of the specified CRG. Upon successful completion of this API, the CRG status is set to inactive.	ENDCRG
Initiate Switchover (QcstInitiateSwitchover)	Causes an administrative switchover to be performed for the CRG. The recovery domain is changed so that the current primary node becomes the last backup and the current first backup node becomes the new primary.	CHGCRGPRI
List CRGs (QcstListClusterResourceGroups)	Generates a list of CRGs and some information about the CRG in the cluster.	PRTCRG
List CRG Information (QcstListClusterResourceGroupInf)	Returns the contents of a CRG object. For example, the recovery domain with the current node roles can be returned.	PRTCRGINF
Remove CRG Device Entry (QcstRemoveClusterResourceGroupDev)	Removes a device entry from a CRG. The device will no longer be a switchable resource.	RMVCRGDEVE
Remove Node From Recovery Domain (QcstRemoveNodeFromRcvyDomain)	Removes a node from the recovery domain of an existing CRG. The node will no longer participate in recovery action for that group of resources.	RMVCRGNOD
Start CRG (QcstStartClusterResourceGroup)	Enables resiliency for the specified CRG. The CRG becomes active within the cluster.	STRCRG

You can find a description of how to create QUSRTOOL commands in B.2, “Unpacking and creating QUSRTOOL APIs and commands” on page 285.

B.4 Object specifier file layout

Note: This section is referenced from 8.2, “ClusterProven defined” on page 163, 9.1.2, “ClusterProven applications” on page 170, and 11.2.2, “Resilient data” on page 199.

The Data Description Specification (DDS) for an object specifier file is shown in Table B-5. The DDS describes the record format used to identify objects for a replication solution.

Table B-5 Record format of object specifier file

Field name	Field type	Field definition
QCSTETYP	Bin(4)	Entry type (0 for library, 1 for directory)
QCSTOTYP	Char(10)	Object type or *ALL
QCSTEOAT	Char(10)	Extended object attributes (e.g. physical file)

Field name	Field type	Field definition
QCSTINEX	Bin(4)	Include or exclude indicator
QCSTRSRV	Char(14)	Reserved
QCSTCSID	Bin(4)	CCSID
QCSTCTID	Char(2)	Country (or region) ID
QCSTLGID	Char(3)	Language ID
QCSTNLSR	Char(3)	NLS reserved field
QCSTFLAG	Bin(4)	Flag byte
QCSTPTHL	Bin(4)	Number of bytes in path name field
QCSTPTHD	Char(4)	Path name delimiter
QCSTRSV2	CHAR(10)	Reserved

QCSTPTHN	Char(5000)
	VARLEN(100)
		Path name (Variable length field)

C

iSeries cluster resources

For customers, independent software vendors (ISVs) and IBM Business Partners who want to investigate iSeries highly available solutions and clusters, refer to the following resources and contacts:

- *iSeries Clustering with Switched Disks* - 2.5 day IBM classroom education, S6224
- iSeries Technology Center: Send e-mail to `rchclst@us.ibm.com`
- PartnerWorld for Developers: `http://www.developer.ibm.com/`
- Information Center Web site: `http://www.ibm.com/eserver/iseries/infocenter`
- iSeries Information Center CD-ROM (English version), SK3T-4091
- iSeries home page: `http://www.iseries.ibm.com`
- IBM @server iSeries cluster home page: `http://www.ibm.com/servers/clusters`
- IBM Direct: Call 1 800-426-2255

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 294.

- *The System Administrator's Companion to AS/400 Availability and Recovery,* SG24-2161
- *AS/400 Remote Journal Function for High Availability and Data Replication*, SG24-5189
- *AS/400e to IBM @server iSeries Migration: A Guide to System Upgrades at V4R5 and V5R1*, SG24-6055
- *Moving Applications to Switchable Independent ASP*s, SG24-6802

  This redbook is currently not available, but is scheduled for publication in the second half of 2002.
- *Roadmap to Availability on the iSeries 400,* REDP0501
- *High Availability on the AS/400 System: A System Manager’s Guide,* REDP0111

Other resources

These publications are also relevant as further information sources:

- iSeries Information Center CD-ROM (English version), SK3T-4091
- *iSeries Backup and Recovery,* SC41-5304
- *System API Reference*, SC41-5801
- Toigo, Jon. *Disaster Recovery Planning: Managing Risks and Catastrophe in Information Systems*. Yourdon Press, 1989. ISBN 0132149419

Referenced Web sites

These Web sites are also relevant as further information sources:

- iSeries home page: `http://www.iseries.ibm.com`
- IBM @server iSeries cluster home page: `http://www.ibm.com/servers/clusters`
- PartnerWorld for Developers: `http://www.developer.ibm.com/`
- Clusterproven applications and high availability: `http://www-1.ibm.com/servers/eserver/iseries/ha`
- Domino for iSeries clustering capabilities: `http://www.ibm.com/eserver/iseries/domino`
- Information Center: `http://www.ibm.com/eserver/iseries/infocenter`
- IBM Benchmark Center: `http://www.developer.ibm.com`
- DataMirror: `http://www.datamirror.com`

- Lakeview Technology: `http://www.lakeviewtech.com`
- Vision Solutions: `http://www.visionsolutions.com`

How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Glossary

This glossary provides definitions for terms used within this redbook. The terms are listed in alphabetical order.

application resilient (type-2) Enables an application (program) to be restarted on either the same node or a different node in the cluster.

backup node Has the ability to take over the functions of the primary node in the event of an outage (planned or unplanned) on the primary node. If there is more than one backup node, the recovery domain defines the order in which control is transferred to the backup nodes.

cluster A group of one or more servers that work together as a single system. A cluster is identified by a ten-character name.

cluster node Each iSeries that is a member of a cluster is a cluster node. Each cluster node is identified by an eight-character cluster node identifier (usually the system name). There are three types of cluster nodes: primary, backup, and replicate.

cluster policies Failover and switchover.

ClusterProven for iSeries An IBM designation that defines certain high-availability requirements that are applied to a software product either by itself or in combination with other software products. A solution that satisfies the technical criteria of these requirements can be validated with IBM and licensed to be marketed with the IBM ClusterProven trademark.

cluster resource Any part of the system that is available across multiple cluster nodes. The three types of system resources that can be resilient are:

- Objects that are kept up to date by using replication
- A resilient application and its associated IP address, which can be switched
- A resilient device that can be switched (IASPs)

Cluster Resource Group (CRG) A Cluster Resource Group is an OS/400 system object that is a set or group of cluster resources. The group describes a recovery domain and supplies the name of the Cluster Resource Group exit program that manages cluster-related events for that group. One such event would be moving an access point from one node to another node. Cluster Resource Group objects are either defined as data resilient (type-1), application resilient (type-2), or device resilient (type-3).

Cluster Resource Group exit program This program is called during different phases of the cluster environment and is responsible for establishing and managing the environment necessary for data and application resiliency within a cluster.

Cluster Resource Group manager (CRGM) Provides object management functions for the CRG's objects, such as creation, deletion, and modification. The CRGM also calls the CRG exit program whenever the status of the CRG changes.

cluster version (cluster versioning) The support to enable nodes in a cluster to communicate with other nodes in the cluster that have a different release level of OS/400.

data resilient (type-1) Enables multiple copies of data that is maintained on more than one node in a cluster.

device resilient (type-3) Is used with IASPs. Every Cluster Resource Group has a Cluster Resource Group exit program associated with it.

failover The system automatically switches over to one or more backup systems in the event of a system failure.

full clustering A *full*, automated high availability environment that uses clustering technology and takes advantage of V5R1 support. Full clustering includes these elements:

- Highly available hardware
- Highly reliable operating system
- Concurrent operations/maintenance
- Data resiliency
- Application resiliency
- Transaction monitoring
- Co-operation between OS/400 functions and business partner solutions

join To become a new member of some entity such as a cluster.

partitioned (cluster partition) When communication with a node is lost, but node failure cannot be guaranteed, a cluster then becomes partitioned.

recovery domain A subset of nodes in the cluster that are grouped together in a Cluster Resource Group for a common purpose such as performing a recovery action. A domain represents those nodes of the cluster from which cluster resources can be accessed. The subset of cluster nodes that is assigned to a particular Cluster Resource Group either supports the primary point of access, secondary (backup) point of access, or replicates.

rejoin To become an active member of a cluster after having been a nonparticipating member. For example, when clustering is restarted on a node after the node has been inactive, the cluster node rejoins the cluster.

replicate node Contains copies of information from the primary node. They do not have the ability to take over the functions of a primary or backup node. Typically, replicate nodes are used to store information for disaster recovery or for functions such as data warehousing.

replication The ability to synchronize a copy of data and objects from one system to another.

resilient resource Any system resource (data, a process, an application, or a device) that is available on more than one node in a cluster and that can be recovered if a node in a cluster fails.

switchover Happens if you manually switch access from one system to another. You would usually do this if you wanted to perform system maintenance such as applying program temporary fixes (PTFs), installing a new release, or upgrading your system.

switchover and failover order The relationship (or order) that you define among the primary node and backup nodes in a recovery domain. In a recovery domain, there can be multiple backup nodes. You specify one node as first backup, another as second backup, and so on. If a primary node fails, the access point for the resilient resources switches to the first backup node.

Index

IBM

Redbooks

Clustering and IASPs for Higher Availability on the IBM @server iSeries Server

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

IBM®

Clustering and IASPs for Higher Availability

on the IBM eserver iSeries Server

Redbooks

Moving the iSeries server beyond 99.9% availability

Independent ASPs and other V5R1 high availabiity solutions

Solutions for application and data resiliency

With OS/400 V5R1, IBM eserver iSeries servers support two methods of clustering. *Replication technology* is one method. The other method is *switchable disk technology*, which is referred to as *independent auxiliary storage pools (IASPs)* on the iSeries server.

This IBM Redbook presents an overview of cluster and switched disk technology available at OS/400 Version 5 Release 1. It explains the concepts and architecture surrounding iSeries clusters. It introduces you to the eserver brand initiative – ClusterProven for iSeries – and explains how it applies to iSeries customers and independent software vendors. Application resiliency can be delivered by exploiting OS/400 cluster technology and cluster management services such as those provided by IBM High Availability Business Partners. It is available through IBM cluster middleware providers. Considerations for this application design are also introduced in this redbook.

This redbook is written for IBM customers, technical representatives, and Business Partners who plan business solutions and systems that are continuously available. You should use this book to gain a broad understanding of the cluster architecture available with OS/400 Version 5, Release 1, where clustering is viable. You should also use it to learn how to plan and implement clustering and independent ASPs.

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

SG24-5194-01 ISBN 0738422355