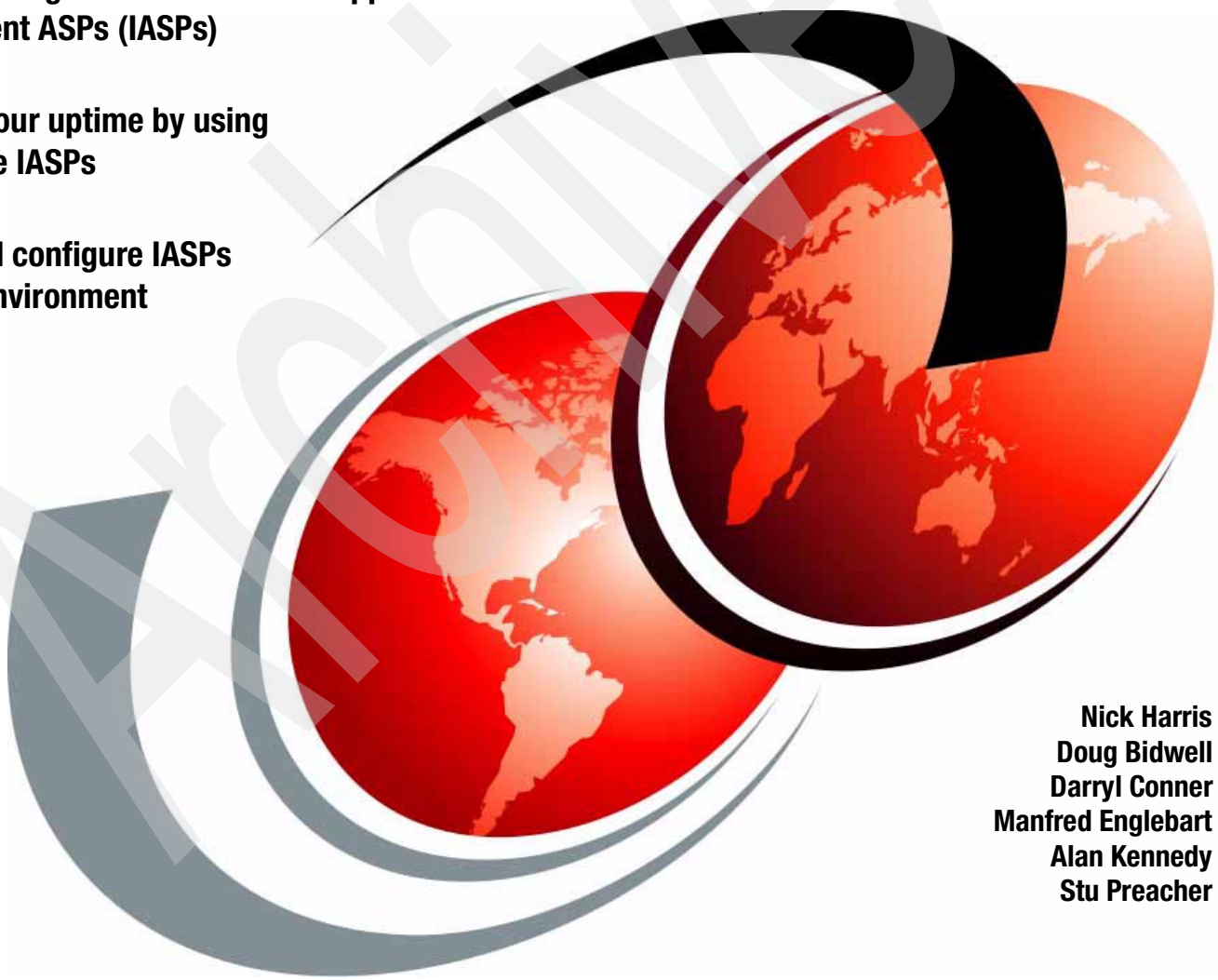
IBM

IBM @server iSeries Independent ASPs

A Guide to Moving Applications to IASPs

- Take advantage of the new V5R2 support for independent ASPs (IASPs)
- Improve your uptime by using switchable IASPs
- Install and configure IASPs for your environment



Nick Harris
Doug Bidwell
Darryl Conner
Manfred Englebart
Alan Kennedy
Stu Preacher

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM @server iSeries Independent ASPs: A Guide to Moving Applications to IASPs

May 2003

SG24-6802-00

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

First Edition (May 2003)

This edition applies to Version 5 Release 2 of OS/400.

Contents

Notices

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

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

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

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

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

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

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

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

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

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

Trademarks

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

@server™	AS/400®	Notes®
eServer™	Distributed Relational Database	Operating System/400®
ibm.com®	Architecture™	OS/400®
iSeries™	Domino™	QMF™
xSeries®	DB2®	Redbooks(logo) ™
Advanced Function Printing™	DFS™	Redbooks™
AnyNet®	DRDA®	S/370™
AFP™	Infoprint®	System/36™
AS/400e™	IBM®	System/38™

The following terms are trademarks of other companies:

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

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

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

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

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

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

This IBM® Redbook explains how to install and configure the new independent auxiliary storage pool (IASP) functionality of OS/400® V5R2. It is designed to help IBM technical professionals, Business Partners, and Customers understand and implement IASP in the IBM @server™ iSeries™ server and under OS/400 V5R2.

In addition this redbook provides the background information that is necessary to plan, implement, and customize this functionality to your particular environment. It provides advice on running native OS/400 applications with either application data or most application objects residing in an IASP. Considering you can also use IASPs in a cluster environment, this redbook shows you the basic steps to make your IASP switchable between two iSeries servers in a high-speed link (HSL) loop.

This redbook does not remove the need for hands-on experience. Hands-on courses and an IASP study are available from the iSeries Technology Center (iTC) in Rochester, Minnesota. You can learn more about their courses and schedules by visiting their Web site at:

http://www-1.ibm.com/servers/eserver/iseries/service/itc/

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 (ITSO), Rochester Center.

Nick Harris is a Senior Systems Specialist for IBM and has spent the last three years in the ITSO, Rochester Center. He specializes in LPAR, iSeries hardware and software, external disk, Integrated xSeries™ Server for iSeries, and Linux. He also writes and teaches IBM classes worldwide on areas of AS/400® and iSeries system design and server consolidation. He spent 13 years in the United Kingdom AS/400 Business and has experience in S/36, S/38, AS/400, and iSeries servers.

Doug Bidwell is Systems Engineering Manager of DLB Associates, Inc., an IBM e-business Business Partner in Valencia, California. He has 32 years of experience in IBM Midrange systems. His areas of expertise include performance, capacity planning, and network administration. He has written extensively on these subjects over the years and has been published in several trade magazines. He currently authors the PTF Guide on a weekly basis for Midrange Server.

Darryl Conner is a Technical Consultant for IBM Business Partner Avnet Computer Marketing, a division of Avnet Inc. Avnet is the world's largest distributor of semiconductors, interconnect, passive and electromechanical components, computer products, and embedded systems from leading manufacturers. He has 19 years of experience in working with the System/34, System/36, System/38™, AS/400, and iSeries servers. His areas of expertise include backup and recovery, system migration, upgrade planning, and implementation.

Alan Kennedy is a Senior MIS Consultant for Office Depot, Inc. of Delray Beach, Florida. He has 18 years of experience on System 3X/iSeries. His areas of expertise include communications, work management, upgrade planning/implementation, and third-party product applications integration.

Stu Preacher is a certified IT Consultant specializing in AS/400 and iSeries systems management and high availability. He has worked for IBM for over 28 years. Previously he worked on System/34 and System/38 and has been working with the AS/400 and iSeries servers since their introduction. Currently, he is group leader for iSeries Technical Support in the United Kingdom and is responsible for iSeries External Storage in Europe. He has worked with many of the largest AS/400 and iSeries customers across Europe.

Patrick Wildt is a Staff Software Engineer for IBM Rochester. He has 13 years of experience on the iSeries platform, with eleven years spent working in the OS/400 development lab in the Systems and Network Management area. He has spent the last two years working in the eServer Custom Technology Center on various customer contracts and engagements. He holds a bachelors degree in computer science from North Dakota State University. His areas of expertise include systems management and communications.

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

Ted Bauer
Jim Denton
Selwyn Dickey
Dennis Martin
Ron Peterson
Jim Ranweiler
Nick Ravn
Rick Sanders
Mike Snyder
Dan Tarara
IBM Rochester

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our 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:

  IBM Corporation, International Technical Support Organization
  Dept. JLU Building 107-2
  3605 Highway 52N
  Rochester, Minnesota 55901-7829

1

Introduction to independent auxiliary storage pools (IASPs)

An independent disk pool, or independent auxiliary storage pool (ASP), is a collection of disk units that can be brought online or taken offline independent of the rest of the storage on a system, including the system ASP, user ASPs, and other independent disk pools. An independent disk pool can be either:

- **Private**: Privately connected to a single system, also known as stand-alone IASPs
- **Switchable**: Switched between two systems or partitions in a clustered environment

Clearly, this is quite a departure from the way in which auxiliary storage (disk) was regarded prior to V5R1. Until then, all iSeries disks were considered to be owned and usable only by a single system. Enhancements made to OS/400 in V5R1 and again in V5R2 make using independent disk pools an attractive option for many customers who are looking for higher levels of availability and server consolidation.

This chapter looks at the advent of the independent disk pool and introduces new terminology and features. It also explains how independent disk pools work, the benefits of implementing independent disk pools, and some restrictions and considerations.

Throughout this book, the terms *independent disk pool* and *independent auxiliary storage pool* are synonymous.

Note: Although clustering support is required to enable switchable disk pools, this book does not go into detail about creating and managing clusters. Instead, refer to *Clustering and IASPs for Higher Availability on the IBM @server iSeries Server*, SG24-5194, for more information about clustering.

1.1 Disk storage: A brief history

When System/38, the predecessor to the AS/400, was introduced, all disk storage was regarded as being in “one big bucket”. When it was first introduced, System/38 had a maximum disk capacity of 0.38 GB. This concept of *single-level storage* had tremendous benefits to programmers and technical support staff. They no longer had to consider where data was stored (although with only 387 MB available, it wouldn’t have been too difficult). This view of storage continued right through the AS/400 range and into the iSeries.

Programs, files, and other structures are regarded simply as “objects” and are grouped into libraries. In the early days, accessing one of these objects was simple. You only had to specify the library or object name, and the object was found. You could even simplify this by providing a list of libraries to be searched to locate an object. This was a completely new and much simplified way to access your data and programs compared to the very rigid structure of more traditional computers such as the IBM S/370™ and follow-on systems. Most computer systems at this time were relatively expensive and tended to be located and managed by a centralized support staff.

All disk drives and the data they contained tended to have similar characteristics. Managing a System/38 or an AS/400 was a pretty simple task. Many companies benefitted considerably from this design by not having to employ support staff to manage their computer system. When the AS/400 was introduced, it had a maximum disk capacity of 27.3 GB.

As prices reduced and company departments became more autonomous, they started to buy their own departmental systems. This was also due in part to the simplified ease-of-use of systems such as the AS/400. As time passed, disk technology developed. Larger and larger disk drives were developed for AS/400 systems, increasing the data capacity significantly. New techniques, such as database journaling, RAID-5, and disk mirroring, were introduced to help improve system availability.

Quite early in the development cycle, IBM introduced the concept of user auxiliary storage pools, which allowed the overall disk storage to be “ring-fenced” into different pools, each of which could be regarded as a “big bucket”. However, the whole was still a single name space, where all library names had to be unique.

And still the disk capacity grew. When the iSeries was announced, it had a maximum capacity of 4,294.9 GB. At the time this redbook was written, the maximum disk capacity on an iSeries holds 38 TB or 37,978.2 GB. This is a 100,000-fold increase since the introduction of System/38! It is no wonder that techniques used to managed disk storage have developed over the years.

1.2 What’s new in V5R2

Independent disk pools were introduced in OS/400 V5R1. They were enhanced in V5R2 to provide support for:

- **Library-based objects**: In previous releases, independent disk pools provided support for integrated file system (IFS) and user-defined file system (UDFS) type objects only. Support for library-based objects was added in V5R2.
- **Up to 223 independent disk pools**: In previous releases, only 67 independent disk pools were supported, numbering from 33 to 99. In V5R2, you can create as many as 223 independent disk pool, from 33 to 255.
- **Disk pool groups**: In V5R2, you can define a disk pool group, which is made up of a primary disk pool and optionally one or more secondary disk pools. Each secondary disk

pool is independent in regard to data storage, but combines to act as one entity. If you make one disk pool available or unavailable, the remaining disk pools in the group are also made available or unavailable at the same time. The primary and secondary disk pools in a disk pool group share the same database.

- **Multiple databases**: In previous releases, only one database was available per system. It could only span libraries in the system ASP and user ASPs 2 to 32. In V5R2, when a primary independent disk pool group is created, it appears as a distinct user database on the server, separate from the system database.
- **Same library name in multiple instances**: With multiple databases, you can now use the same library name in multiple instances on the same system. This works as long as each occurrence of the library name exists in a separate database.
- **System ASP (*SYSBAS)**: This remains unchanged from previous releases. It contains OS/400 and Licensed Program Products (LPPs), plus any user objects.
- **Basic user ASPs**: Prior to V5R2, auxiliary storage pools 2 to 32 were known as *user ASPs*. Their function has not changed, although they are now often referred to *basic user ASPs* or *basic disk pools* in V5R2.
- **Independent disk pool**: This disk pool contains objects, directories, or libraries that contain the objects, and other object attributes such as authorization and ownership attributes. An independent disk pool can be made available (varied on) and made unavailable (varied off) to the server without restarting the system.

  When an independent disk pool is associated with a switchable hardware group, it becomes a *switchable disk pool* and can be switched between one iSeries server and another iSeries server in a clustered environment. An independent disk pool that is not associated with a cluster resource group is referred to in OS/400 application programming interfaces (APIs) as a *private disk pool*. Independent disk pools can also function in conjunction with other independent disk pools in a disk pool group.

  The following definitions describe the three types of independent disk pools:

  – *User-defined file system*: An independent disk pool that contains only user-defined file systems. It cannot be a member of a disk pool group unless it is converted to a primary or secondary disk pool. UDFS disk pools were made available with V5R1.

  – *Primary disk pool*: An independent disk pool that defines a collection of directories and libraries and may have other secondary disk pools associated with it. Primary disk pools can only be implemented on V5R2 or later of OS/400.

  – *Secondary disk pool*: An independent disk pool that defines a collection of directories and libraries and must be associated with a primary disk pool. Secondary disk pools can only be implemented on V5R2 or later of OS/400.

These features are shown in Figure 1-1 and are summarized in Table 1-1.

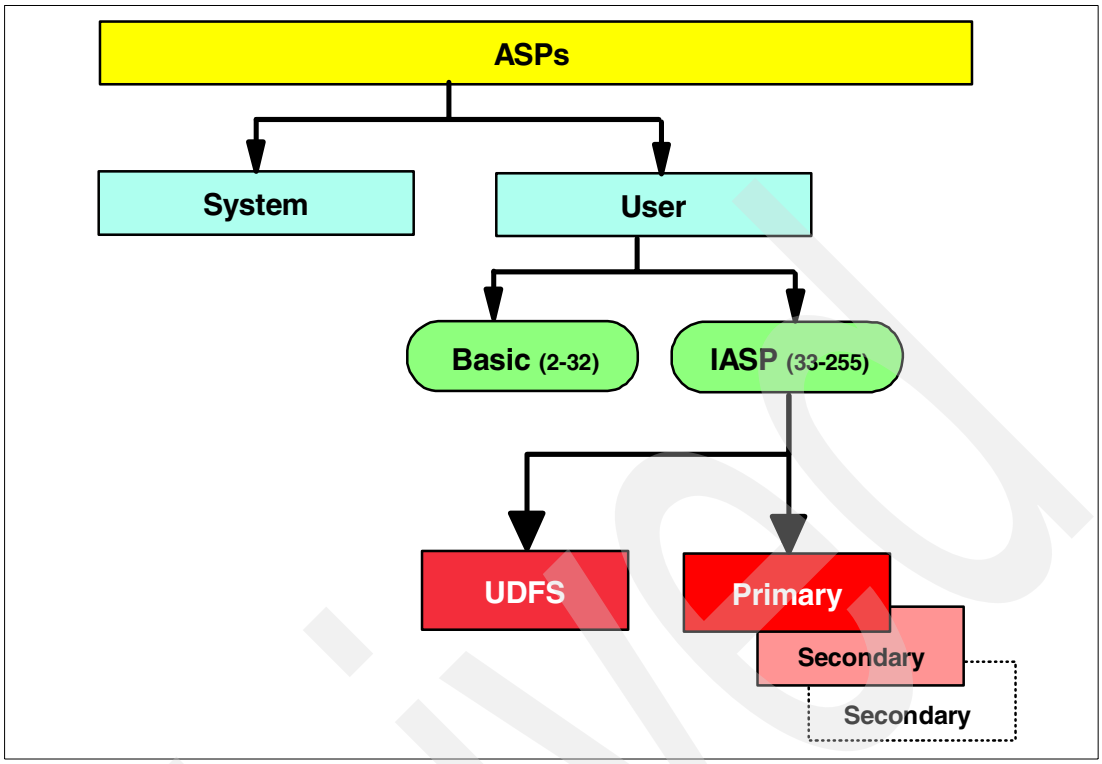


Figure 1-1 ASP explanation

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

Type of ASP	OS/400 release supported	ASP numbers	Maximum quantity supported on the system
System ASP [2]	All	ASP 1	1
Basic user ASP [1, 2]	V4R5 and earlier	ASP 02 to ASP 16	15
Basic user ASP [1, 2]	V5R1	ASP 02 to ASP 32	31
Independent ASP	V5R1	ASP 33 to ASP 99	67
Independent ASP	V5R2	ASP 33 to ASP 255	223

Notes:
1. This is also referred to as basic ASP or user ASP.
2. *SYSBAS is the system ASP and all basic user ASPs.

1.3 Positioning IASPs

There are several possible configurations, but only two basic environments in which independent disk pools can be used:

- A single system environment with a single iSeries server (non-logical partition (LPAR))
- A multisystem (or multipartition) environment managed by an iSeries cluster

1.3.1 Single system environment

In a single system environment, you can take independent disk pools offline or make them unavailable, independent of other disk pools. You can also bring the independent disk pool online or make it available, while the system is active, without performing an initial program load (IPL).

Figure 1-2 shows an example of multiple databases that reside in independent disk pools. This example has independent disk pools for the Payroll data, Order entry data, and data for Companies 1, 2, and 3. The actual application code could reside in the system ASP or another disk pool (either a user ASP or another IASP).

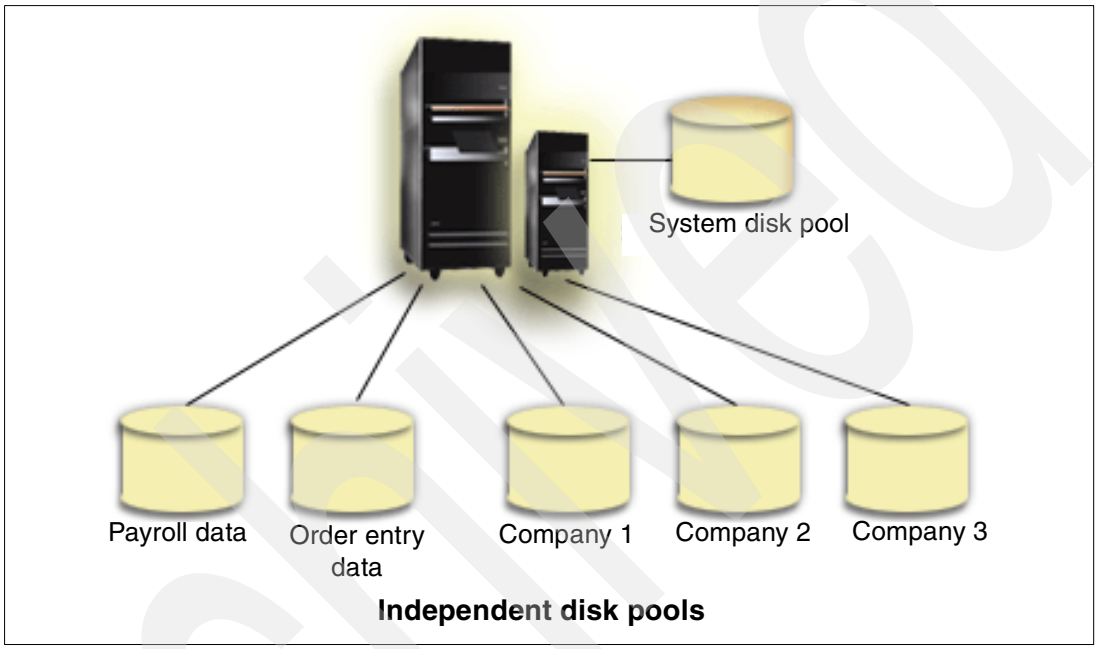


Figure 1-2 Private disk pools

A typical use of independent disk pools as shown in this example is for server consolidation of multiple branch office or store systems. Corporate data can reside in the other independent disk pools. Segmenting your databases in this manner allows for greater control and flexibility.

1.3.2 Multisystem clustered environment

Consider a multisystem clustered environment, where the servers are members of an iSeries cluster, and an independent disk pool is associated with a switchable hardware group in that cluster. In this environment, independent disk pools can be switched between systems without having to perform an IPL. In this environment, the independent disk pool is switchable because it resides on a switchable device. This device can be an external expansion unit or tower. Or it can be an IOP on the bus shared by two or more logical partitions.

This switch can be either automatic in the case of an unplanned outage, or manual by administering a switchover. The independent disk pool can be switched because it is self contained. This can be a significant advantage because it allows for continuous availability of data, one of the primary benefits of independent switchable disk pools.

Figure 1-3 shows an example of a switchable disk pool. It has two system units and four expansion units. The lower three expansion units are owned by the system unit on the left of the diagram. The upper expansion unit is owned by the system unit on the right. The lower three expansion units are switchable disk pools and are normally used by the system on the left.

In the event of an outage, either planned or unplanned, these three expansion units can be switched to the backup system on the right. This can be done manually in the case of a planned outage (for example, in the event of a system software upgrade) or automatically by using Cluster Resource Services in the event of an unplanned outage such as a hardware failure on the system unit.

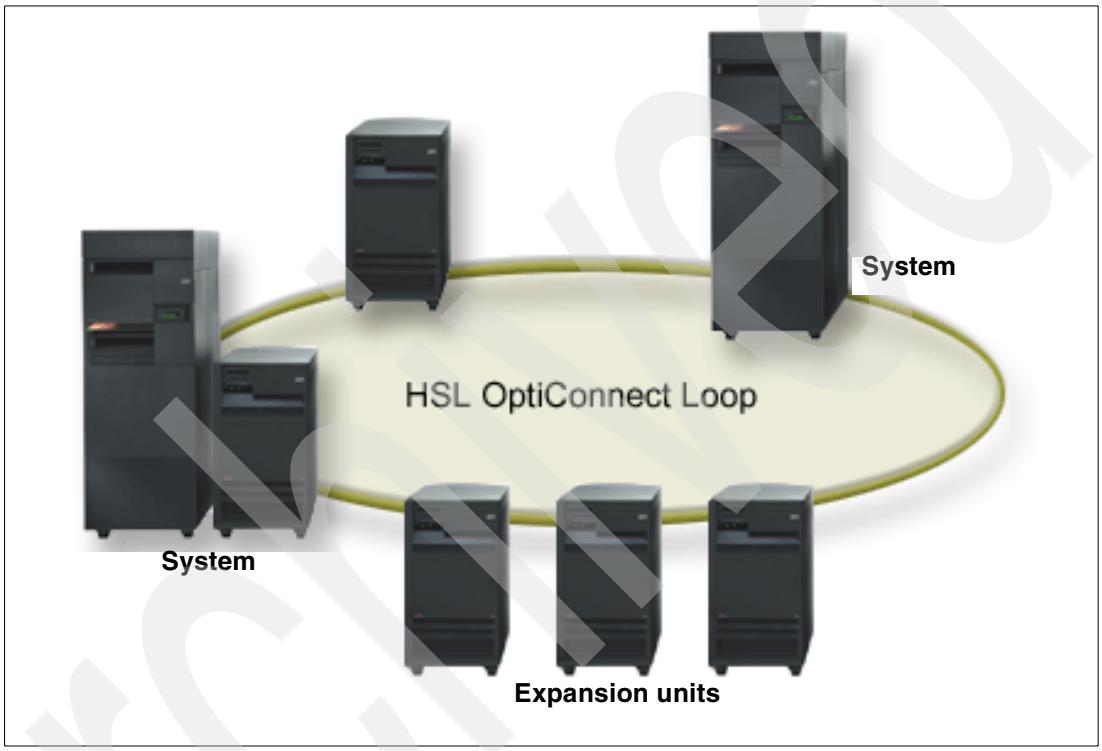


Figure 1-3 Switchable disk pools

1.3.3 Who benefits from using independent disk pools

The concept of “independent storage” is new to the iSeries server. For many years, iSeries customers, and AS/400 and System/38 customers before that, used a single view of storage, which was totally owned by the system to which it was attached.

User ASPs were introduced on System/38 to allow the disk storage attached to a single system to be grouped into separate pools. However, these pools always had a close relationship to the system ASP. You could not have duplicate library names in different user ASPs. In effect, all the storage attached to a system was a single name space.

In more recent versions of OS/400, various file systems, such as QOpenSys, QNetWare, and QNTC, were introduced. They were generically known as the *integrated file system* (IFS). By defining user-defined file systems, the IFS objects could be stored in different user ASPs. However, these were limited to only being available on the system to which the disk storage was attached.

In V5R1 of OS/400, the first move away from this single view of storage was introduced with the ability to switch IASPs between systems or logical partitions. This initial introduction of IASPs was limited to IFS objects. As such, iSeries customers had limited use of IASPs for most Domino™, Web serving applications, and Integrated xSeries Adapters or Servers.

With the enhancements introduced in V5R2 to support the traditional library and object structure in the QSYS.LIB file system used by the majority of iSeries applications, independent disk pools introduce many new potential advantages to most customers. These include:

- Availability:
  - Keep data available to an application even in the event of a single system outage
  - An alternative process to replicating data from one system to another
  - Isolate disk unit failures within the independent disk pool
- Server consolidation of “branch office” type systems
- Workload balancing across multiple servers through switching
- Isolation of low-use data with the ability to go online only when needed: Reduces system start time
- Reclaim storage by independent disk pool
- Isolation of data associated with specific applications
- Save and restore management by independent disk pool
- Perform application maintenance that does not affect the entire system

2

Planning for IASPs

This chapter discusses the various elements of planning for independent auxiliary storage pools (IASPs). The planning steps illustrated here represent the process involved. The full extent to which planning for IASPs can be taken is a subjective process.

Costs are always a consideration. However, for purposes of this discussion, we assume that all costs are covered.

2.1 Business needs

A business need or requirement generally drives the upgrade of every system. Usually, business needs are quantified, in terms of volumes of data, throughput, response time, etc.

When carrying out the planning for IASPs, keep in mind the business needs. In each step of the planning process, you must satisfy the business need before you approve and adopt the plan. This constant verification adds to support and justification for your environment. It may also assist with determining and moving to higher levels of availability.

2.2 Performance requirements

Performance is usually one metric of business needs. Even if it's not a primary consideration, maximizing throughput of the configuration is a small effort with a potentially high return. Consider the following key performance areas:

- **Processor capability**: In an IASP environment, the processing resources are not separated or divided among the various applications. They are available to everyone. Control is managed using OS/400 work management techniques.
- **Number of disk arms**: This is important when moving from a single auxiliary storage pool (ASP) environment to an IASP environment. If an application requires 20 arms to achieve good performance in the system ASP, it may take 20 arms in an IASP. Arms are also required in *SYSBAS, so don't just remove them all and leave the load source unit.
- **Size of system ASP**: The size of the system ASP is important for two reasons:
  - Performance of the operating system and any applications that remain in *SYSBAS
  - Performance of SQL functions related to the application in the IASP: The temporary space used for running queries is still found from the system ASP. There needs to be enough space to run the queries and sufficient arms to allow the application to perform to requirements.
- **Size of the system ASP related to IASP IPL time**: The relationship between the size and number of objects in the system ASP and the IASP affects the vary-on time of the IASP. Keep the system ASP size and number of objects at a minimum, or the IASP IPL may be protracted.
- **On demand processors**: With the new IBM @server On/Off Capacity on Demand for iSeries models, you have the potential for considerable changes in processing environments. These changes can severely impact performance. Changes to the number of processors cause the access plan to be revalidated and may require a different access plan.

  For a few not-too-complex queries, this should go unnoticed. But for thousands of small queries, the addition of a second processor to revalidate the plan can cause a significant performance degradation. Similarly, changes in the number of processors increases or decreases the amount of parallelism. Very large queries may change their access plan, resulting in variable performance.
- **SQL packages**: These packages may also be affected by moving their data to an IASP that is switchable. If the IASPs are different sizes and use the same SQL package, the access plan changes each time a different IASP is selected by a user.

As we progress through this book, we mention performance at various points.

2.2.1 Structure of IASPs

Because of the structure of IASPs and their relationship with the system ASP, there is a performance impact to the system for each IASP in use. The more IASPs are involved, the bigger the impact is. During normal operations, this overhead is in communications between the IASP tables and system tables, keeping them synchronized. There is a much greater impact when these IASPs first come online to the system. A few reasons for this are:

- If this is the first vary on since an IPL, the objects are verified for consistency.
- After each vary on of the IASP, the user IDs and AUTLs are created as required, if they don't already exist.
- Making more than one IASP available at a time to the system means that this synchronization between the system and the ASP must take place concurrently (serially can yield better system performance).

2.2.2 Disk drives: Arms versus capacity

The number of disk arms on any system is vital for good application performance. This section discusses the impact of disk arms, and why you need to achieve some balance between number of arms and disk capacity. Any IBM direct access storage device (DASD) sold as compatible with V5R1 or V5R2 is compatible with IASP usage.

Mixing drive capacities within a pool is the most difficult for tracking performance characteristics, at the pool level. If application performance is tracked, the job is easier if all disk drives within a disk pool are the same speed and size.

Mixing drives distorts the performance characteristics available though performance tools. Mixing drive sizes may significantly reduce performance. The amount and type of data to be placed within the IASP determines the type and placement of the disk.

The type of data to be placed in an IASP may also determine the performance requirements. If the data is historical and infrequently accessed, slow access or reduced performance may be perfectly acceptable.

Arms

Systems can run (slowly) with as few as one disk arm in the system ASP (you would *not* actually do this). In testing, as few as five disk arms in the system ASP showed degradation, when supporting an IASP of thirty arms. When ten arms supported the same IASP of thirty arms, performance was acceptable.

Moving applications to IASPs and sharing *SYSBAS is not the same as moving applications from a single partition to a multipartition server with a managing partition (also known as a *thin primary*). You should not remove all resources from *SYSBAS.

There is not a rule of thumb for a ratio of arms between the system ASP and an IASP. However, arms in *SYSBAS are very important for performance.

Capacity

The size relationship between *SYSBAS and an IASP is important. The ratio of one to two (1:2) or one to three (1:3) for disk arms can be a good start for sizing disk capacity. If 30 GB of disk is required in the IASP, then 10 GB of disk is required for the system ASP. Keep in mind that all temporary objects for applications residing on an IASP are created in the system ASP.

When the application runs in the system ASP, the amount of storage used is a combination of data storage and temporary storage plus other stored objects. The real data storage requirement for an IASP is somewhat less than the total original storage capacity.

But as disk arms do, a balance between *SYSBAS and an IASP must be gained. If not, there is the risk of serious performance problems.

Important: When you run a query over data in an IASP, temporary objects are automatically created in the system ASP. If you do not allow sufficient capacity in the system ASP for the temporary objects, the system ASP will fill. If the query is allowed to run and fill the system ASP, the system will crash.

Qualifier

We also tested the actual performance characteristics and sizings centered around IASPs. Every real customer situation is going to be different. The rule of thumb above is presented as a starting point. Your own requirements may vary.

2.2.3 Placement for performance

After you determine the correct arms for the capacity necessary for the application being placed in the IASP, you must then determine the placement of disks within the expansion units and cards on the available buses.

Disks and arms should be spread evenly along among the available input/output adapters (IOAs) in the tower (5074/5079). PCI cards should be placed from right to left, according to the card placement rules, within the available buses or multi-access bridge sets.

2.3 Software licensing requirements

System ASPs and user ASPs are supported under all releases. UDFS ASPs are supported under V5R1. Primary and secondary ASPs are supported under V5R2. You need to consider other software components that make up and are required for your system to operate.

Where the IASP is switchable and the application resides in *SYSBAS, a copy of the application is required on each server. Copies of OS/400 optional products and Licensed Program Products (LPPs) are required on all servers that have switchable resources.

2.3.1 Required software

The required software is iSeries Access for Windows, iSeries Navigator (5722-XE1). iSeries Navigator is the graphical user interface (GUI) for managing and administering your iSeries server from your Windows desktop. It is required to perform some of the disk management tasks necessary to implement independent disk pools.

2.3.2 Optional software

The following products are optional:

- **OptiConnect for iSeries, Product Option 23 (5722-SS1)**

  This product provides high-speed transparent access to data through fiber optic bus connections and performance enhancements to iSeries Distributed Data Management (DDM). This product is not required for normal IASP functionality. This is a charged feature.

- **Object Connect, Product Option 22 (5722-SS1)**

  This product provides support to simply and efficiently move individual objects, entire libraries, or entire integrated file system (IFS) directories from one iSeries server to another over a standard communications connection or over a high-speed fiber optic bus. Systems can be connected via:

  – Standard advanced program-to-program communication (APPC) using Advanced Peer-to-Peer Networking (APPN)

  – TCP/IP communications lines using AnyNet®

  – A fiber optic bus using OptiConnect for iSeries

  The economy of not requiring intermediate save file procedures and copies to distribution queues saves DASD. It improves performance in a manner that is non-disruptive to system operations. This product is not required for normal IASP functionality. This is a no-charge feature.

- **OS/400 - HA Switchable Resources, Product Option 41 (5722-SS1)**

  This is required when setting up simple clustering, for switching between two systems.

- **HA Journal Performance, Product Option 42 (5722-SS1)**

  This is a journal caching feature. You should review it if there is a plan to put journals in secondary IASPs.

2.4 Restrictions

Since independent disk pools are self-contained, they present unique restrictions. All of the necessary system information associated with the objects contained on an independent disk pool is contained in that pool. Because of this characteristic, there are certain restrictions when using independent disk pools.

2.4.1 Switching independent disk pools between V5R1 and V5R2 systems

After an independent disk pool is made available on a V5R2 system, it cannot be made available on a V5R1 system. It is possible to switch a V5R1 independent disk pool to a V5R2 system and make it available. However, after it is made available on the V5R2 system, its internal contents are changed, and it cannot be made available to the V5R1 system again.

2.4.2 Spooling limitations

Neither OUTQs nor JOBQs can be stored on an independent disk pool. This means that any output from or input to applications resident in an IASP are separated from that application when or if that application is switched to another system.

However, some external printing resources may be stored there. You may store formatting objects, such as *FNTRSC, *FORMDF, *OVL, *PAGDFN, and *PAGSEG, in an independent disk pool. For the writer job to access these objects, you must set the disk pool group with the Set Auxiliary Storage Pool Group (SETASPGRP) command before using the Start Printer Writer (STRPRTWTR) command. This makes the objects available in the library name space.

If the actual printouts are important, they may be used as a transaction log. Another alternative is to use Infoprint® Server for iSeries. The print is spooled to the normal OUTQs in OS/400, with the resulting output in Portable Document Format (PDF) from Infoprint Server. You can store these PDFs in the IFS within an IASP. This means that the printed output will switch with the IASP. This does not solve the problem of prints that have not yet completed and these need to be handled within application recovery after switchover or failover.

2.5 Application integration

On V5R1, only IFS-supported objects are supported in an IASP. For V5R2, refer to the list of supported and unsupported objects in “Table F-1 lists all object types on V5R2 and whether they are supported in an IASP. This table is in Object type sequence.” on page 201. Make careful note of the considerations by object type for supported objects.

Traditionally, data objects pertaining to an application area are stored in a data library. Program objects pertaining to an application area are stored in a program library. Other objects common to the application area are stored in system libraries or libraries designated as common to that application.

USRPRF, AUTL, JOBQ, and OUTQs are a few of the object types that cannot exist in an IASP. These are left to default to system libraries at application installation time in most cases. However, be sure to consider how these are replicated, saved, restored, or used in light of independent ASPs

You need to review the applications to be loaded on the ASPs in terms of their current level of integration between themselves and the operating system. The example application in Chapter 8, “Installing applications” on page 107, involves the Phoenix system subsystem descriptions. We decided to keep these descriptions and their entries in the system ASP. In this case, 10 branches are distributed and are then consolidated into one central system. The level of integration may require drilling down to the member level to determine the best way to consolidate them, yet still use an IASP.

2.6 Authority considerations

Considerations relating to authority is a key part of moving application to IASPs. This section discusses the characteristics of user profiles and authorization lists as they relate to the security of an IASP.

2.6.1 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 is 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

Figure 2-1 illustrates this concept.

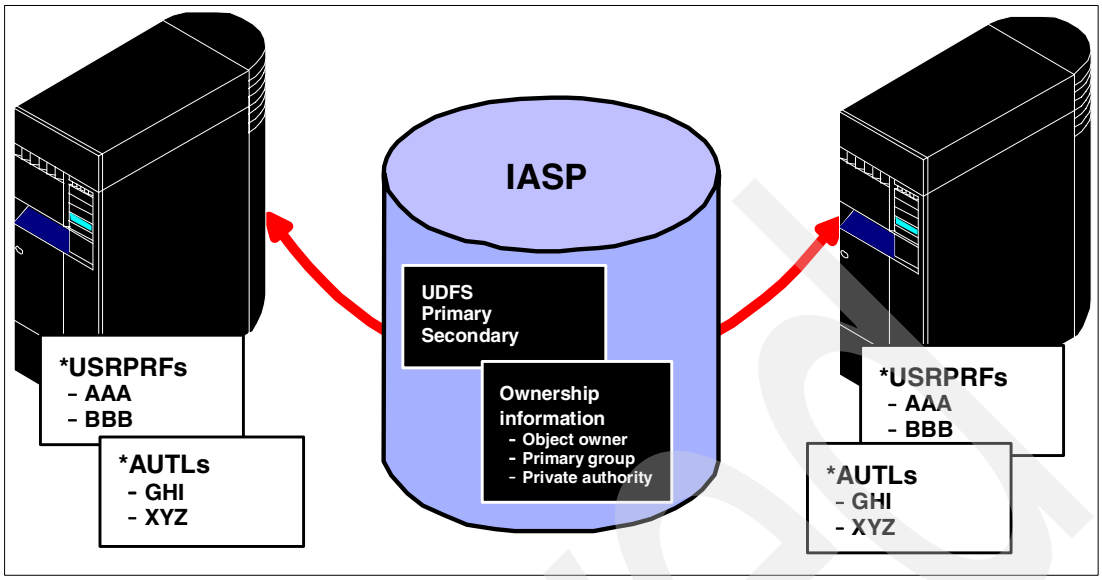


Figure 2-1 IASPs and user profiles

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 disk space within the IASP. The percentage varies and depends on the size and number of objects referenced by these structures.

For each user profile that owns or has private authority to an object in an IASP, the system stores information about the user profile in an internal structure called a *user profile extension*. The user profile extensions, while not visible nor accessible to users, are stored in the IASP.

An IASP may be varied on by the Vary Configuration (VRYCFG) command, an IASP failover, or being switched from another node. Anytime this happens, the system matches the user profile extension information in the IASP with a corresponding user profile object in the system ASP. This match is based solely on the name of the user profile.

If a user profile object does not exist with the name saved in a user profile extension, then the system creates a user profile object with the saved name. In this case, the user profile is created with the attributes listed in Table 2-1.

Table 2-1 User profile attributes when its object does not exist with the name of the extension

Keyword	Parameter description
USRPRF	Saved name
PASSWORD	*NONE
STATUS	*DISABLED
UID	Saved value or *GEN
GID	Saved value or *GEN
TEXT	Created by auto-configuration
All others	CRTUSRPRF command default values

In addition, during vary on of an IASP, the system verifies that the user ID number (UID) and group ID number (GID) values saved in the user profile extension are the same as the values in the matching user profile object. If these values do not match, the system performs recovery functions to ensure that the system is using a consistent set of UIDs and GIDs for all user profiles and for all objects on the system that use the UID or GID values.

The result of these system operations during the vary on of an independent disk pool means:

- If a user profile is deleted while an IASP is varied off or switched to another node, the user profile may reappear when the IASP is varied on.
- If the UID or GID values of a user profile are changed (including changing a user profile to be a group profile) while an IASP is varied off or switched to another node, then the UID or GID values associated with objects on the IASP may change during vary on processing.
- The time to vary on an IASP may be longer if user profiles are deleted or their UID or GID values changed.

2.6.2 Planning for user profiles and space to be used

If the Maximum Allowed Storage (MAXSTG) parameter in the user profile is set to *NOMAX, then no further action is required. The system automatically expands storage as necessary. If this parameter in the user profiles has a value in it, then you must make considerations for expansion of that size.

For each user profile on the system, an amount of storage can be set. If a value is set for that parameter, that same value can be used in each IASP as it is created or made available on the system.

If the system ASP has a large number of user profiles, or the system has a large number of objects, consider space for this storage when configuring the IASP.

2.6.3 Authorization lists (AUTL) and IASPs

Authorization lists are handled in a similar manner to user profiles. Authorization lists are only stored in the system ASP. The system maintains an “extension” on the IASP. *AUTL objects are matched to extensions while varying on. An *AUTL is created if one doesn’t exist.

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 and any IASPs.

There is only 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.

Figure 2-2 illustrates this concept. Authorization list names 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 authorization list name AUTL333 because the object LHR is not stored on the IASP.

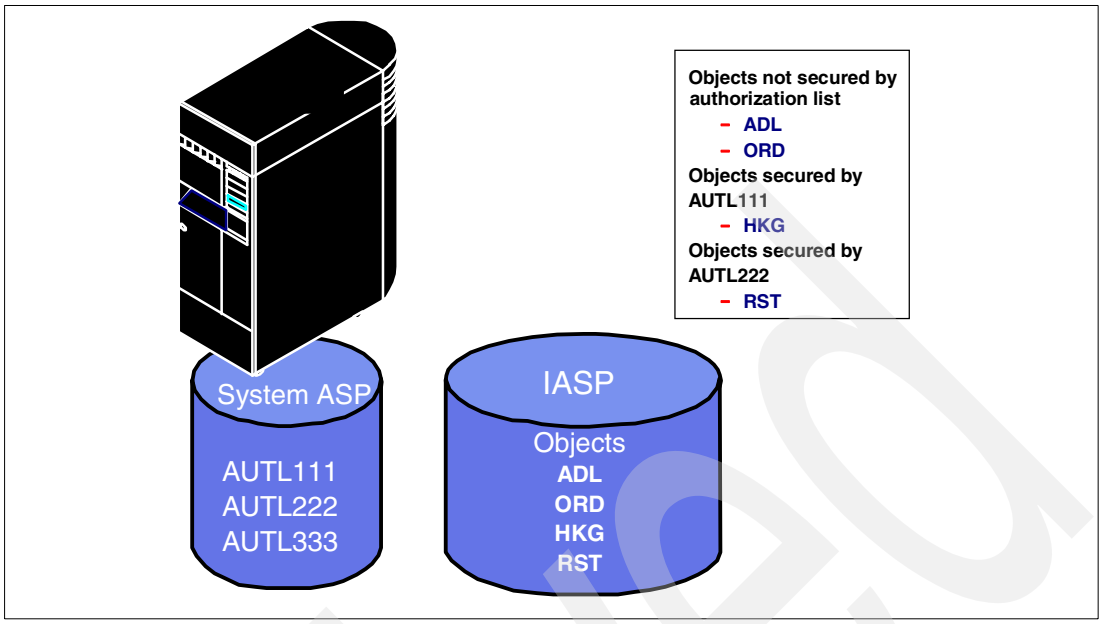


Figure 2-2 IASPs and authorization lists

As with user profiles, the System Licensed Internal Code (SLIC) 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.

2.6.4 Accounting for space used by user profiles and authorization lists

The total storage attributed to an individual user profile is the amount of the storage allowed in each online IASP and by the user in the local system and basic user ASPs. Use the Display User Profile (DSPUSRPRF) command to display the value of this storage. If the system has *n* IASPs, then the total storage that a user profile is allowed to use is the result of ((*n*+1) *MAXSTG).

To prevent disruptions to a system, the limit for the maximum amount of auxiliary storage that can be allocated by a user profile in an IASP is not enforced when bringing that IASP online. That is, if bringing an IASP online causes a user profile to exceed its storage limit for that IASP, the storage limit is allowed to exceed the specified amount. Subsequent requests for additional storage cause a "User Profile Storage Limit Exceeded" exception (message ID xxxx).

2.7 Capacity planning

When designing the use of IASPs, consider the items in 2.2, "Performance requirements" on page 10, as well as the total capacity required in each rack or ASP. Consider the amount of disk required, performance requirements, the amount of rack space required, switchability of the configuration, and the amount of floor space required.

Keep in mind previous discussions about temporary storage space. Refer to 2.2.2, "Disk drives: Arms versus capacity" on page 11.

Any application has some algorithm for growth attached to it. Whether it is carefully done with an actual specification, or guessed, there is some growth associated with most applications. This growth determines the amount of disk to be used.

All IASPs or disk pools within the rack are switched when the rack is switched. When the application in Pool33 is switched, and Pool34 exists on disk in the same rack, Pool34 (and its resident application) are also switched.

2.8 Hardware configuration and physical planning

The first step after any capacity planning session is to begin the sizing and configuration of a system to fulfill the requirements. When building the configuration, keep in mind and factor the following items into the planning.

2.8.1 Minimum hardware requirements

To create an IASP, you must first make some disk available. You can achieve this by removing existing disk units from the configuration or adding new disks into the configuration or system.

Attention: When removing disks from the existing configuration, you must exercise extreme caution. If you are working within an input/output processor (IOP) or IOA and within a RAID set, be careful not to create an environment that can result in reduced RAID protection or damage. Ideally you should use a minimum of all the disks under one IOA.

To move data from these disks and speed up the remove process, you can use the Disk Migrate While Active (DMWA) process. This is documented in *IBM @server iSeries Server Migration: System Migration and Upgrades at V5R1 and V5R2*, SG24-6055. After you move all possible data, then you can remove the disk from the configuration and make it available for the IASP use. DMWA cannot move data between IASPs.

If you are considering making the IASP switchable, you must consider these two ways:

- An LPAR environment, which is the simplest way to switch an IASP by IOP
- Between servers, where a whole expansion tower must be switched

An IASP installation requires the presence of non-configured disk units within an 0578, 0588, 5074, 5075, 5078, 5079, 5088, 5094, 5095, 5294, 8079, or 8094 tower. These disk units are formatted during the IASP definition process using iSeries Navigator.

2.8.2 Physical versus logical switching

When a tower is referred to as being *switchable*, it is meant in the logical sense of the word. The software in the operating system performs the switching of the resources within a tower from use by one system to use by another. There is no physical switching of cables. When the term *tower* is used in conjunction with switchable, it relates to the 5075 or 5074 expansion towers. The 5074 tower contains one PCI bus, plus cards, 45 possible disk drives, or the 5075 tower.

The 5079 is conceptually two 5074 towers, one on top of another, containing as many as 90 disk drives. They are housed in a single tower. Each of the two towers is switchable, independent of the other, unless the IASP is configured to span both towers.

2.8.3 PCI card switching

Placement of cards within the overall system configuration is critical. You must first consider that any PCI cards in the switchable entity, at the time the switchable entity is switched within the cluster, will be switched with the rack. Then you must consider that these cards become available to the switched-to system when switching is complete.

2.8.4 ASPs and expansion towers

Keep in mind that you can configure multiple IASPs on an IOP or in a tower, and everything in the tower switches when the switchover is performed. This may require multiple towers, with fewer disks in each, to meet all requirements. Or this may require one tower with spare capacity.

2.8.5 High-speed link (HSL) cable placement

HSL cables are often shown in a loop. Figure 2-3 shows two systems in a cluster and in a device domain. One switchable tower (a 5075) is owned by the Model 270.

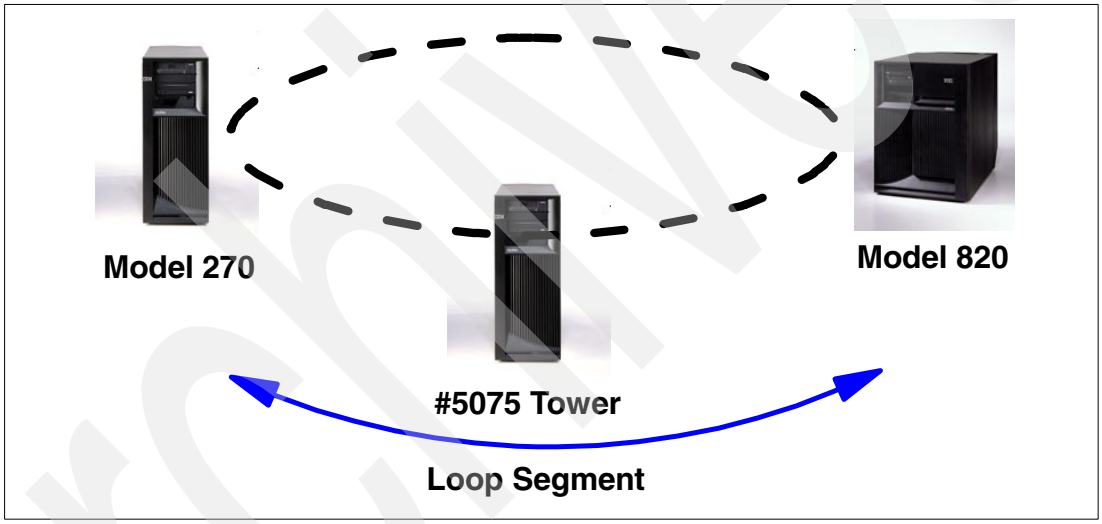


Figure 2-3 Switchable tower in an HSL loop

Information only moves down one segment at a time. In a loop, the second segment is redundant and is used in the event the first leg fails.

The HSL cables come in varying lengths, and there are limits to their length. In addition, certain types of equipment attached via HSL cables require cabling to specific HSL ports to optimize throughput. Be aware of this when building the configuration and designing the layout of the computer room.

2.8.6 HSL port feature

HSL connectivity is required for a multiple system switchable implementation of an IASP. HSL ports must be of the correct feature code to support this functionality at V5R1 and beyond. Table 2-2 shows the compatibility between IASPs and HSLs.

Table 2-2 IASP and HSL compatibility

HSL feature	Description	Model/ feature	Processor/ tower	Release implemented	IASP capable
9691	Base Bus Adapter	5074, 5078, 5079, 0578	T	V4R5	Y
2739	Optical Bus Adapter	5074, 5078, 5079, 0578	T	V5R1	Y
2887	HSL-2 Bus Adapter	5094, 5095, 5088, 0588	T	V5R2	Y
9732	HSL Ports - 8 Copper	830/2400	P	V4R5	N
2777	HSL Ports - 8 Copper	830/2400	P	V5R1	Y
2774	HSL Ports - 2 Opt/6 Copper	830/2400	P	V5R1	Y
9733	HSL Ports - 8 Copper	830/ 2402, 2403	P	V4R5	N
2754	HSL Ports - 8 Copper	830/ 2402, 2403	P	V5R1	Y
2758	HSL Ports - 2 Opt/6 Copper	830/2402, 2403	P	V5R1	Y
9737	HSL Ports - 16 Copper	840	P	V4R5	N
2755	HSL Ports - 16 Copper	840	P	V5R1	Y
2759	HSL Ports - 4 Opt/12 Copper	840	P	V5R1	Y
2776	HSL-2 Ports - 8 Copper	890	P	V5R2	Y
2788	HSL-2 Ports - 8 Optical	890	P	V5R2	Y
9730	HSL-2 Ports - 2 Copper	890	P	V5R2	Y
9887	HSL-2 Bus Adapter	9094		V5R2	
2785/9785	HSL-2 Ports Copper	52M/0155, 0156, 0157, 2472, 2473, 2474, 2484	P	V5R2	Y
2786/9786	HSL-2 Ports Optical	52M/0155, 0156, 0157, 2472, 2473, 2474, 2484	P	V5R2	Y
2776	HSL-2 Ports - 8 Copper	52H	P	V5R2	Y
2788	HSL-2 Ports - 8 Optical	52H	P	V5R2	Y

2.8.7 HSL cabling rules for multiple system switched tower implementation

General HSL cabling requirements for a number of towers, inclusive of Integrated xSeries Adapter towers, are model dependent. They are specified in *IBM @server iSeries and AS/400e™ System Builder*, SG24-2155, for V5R2. IASP switchable towers also have special requirements. Although the towers are model dependent, the following requirements apply:

- You may place a maximum of two cluster nodes (systems) on a single HSL loop. Keep in mind that there can be 32 LPARs per system. Therefore, there can be 64 nodes on an HSL loop.
- When an HSL loop contains more than two nodes (systems), towers are not supported.
- You may place a maximum of four switchable external towers on a single HSL loop, with up to three of those towers on a single segment of the loop.
- A switched tower in an HSL loop segment must be adjacent to the alternate system or to another tower owned or defined as switchable to the alternate system.
- All switchable towers on one loop segment must be in the same system power control network (SPCN) power domain (attached to the same system).
- When other switchable towers are on another loop segment, and the disk pools on those towers are in the same device cluster resource group (CRG) as other disk pools on a different loop segment, all the switchable towers on both loop segments must be in the same SPCN power domain.

2.8.8 SPCN cable considerations

SPCN cables chain from the main tower or its expansion tower to each device consecutively. Unlike HSL cables, the last device on the chain does not loop back to the source. SPCN cables are used to control the powering sequence of the hardware, among other things.

SPCN control for a switchable tower remains with the source system. If the tower is switched and the target system is powered off, the switched tower remains powered on. This situation requires care since there is still main voltage in the switched tower, even though the system appears to be powered off.

You cannot switch disk drives that are attached through SPD IOPs, in migration towers, between nodes on a clustered physical systems. However, you can switch them between LPARs. You can also create them as independent disk pools.

2.9 Aesthetics

Aesthetics are usually the last consideration when placing an order and configuration for a system. However, building a drawing of the desired configuration, especially the towers and racks involved, can often save costly errors when you assemble the configuration upon delivery, especially in the area of cables. Design it one way, assemble it another way, and the end result may require additional cables.

Planning for the “showcase” look at this end of the cycle can save you from a lot of headaches later. In some companies, the need to showcase the computer literacy side of the company may require the placement of racks and towers so tours, for example, won’t interfere with day-to-day operations. For example, you may need to consider:

- Primary rack placement
- Secondary rack placement
- Additional rack placement
- Tape rack placement
- Console placement

2.10 Physical planning requirements

This section outlines the physical planning requirements and considerations for IASPs:

- For placement and cabling assistance, see the *HSL and SPCN cable planning guide* at:

  http://www.as400.ibm.com/tstudio/planning/hslguide.htm

- The time to vary on an IASP during the switching process depends on the number of objects on the *SYSBAS, and not the size of the database objects. If possible, keep the number of objects small.
- For a quicker vary on or off, keep the UID and 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.

  Important: Having different UIDs between systems lengthens the vary-on time substantially because UIDs that don't exist must be created.

- All the disk units within a tower are switched as an entity. When a tower containing the IASP is switched, all other IOPs and 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.

2.10.1 Advantages of using IASPs

An additional method of iSeries availability is enabled through the use of IASPs. This section lists some of the advantages of using IASPs:

- 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 needing to buy a duplicate set of disks for the backup system.
- The contents of a switchable IASP can be made available to the backup system without any kind of replication or extra replication software, provided some objects are already available on the target system.
- 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 IASPs.

  Replication requires more CPU cycles when replicating to a backup system.

- No network traffic is associated with IASPs.

  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 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, journal entries can become “trapped” on the source system at the time of failure and do not arrive at the target machine.

- When used in conjunction with journaling, IASPs can be used as archives, or save/restore repositories, without needing high availability business partner (HABP) software.

2.10.2 Limitations of using IASP

This section list some limitations of using independent disk pools:

- In the same fashion as disks in *SYSBAS, independent disk pools represent a single point of failure in the system.

  If the disks in the independent disk pool are permanently damaged and the data is unrecoverable, data is available only up to the last backup copy. Independent disk pools protect the iSeries server against system failures, but not against disk failure.

- Because of loop limitations with HSL, the systems must be within 250 meters of each other with fiber HSL and 15 meters with copper HSL. When migrating from PCI to PCI-X systems and tower, there may be a change in the cable length from 15m to 10m. Plan carefully.

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

- If the IASP configuration involves an HSL loop, a V5R1 supported HSL adapter is required.

  HSL adapters prior to V5R1 do not work with switchable IASPs. However, systems with original HSL cabling can be upgraded to newer HSL features.

- The independent disk pool works with only one system at any one time. The independent disk pool cannot be used for balancing workload, other than switching to another, lower utilized, system.

  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.

2.11 Independent disk pool planning checklist

The “Independent disk pool checklist” on page 182 is presented as a means of approaching the design and implementation of IASPs from an organized perspective. It is not meant to be definitive.

Each implementation is unique to the environment and applications that run in it. Be sure to add the unique aspects of your design and implementation as necessary.

2.12 Disk planning worksheet

The “Disk planning worksheet” on page 184 helps you to plan for installing disk drives and the IASP in which they are to be installed. You can use this worksheet in conjunction with the printing capabilities of Management Central to create a plan and disaster recovery documentation.

This worksheet is based on a 5074 PCI Expansion Tower. Most other frames can be made to fit within this example by renumbering the slots. The pre-numbering is the sequence the disk should be installed for optimum performance.

2.13 PCI card placement planning worksheet

The “PCI card placement planning worksheet” on page 186 helps you to plan for installing PCI cards and the IASP in which they are to be installed. You can use it in conjunction with the printing capabilities of Management Central to create a plan and disaster recovery documentation.

This worksheet is based on a 5074 PCI Expansion Tower. Most other frames can be made to fit within this example by renumbering the slots.

2.14 Positioning independent disk pools

Considering the introduction of independent disk pools on the iSeries server, you may ask:

- Are HABP 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 2-3 to compare the needs of the business to the functions available with a HABP replication and that are offered by independent disk pool switched disks.

Table 2-3 Comparison of HABP replication and independent disk pool switched disk

	HABP replication solution	**Independent disk pool 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	Apply lag + replication failover overhead application restart = 5 - 30 minutes	IASP vary on + application failover restart = ~15 minutes
Typical switchover time	Apply lag + replication switchover overhead application restart = ~5 minutes	IASP vary on + application switchover restart = ~5 minutes
Real time coverage	Objects journaled	Objects in IASP

	HABP replication solution	Independent disk pool switched disk solution
Objects supported	A very large set	IFS and QSYSLIB with some exceptions
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 (may include Cluster Resource Services)	Requires Cluster Resource Services support on the two systems
Note: Actual times may vary widely depending on configurations, type of failover, and applications or objects involved.		

While it is true that IASPs provide a new means of high availability on the iSeries server, it does not replace HABP solutions. Independent ASPs co-exist with HABP solutions.

Consider these characteristics of IASP and HABP solutions:

- HABP 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 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.
- An HABP solution provides switchover capability between two systems.

  The HABP level of monitoring between systems can be primitive. However, the HABP 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. After it is properly setup, the switchover or failover to the backup system can be nearly seamless.

3

Configuration examples

Independent auxiliary storage pools (IASPs) allow you to configure multiple disk storage units into one group. The group can include:

- Non-switchable IASP:
  - Any or all units in the system tower, except the load source
  - Any one or more disks anywhere on the system
- Switchable IASP (all disk units in a tower other than the system tower):
  - All the units in one tower
  - A subset of the disk units in a tower
  - Multiple IASPs in a tower
  - Disk units from two or more I/O towers

To illustrate this concept, this section describes six possible IASP configurations.

Important: In all cases where the term *switchable* is used, keep in mind that, when the disk is switched, everything in the tower is switched. This includes all PCI cards in the tower and other disks in the tower.

3.1 Configuring IASPs

All devices packaged in a single I/O tower are switched from one iSeries server to another when configured in a cluster. The non-ASP devices, such as workstation controllers, Ethernet cards, 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, such as varying on the non-disk devices.

3.2 Non-switchable IASPs

Non-switchable IASPs are available in any system that supports V5R2. This provides a great deal of flexibility for testing and implementing an IASP.

3.2.1 Any one or more disks anywhere on the system

Figure 3-1 shows that any one or more disks anywhere on the system can be created as an IASP. This may be used where a separate database is required. However, the ability to switch isn't necessary.

This is not ideal because it does not cater to disk failure. If you create two IASPs with this number of disks, you must rearrange your storage and free RAID set disks under input/output processors (IOPs).

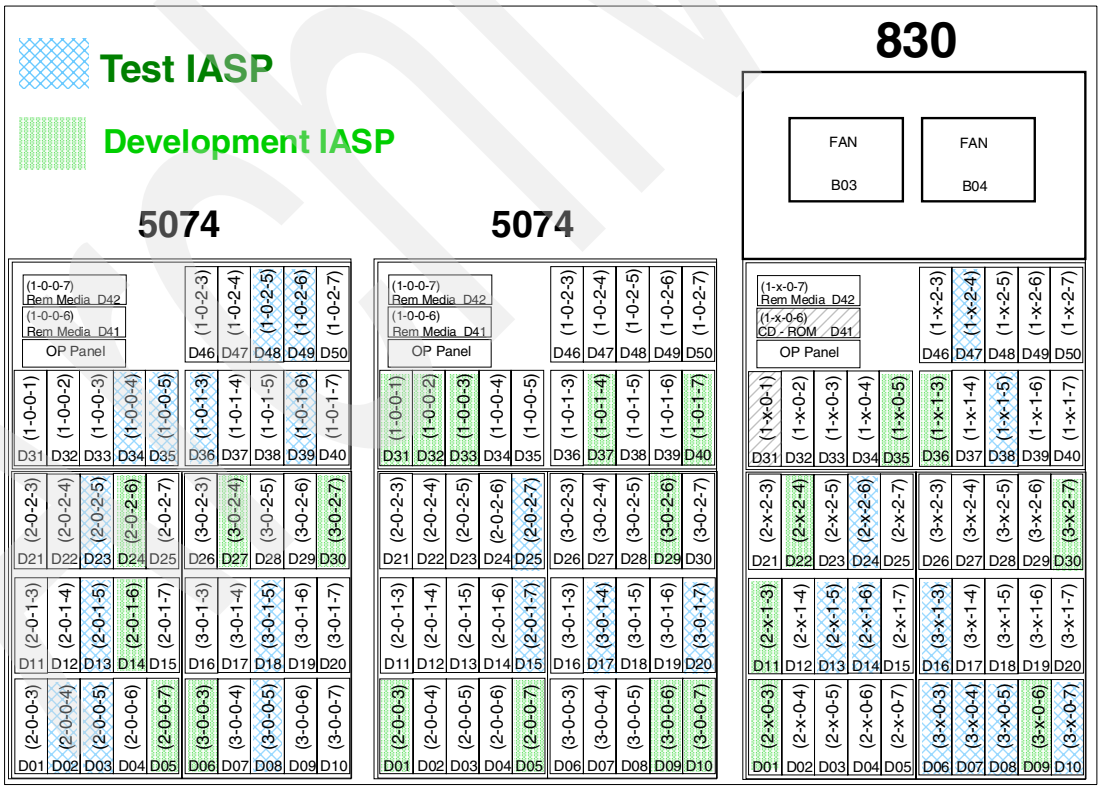


Figure 3-1 Any one or more disk drives

3.3 Switchable IASPs

Switchable IASPs apply to both logically partitioned and non-logically partitioned systems. The switchable resource can be as small as an IOP in a logically partitioned environment. However, the smallest switchable unit in a non-logically partitioned environment is an expansion tower.

3.3.1 Complete towers and IASPs

The example in Figure 3-2 shows two IASPs configured in three towers. One IASP is for the the tower that is dedicated to a Canadian application. The other IASP spans two towers that hold the USA application.

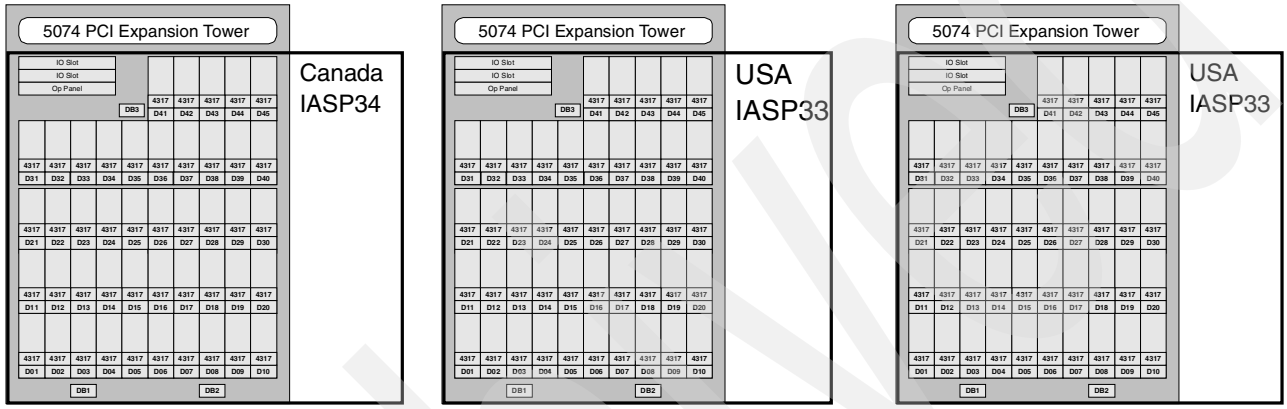


Figure 3-2 All units in the tower and IASP

3.3.2 Subset of the disk units in a tower

The example shown in Figure 3-3 illustrates an IASP configuration with a single IASP in a tower. How the rest of the slots are used is a matter of choice. If the tower or IASP is used in switched configuration, any IASPs other than IASP51 in the same tower must be switchable.

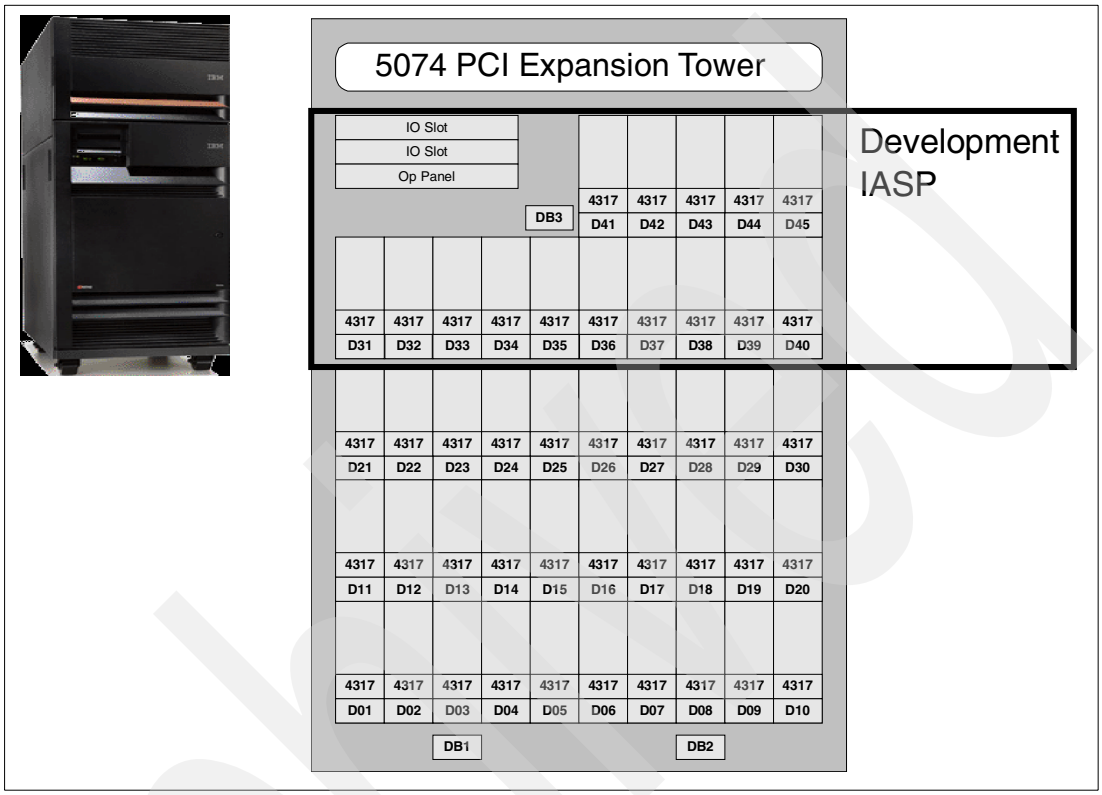


Figure 3-3 Subset of units in a disk tower as an IASP

3.3.3 Multiple IASPs in a tower

The example in Figure 3-4 illustrates an IASP configuration with two IASPs in the same tower. IASPs named ABC Production and XYZ Production 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. In this case, both of the application databases in the tower can be switched together. You must apply application priority to the IASPs in relation to their particular business priorities.

Remember in this case, the IOPs and input/output adapters (IOAs) also switch with the tower.

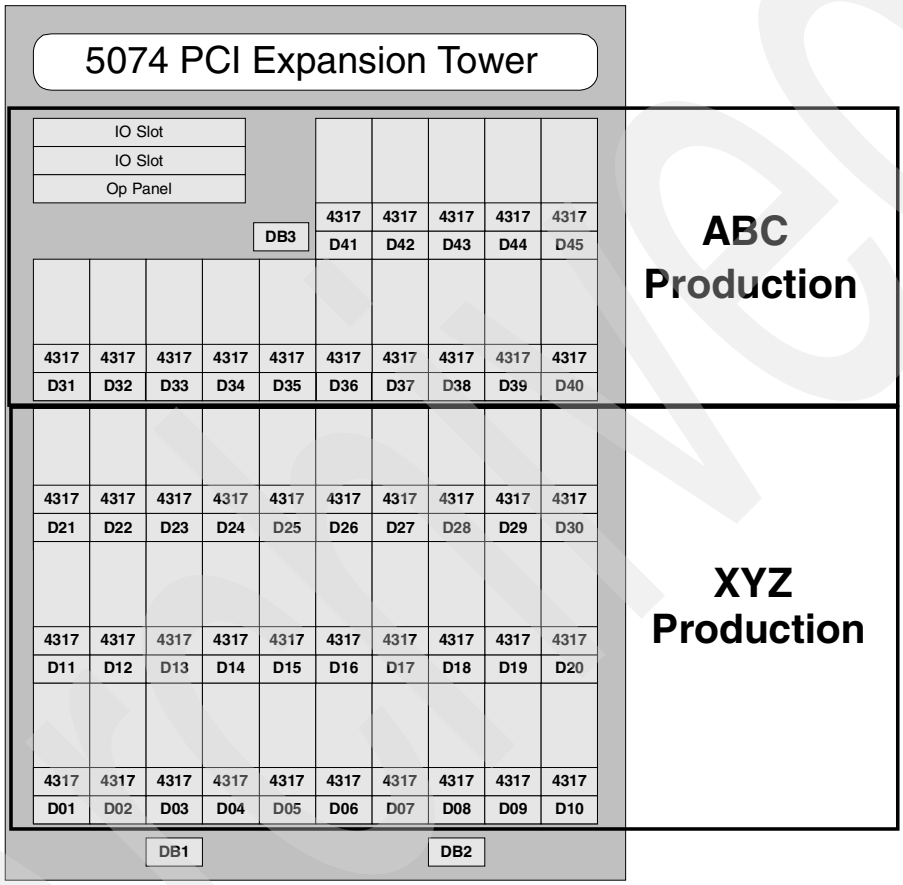


Figure 3-4 Multiple IASPs in a tower

3.3.4 Disk units from two or more I/O towers

Figure 3-5 shows how an IASP can span towers. In this example, there are three IASPs: development, payroll, and production.

Practically speaking, the production IASP does not represent an optimum configuration. This is because when production is switched, payroll and development must also switch. A better configuration is for production to be fully contained within one of the towers. Then development and payroll can share the second tower.

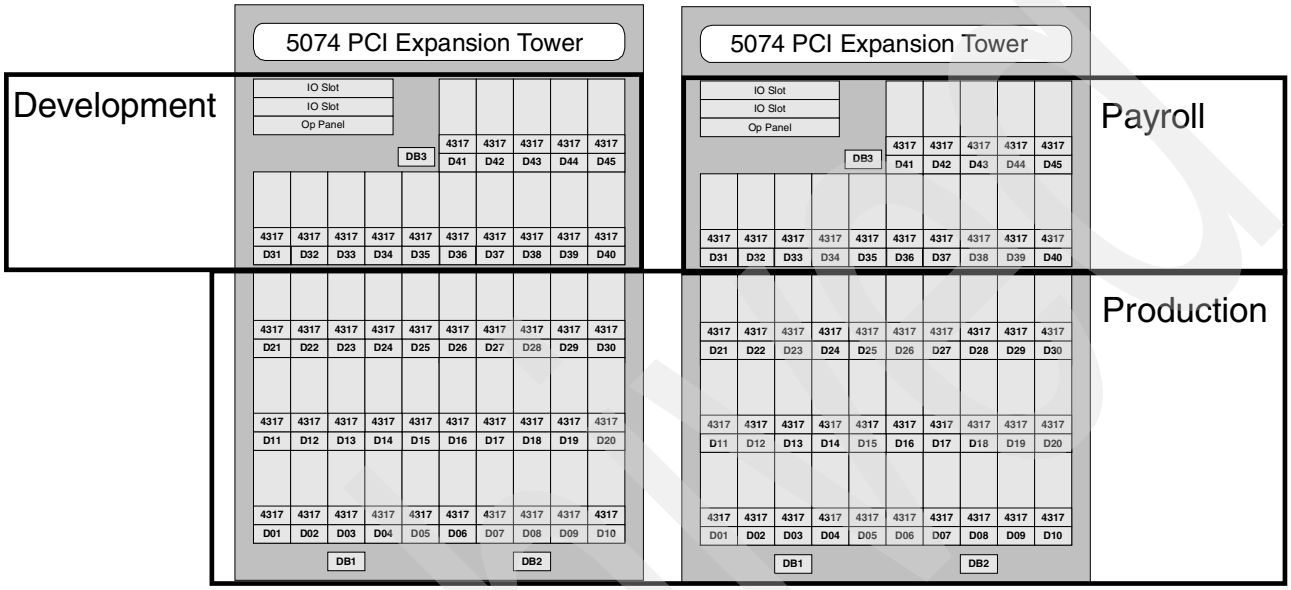


Figure 3-5 Disk units from two or more towers in an IASP

The #5079 tower is a hardware package that behaves the same as a #5074 tower, but contains twice the disk units and twice the PCI card slots. In terms of switching, each half of the 5079 can be switched independently of the other. Remember to cable it correctly if you want independent switching.

4

Application considerations

This chapter addresses considerations for the relational database as it applies to independent disk pools, as well as any programming issues that may arise as a result. It begins with an example of multiple relational databases (RDBs) on a system. Then it explains the implications of addressing these multiple RDBs in both a traditional and Structured Query Language (SQL) programming environment.

4.1 Name space, independent disk pools, and RDBs

Prior to the introduction of library-capable independent auxiliary storage pools (IASPs), any thread, including the primary or only thread for a job, could reference the following libraries by name:

- The QTEMP library for the thread's job, but not the QTEMP library of any other job
- All libraries within the system ASP
- All libraries within all existing basic user ASPs

This set of libraries formed the library *name space* for the thread and was the only possible component of that name space. Although there was not a formal term for this name space component, it is now referred to as the *SYSBAS component of the name space. It is a required component of every name space.

With library-capable IASPs, a thread can reference, by name, all of the libraries in the IASPs of one ASP group. This adds a second, but optional, component to the name space and is referred to as the *ASP group component* of the name space. A thread that does not have an ASP group component in its name space has its library references limited to the *SYSBAS component. A thread with an ASP group component to its library name space can reference libraries in both the *SYSBAS and the ASP group components of its name space.

With V5R2 and library-capable IASPs, the definition of a name space has become *SYSBAS and zero or one ASP group. A system that has more than one ASP group can simultaneously have some active threads with only a *SYSBAS component to their name space, while other threads have both a *SYSBAS and an ASP group component. It is not necessary that the ASP group component be the same for all threads.

Library names no longer must be unique on a system. However, to avoid ambiguity in name references, library names must be unique within every possible name space. Since *SYSBAS is a component of every name space, presence of a library name in *SYSBAS precludes its use within any IASP. Since all libraries in all IASPs of an ASP group are part of a name space, for which the ASP group is a component, existence of a library name within one IASP of an ASP group precludes its use within any other IASP of the same ASP group. Because a name space can have only one ASP group component, a library name that is not used in *SYSBAS can be used in any or all ASP groups.

OS/400 has a file interface and an SQL interface to its databases. The file interface uses the name space to locate database objects. For compatibility, SQL maintains a catalog for each ASP group. This catalog resides in the primary IASP of the ASP group. The catalog is built from the objects that are in a name space that has the ASP group and *SYSBAS as its two components. The names database and the name space are somewhat interchangeable because they refer to the same set of database objects.

Each name space is treated as a separate relational database by SQL. It is required that all RDBs whose data is accessible by SQL are defined in the RDB directory on the system.

Note that the name space is a thread attribute and can be specified when a job is started. When it is referenced as a *job attribute*, it technically means the "thread attribute for the initial thread of a single-threaded job".

The example in Figure 4-1 illustrates the possible ways to set up direct access storage device (DASD) in various IASP configurations.

If we assign unique database names to each IASP (instead of using the default IASP names for the RDB name), we get a much more meaningful RDB schema.

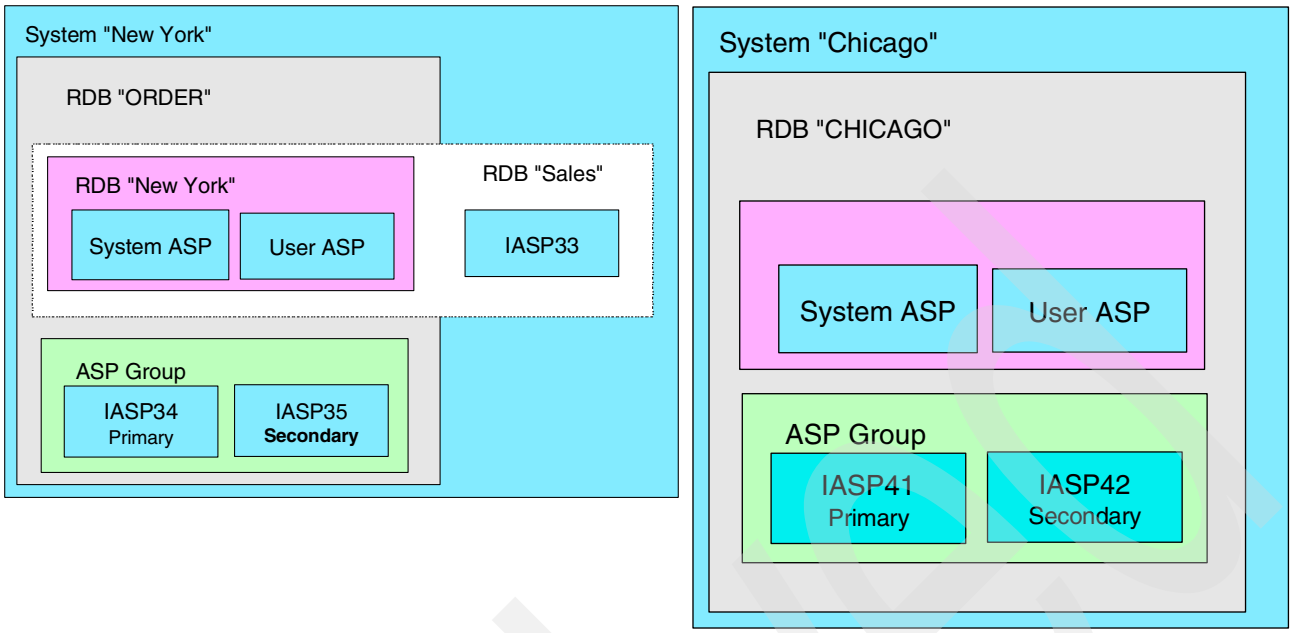


Figure 4-1 RDB configuration examples

Figure 4-2 shows the RDB entries on the system.

```
                Work with Relational Database Directory Entries

Position to  . . . . . .

Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display details   6=Print details

       Relational          Remote
Option Database            Location                  Text

  ___  NEWYORK             *LOCAL                    System/User ASP
  ___  SALES               129.168.1.30              IASP33
  ___  ORDER               ORDER                     IASP34 & 35
  ___  CHICAGO             CHICAGO                   CHICAGO
  ___  CUSTOMER            CUSTOMER                  CHICAGO: IASP41 & 42
```

Figure 4-2 WRKRDBDIRE on system NEWYORK

Figure 4-3 shows the external organization of the databases. This provides a platform for discussing the issues surrounding applications using IASPs in V5R2.

The two systems in Figure 4-3 include five RDBs. *System New York* is configured with:

- The system ASP and a user ASP (together they form *SYSBAS)
- Three IASPs (IASP33, IASP34, and IASP35), two of which comprise an ASP group (IASP34 and IAS35)

On this system, RDB NEWYORK includes the system ASP and the user ASP; RDB SALES includes the system ASP, the user ASP, and IASP33; and RDB ORDER includes the system ASP, the user ASP, IASP34, and IASP35.

System Chicago is configured with:

- A system ASP
- A user ASP
- Two IASPs (IASP41 and IASP42) in an ASP group

On this system, RDB Chicago includes the system ASP and the user ASP. RDB Customer includes the system ASP, the user ASP, and the ASP group consisting of IASP41 and IASP42.

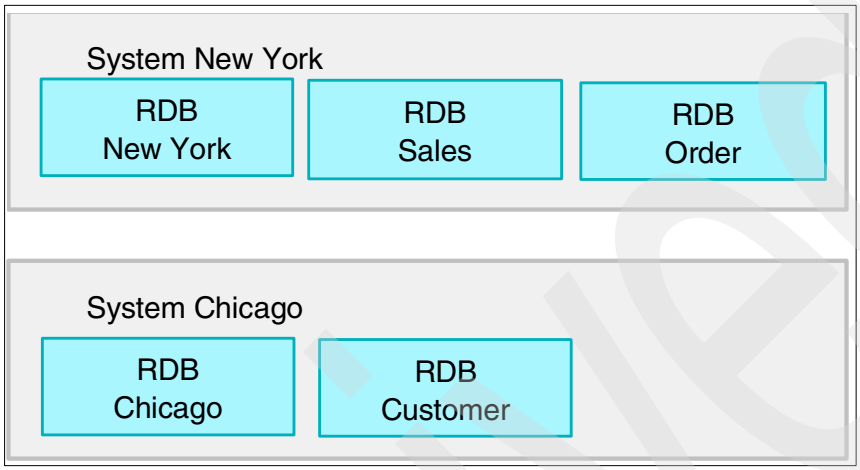


Figure 4-3 External organization of databases

4.2 Relational database directory

The OS/400 program uses the RDB directory to define the RDB names that can be accessed by applications running on an iSeries server. OS/400 also uses the RDB directory to specify whether the connection uses Systems Network Architecture (SNA) or Internet Protocol (IP), and to associate the RDB names with their corresponding network parameters.

The RDB directory allows an application requester (AR) to accept an RDB name from the application and translate this name into the appropriate IP address or host name and port. Or it can translate it into the appropriate SNA network identifier and logical unit (LU) name values for communications processing. As of V5R2, the RDB directory can also specify the user's preferred outbound connection security mechanism. The relational database directory can also associate an Application Requester Driver (ARD) program with an RDB name.

Each iSeries server in the distributed relational database network must have a relational database directory configured. There is only one relational database directory on a system. Each AR in the distributed relational database network must have an entry in its relational database directory for its local RDB and one for each remote and local user RDB that the AR accesses. Any system in the distributed RDB network that acts only as an application server does not need to include the RDB names of other remote RDBs in its directory.

The RDB name assigned to the local RDB must be unique from any other RDB in the network. Names assigned to other RDBs in the directory identify remote RDBs or local user databases. The names of remote RDBs must match the name that an AS uses to identify its local system database or one of its user databases, if configured. If the local system RDB name entry for an application server does not exist when it is needed, one is created automatically in the directory. The name used is the current system name displayed by the Display Network Attributes (DSPNETA) command.

4.3 Distinctions

The distinctions between a configuration of an IASP and the RDB created when setting up an IASP are subtle yet powerful. Besides the obvious benefits this additional functionality brings to the overall effectiveness of OS/400, there is the potential for complications. Listed here are statements that when considered make sense, but are not immediately obvious to the uninitiated.

The distinction between CRTLIB and CRT COLLECTION is that CRTLIB defaults to ASP 1, the system ASP (*SYSBAS). CRT COLLECTION defaults to the current library name space (RDB).

Using the RESTORE or RSTLIB command runs the CRTLIB command under the covers (unless the library already exists). As stated earlier, by default, this is done to ASP 1 unless specifically qualified beforehand with the Set Auxiliary Storage Pool Group (SETASPGRP) command.

4.4 IASP as a separate database

In V5R1, the independent ASP was not assigned a separate database name. OS/400 supported only a single database, which usually defaulted to the name of the system. This was set up using the Add Relational Database Directory Entry (ADDRDBDIRE) command. You could work with the database by using the Work with Relational Database Directory Entries (WRKRDBDIRE) command.

In V5R2, each IASP is given a database name. The IASP name is used if none is assigned when the IASP is created. This is done during the creation of an IASP (disk pool). It is actually a name space that is treated as a separate RDB by SQL. A name space, and therefore an RDB, may consist of the system ASP, user ASPs, and an ASP group. The ASP group may, in turn, consist of a primary and zero or more secondary IASPs.

The user may create additional RDBs on an iSeries server by configuring IASPs on the system. Each *primary* IASP is an RDB. It consists of all the database objects that exist on the IASP disks. Additionally, all database objects in the system RDB of the iSeries server, to which the IASP is connected, are logically included in a user RDB. Therefore, the name of any schema created in a user RDB must not already exist in that user RDB or in the associated system RDB.

Although the objects in the system RDB are logically included in a user RDB, certain dependencies between the objects in the system RDB and the user RDB *are not* allowed. These include:

- Creating a view into a schema that exists in the same RDB as its referenced tables, views, or functions.
- Creating an index into a schema that exists in the same RDB as its referenced table.
- Creating a trigger or constraint into a schema that exists in the same RDB as its base table.
- Ensuring that the parent table and dependent table in a referential constraint both exist in the same RDB.
- Creating a table into a schema that exists in the same RDB as any referenced distinct types.
- Ensuring that the parent table and dependent table in a referential constraint both exist in the same RDB.

Other dependencies between the objects in the system RDB and the user RDB are allowed. For example, a procedure in a schema in a user RDB may reference objects in the system RDB. However, operations on such an object may fail if the other RDB is not available, such as when a user RDB is varied off and then varied on to another system. A user RDB is local to an iSeries server, while the IASP is varied on. IASPs can be varied off on one iSeries server and then varied on to another iSeries server. Therefore, a user RDB may be local to a given iSeries server at one point in time and remote at a different point in time.

4.4.1 Connections

In an SQL environment, SQL CONNECT is used to specify the correct database. To achieve best performance, make sure the database being connected to corresponds with your current library name space. You can use the SETASPGRP command to achieve this. If the SQL CONNECT function is not operating within the same library name space, the application uses Distributed Relational Database Architecture™ (DRDA®) support, which can affect performance.

There are two types of connections: application connections and system connections.

Application connections

Application connections are established based on the following rules:

- These connections are similar to the types of connections supported in previous releases. Under this type of connection, only one local connection per activation group is allowed at a time. Any other connections use DRDA.
- Distributed connection rules for both Distributed Unit of Work (DUW) and Remote Unit of Work (RUW) are applied equally to all RDBs and to implicit and explicit connections. The rules include:
  - If RUW is used, a connection request fails with the SQL0752 "`Connection cannot be changed`" error message if there are pending transactions.
  - If RUW is used, only one active connection is allowed. All the resources associated with the current connection are released before a new connection is started. For example, all open cursors are closed, all prepared statements are destroyed, and all normal SQL locks are released.
- The CONNECT statement to a local RDB is processed as follows:
  - For RUW, the local connection is started if the target RDB corresponds to the current name space. Otherwise, DRDA is started.
  - For DUW, the local connection is started if the target RDB corresponds to the current name space and there are no existing local connections. Otherwise, DRDA is started.
- If the first SQL statement in the activation group is not CONNECT, an implicit connection occurs. For a local program, SQL starts the local connection to the current name space (RDB). For a distributed program, SQL starts the DRDA connection to the RDB specified in the RDB parameter at compile or SQL package creation time.
- If the activation group is already connected to an IASP, the first SQL statement following a name space switch to an IASP that is different from the current connection must be CONNECT or SET CONNECTION. Otherwise the SQL statement causes the SQL0752 "`Connection cannot be changed`" error message with reason code 9.

4.4.2 System connections

System connections are new to V5R2. They are limited to use by the operating system. Under the connection, one local connection is supported per job at a time. The system connection may be used along with application connections at the same time.

4.4.3 Switching RDBs

The RDB (name space) can be switched using the SETASPGRP CL command. An RDB cannot be switched by simply using the CONNECT statement with a new database name. You must use one of the three methods prior to using the CONNECT statement.

RDB switches are not allowed in stored procedures, user-defined functions, or triggers.

4.4.4 Object creation

Objects that are tightly coupled to other objects must be created in the same IASP. Notice that this is not in the same name space, but the same IASP.

SQL objects do not allow spanning across IASP boundaries. Spanning between primary and secondary IASPs in an ASP group *is* allowed. Spanning between *SYSBAS and an IASP *is not* allowed. For example, applications cannot create a view across libraries in both *SYSBAS and an IASP.

Applications that use commitment control may not update objects in *SYSBAS and the IASP within the same scope of work or within the same connection. Often the application programmer may be unaware that there are objects in *SYSBAS. These may be control tables or fields in some product library. Or they may be views or indices where the name was not qualified.

While it is possible to create files, tables, etc. into QSYS2, the corresponding library in the independent disk pool prevents this from occurring. Most applications that create data in QSYS2 don't realize it and fail when running in an independent disk pool.

Consider the following example:

```
CHGCURLIB DEMO10
create view ICTABLES(Owner, tabname, type) as select table_schema, TABLE_NAME, TABLE_TYPE
from SYSTABLES where table_name like'IC%'
```

In this example, the view ICTABLES is not built in the current library (DEMO10) as you would expect. It is built in the library of the first table that is mentioned, which is QSYS2. It fails when accessing the independent disk pool because creation of objects in QSYS2XXXXX is prevented.

4.4.5 System-wide statement cache (SWSC)

A separate SWSC is created and maintained on each IASP. Multiple sets of system cross-reference and SQL catalog tables are defined and maintained on each IASP.

The IASP version of QSYS and QSYS2 contain cross-reference and SQL catalog tables with "merged views" of all the SQL and database objects that are accessible when connected to the IASP.

4.4.6 Start Query Management Query (STRQMQRY) and Start Query Management Procedure (STRQMPRC) RDB support

You can resolve the SQL objects (tables, functions, views, types) that are referenced in a Query Management Query (*QMQRY) object. To do this, you use the RDB specified on the RDB parameter or the RDB specified on the CONNECT/SET CONNECTION commands. This RDB may be an IASP. The query management objects referenced must be in the current RDB (name space).

When output from a STRQQRY command is directed to an output file, Query Management ensures that the output file is created on the RDB (name space) that was current at the time the SRQMQRY is executed. If the RDB is an IASP, on V5R2, printed output is in the system ASP, under control of spooling.

4.5 ODBC considerations

The ODBC connection defaults to the database specified in the job description of the user profile. You can override it. Simply select the **Override default database with the following** option and enter the name of the database to be overridden. See Figure 4-4.

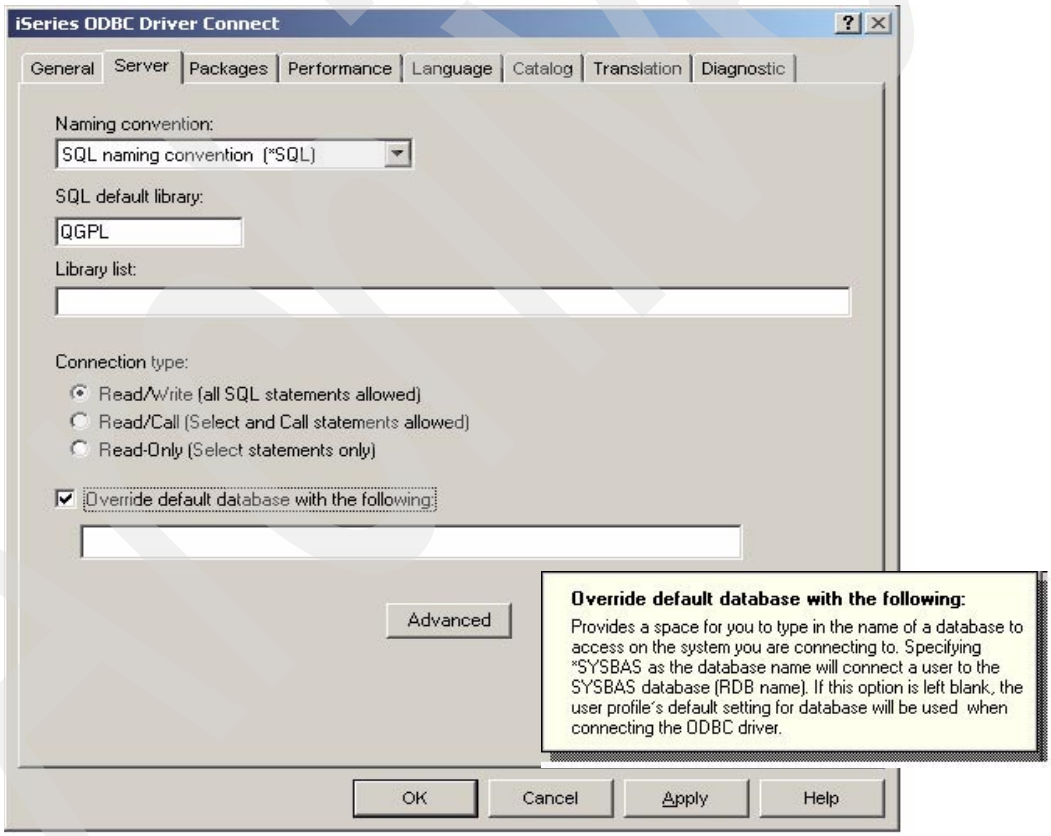


Figure 4-4 iSeries ODBC Driver Connect window

4.5.1 JDBC considerations

You may need to review access to IASPs using the IBM Toolbox for Java JDBC driver. You should set the connection property “DatabaseName” to the new database name as required by the application. Keep in mind that the disk pool name is the default name of the database unless it is changed.

The JDBC connection requires the database name. The disk pool name and the database name may be different. The JDBC driver does not use DRDA for SQL access to iSeries databases.

4.6 Systems-managed access-path protection (SMAPP) considerations

System-managed access-path protection can be a significant part of a recovery situation. Moving the target recovery time to as short as possible improves the failover times in the same way it improves abnormal initial program load (IPL) recovery times.

4.7 Query/400

Query/400 only references objects in the current RDB (name space). A *QRYDFN object created in the system ASP may reference files in an IASP and vice versa. If a *QRYDFN object created to reference objects in an IASP runs when a different IASP is set as the current RDB (name space), the *QRYDFN runs successfully if the new IASP contains objects with the same name and the file formats are compatible.

4.8 System values

Before you implement independent disk pools, examine how you use the following system values. System values have no access to the SETASPGRP command. In most cases, the programs they reference as their values must exist in *SYSBAS. The system values that are affected by an implementation of independent disk pools are:

- **QALWUSRDMN**: Allow user domain objects in libraries

  This value specifies which libraries may contain user domain user (*USRxxx) objects. You can specify up to 50 individual libraries or all libraries on the system.

- **QATNPGM**: Attention program

  This value specifies the name and library of the attention program. This program must exist in the system ASP or in a basic user ASP.

- **QBOOKPATH**: Book and bookshelf search path

  Regardless of how you set this value, it does not affect the operating system. The operating system no longer uses this system value. However, your application programs may use it.

- **QCFGMSGQ**: Configuration message queue

  This system value allows you to specify the default message queue the system uses when sending messages for lines, controllers, and devices. The message queue must exist in the system ASP or in a basic user ASP.

- **QCTLSBSD**: Controlling subsystem

  The controlling subsystem is the first subsystem to start after an IPL. One subsystem must be active while the system is running. This is the controlling subsystem. Other subsystems can be started and stopped. If this subsystem description cannot be used (for example, it is damaged), the backup subsystem description QSYSSBSD in the library QSYS can be used. A subsystem description specified as the controlling subsystem cannot be deleted or renamed after the system is fully operational.

- **QIGCCDEFNT**: Double-byte code font

  This value is used when transforming an SNA character string (SCS) into an Advanced Function Printing™ Data Stream (AFPDS). It is also used when creating an AFPDS spooled file with shift in/shift out (SI/SO) characters present in the data. The IGC coded font must exist in the system ASP or in a basic user ASP. The shipped value is different for different countries or regions.

- **QINACTMSGQ**: Inactive job message queue

  This value specifies the action the system takes when an interactive job has been inactive for an interval of time (the time interval is specified by the system value QINACTITV). The interactive job can be ended, disconnected, or message CPI1126 may be sent to the message queue you specify. The message queue must exist in the system ASP or in a basic user ASP.

  If the specified message queue does not exist or is damaged when the inactive timeout interval is reached, the messages are sent to the QSYSOPR message queue. All of the messages in the specified message queue are cleared during an IPL. If you assign a user's message queue as QINACTMSGQ, the user loses all messages that are in the user's message queue during each IPL.

- **QPRBFTR**: Problem log filter

  This value specifies the name of the filter object used by the Service Activity Manager when processing problems. The filter must exist in the system ASP or in a basic user ASP.

- **QPWDVLDPGM**: Password validation program

  This value provides the ability for a user-written program to perform additional validation on passwords. The program must exist in the system ASP or in a basic user ASP.

- **QRMTSIGN**: Remote sign-on control

  This system value specifies how the system handles remote sign-on requests.

  - *Remote session program*

    All values are supported as described for AS/400 Display Station Pass-Through, AS/400 Client Access Work Station Function (WSF), and other 5250 emulation programs on programmable workstations. For information about the level of support provided for TELNET sessions, review the documentation on this system value in the iSeries Information Center on the Web at:

    `http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html`

  - *Program*

    This options allows you to specify the name of a program and library to decide which remote sessions to allow and which user profiles to automatically sign on from which locations. The program must exist in the system ASP or in a basic user ASP.

- **QSRTSEQ**: Sort sequence

  This system value specifies the default sort sequence algorithm to be used by the system.

  - *Sort sequence table name*

    This specifies the name and library of the sort sequence table to be used. The sort sequence table must exist in the system ASP or in a basic user ASP.

- **QSTRUPPGM**: Startup program

  This value specifies the name of the program called from an autostart job when the controlling subsystem is started. This program performs setup functions, such as starting subsystems and printers. The program must exist in the system ASP or in a basic user ASP.

- **QSYSLIBL**: System part of the library list

  When searching for an object in the library list, the libraries in the system part are searched before any libraries in the user part are searched. The list can contain as many as 15 library names. The libraries must exist in the system ASP or in a basic user ASP.

- **QUPSMSGQ**: Uninterruptible power supply (UPS) message queue

  This value specifies the name and library of the message queue that will receive UPS messages. It allows you to monitor the message queue and control the power down. If the message queue is not the system operator message queue (QSYS/QSYSOPR), all UPS messages are also sent to the system operator message queue.

- **QUSRLIBL**: User part of the library list

  When searching for an object in the library list, the libraries in this part are searched after the libraries in the system part and after the product library and current library entries. The list may contain as many as 25 library names. The libraries must exist in the system ASP or in a basic user ASP.

4.9 Network attribute considerations

When you set up independent disk pools for the first time, or move applications to independent disk pools, consider some of the keywords and parameters for the system network attributes. If the keywords and parameters highlighted in the following sections are in use, review them for the impact that independent disk pools may have on their use. These parameters are on the Change Network Attributes (CHGNETA) command. Some of them are on the Retrieve Network Attributes (RVTNETA) command.

For more information about these commands, see the CL Command Finder function in the iSeries Information Center.

`http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html`

To access this function, type `CL Command Finder` in the Search field.

4.9.1 Alert Filters (ALRFTR)

This parameter specifies the qualified name of the alert filter used by the alert manager when processing alerts. The alert filter must exist in the system ASP or in a basic user ASP. The possible values are:

- **SAME**: The value does not change.
- **NONE**: An alert filter is not used.

The name of the alert filter can be qualified by one of the following library values:

- ***LIBL**: All libraries in the job's library list are searched.
- ***CURLIB**: The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.
- **library-name**: Specify the name of the library to be searched.
- **filter-name**: Specify the name of the alert filter.

4.9.2 Message Queue (MSGQ)

This parameter specifies the qualified name of the message queue where messages received through the SNADS network are sent for users with no message queue specified in their user profile or whose message queue is not available. The message queue must exist in the system ASP or in a basic user ASP. The possible value is:

- ***SAME**: The value does not change.

The possible library values are:

- ***LIBL**: The library list is used to locate the message queue.

  When *LIBL is used as the library name, the library list of the job calling this command is searched to find a message queue with the specified object name. If the message queue is found, the name of the library in which it is found is used in the fully qualified name and it is stored. If the message queue is not found, an exception is signaled, and no network attributes are changed.

- ***CURLIB**: The current library for the job is used to locate the message queue. If no library is specified as the current library for the job, the QGPL library is used.

- **library-name**: Specifies the name of the library where the message queue is located.

  When the library name or *CURLIB is specified, this command attempts to find the message queue. If the message queue cannot be found in the specified library, a diagnostic message is sent. If all other parameters on the command are specified correctly, and whether this command can find the message queue in the library specified, the MSGQ network attribute is changed to the qualified message queue name.

- **message-queue-name**: Specifies the name of the message queue.

4.9.3 Distributed Data Management Access (DDMACC)

This parameter specifies how the system processes distributed data management (DDM) and DRDA requests from remote systems for access to the data resources of the system. The DDM and DRDA connections refer to APPC conversations or active TCP/IP or OptiConnect connections. Changes to this parameter are immediate and apply to DRDA, DDM, or DB2® Multisystem applications. However, jobs that are currently running on the system do not use the new value. The DDMACC value is accessed only when a job is first started. You must specify a special value or program name that dictates how the requests are to be handled.

If a program name is specified, the program must exist in the system ASP or in a basic user ASP. The possible values are:

- ***SAME**: The value does not change.
- ***REJECT**: This system does not allow DDM and DRDA requests from remote systems. However, this system can still use DDM or DRDA to access files or SQL tables on remote systems. Source (client) systems cannot access files or SQL tables on an iSeries that specifies *REJECT.
- ***OBJAUT**: If the user profile associated with the DDM or DRDA job is authorized to the files, all file or remote SQL requests are accepted. Object authorities, such as read, write, or update, must also exist for the files.

The possible library values are:

- ***LIBL**: The library list is used to locate the validation program.

  If *LIBL is used as the library name, the library list of the job calling this command is searched to find the program name with the specified object name. If the program name is found, the name of the library in which it is found is used in the fully qualified name and it is stored. If the program name is not found, an error message is sent, and no network attributes are changed.

- ***CURLIB**: The current library for the job is used to locate the validation program. If no library is specified as the current library for the job, the QGPL library is used.

- **library-name**: Specifies the name of the library where the validation program is located.

  When the library name or *CURLIB is specified, this command attempts to find the program name. If the program name cannot be found in the specified library, a diagnostic message is sent to the user. If this command can find the program name in the specified library, the DDMACC network attribute is changed to the qualified program name, if all other parameters on the command are specified correctly.

- **program-name**: Specifies the name of the validation program.

  The program name is the name of the customer validation program that can supplement system object-level security. This user-exit program can restrict user access to *PUBLIC and privately authorized files. The target DDM support calls the user program each time a file is read. The user exit program indicates to DDM whether the request must proceed or end. The iSeries object level security still applies.

  To learn more about this program, see the File Management topic, under File systems and management, in the iSeries Information Center on the Web at:

  `http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html`

4.9.4 PC Support Access (PCSACC)

This parameter specifies how Client Access/400 requests are handled. You must specify a special value or program name that dictates how the requests must be handled. This permits greater control over Client Access/400 applications. Changes to this parameter are immediate. However, jobs currently running on the system do not use the new value. The PCSACC value is used only when a job is first started.

If a program name is specified, the program must exist in the system ASP or in a basic user ASP. The following values are possible:

- ***SAME**: The value does not change.
- ***REJECT**: The system rejects all requests from the client.
- ***OBJAUT**: Normal object authorizations are checked for this client request (for example, authorization to retrieve data from a database file for a transfer facility request).
- ***REGFAC**: The system uses the system's registration facility to determine which exit program (if any) to run. If no exit program is defined for an exit point, and this value is specified, *OBJAUT is used.

The possible library values are:

- ***LIBL**: The library list is used to locate the program.

  When *LIBL is used as the library name, the library list of the job calling this command is searched to find the program name with the specified object name. If the program name is found, the name of the library in which it is found is used in the fully qualified name, and it is stored. If the program name is not found, an error message is sent and no network attributes are changed.

- ***CURLIB**: The current library for the job is used to locate the program. If no library is specified as the current library for the job, the QGPL library is used.

- **library-name**: Specifies the name of the library where the program is located.

  When the library name or *CURLIB is specified, this command attempts to find the program name. If the program name cannot be found in the specified library, a diagnostic message is sent to the user. If all other parameters on the command were specified correctly, and whether this command can find the program name in the specified library, the PCSACC network attribute is changed to the qualified program name.

- **program-name**: Specifies the name of the program.

  The program name is the name of the customer-supplied Client Access/400 host system application exit program that can supplement system object-level security. This user-exit program can restrict requests handled from the client. Each personal computer support application calls the exit program for requests from the client. Two parameters are passed to the user-exit program. The first describes the client request (which application and what kind of request). The second is used by the exit program to indicate to the client support application whether this client request must be handled.

4.10 Journaling considerations

This section explains some of the rules for journaling where independent disk pools are concerned.

4.10.1 Journaling boundaries

The objects being journaled, the journal (*JRN) itself, and the associated receivers (*JRNRCV) must be in either the system and user ASP combined (*SYSBAS) or in an ASP group. The function of journaling requires access to all three sets of objects at all times.

Figure 4-5 represents the concept that you cannot break up your journaling partners across an IASP boundary.

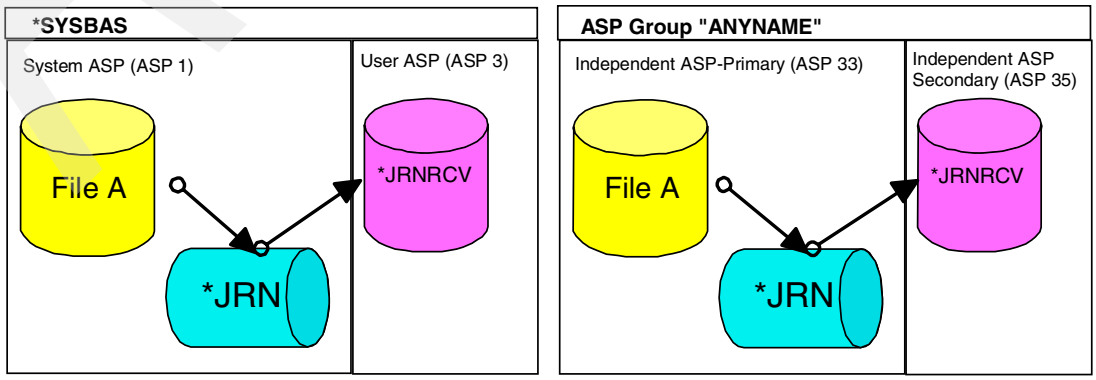


Figure 4-5 Journaling boundaries

4.10.2 Journaling with user-defined file system (UDFS) and library-capable independent disk pools

UDFS independent disk pools only have a user-defined file system. UDFS independent disk pools cannot store journals and receivers. In contrast to UDFS disk pools, library-capable independent disk pools (primary or secondary) have libraries and can store journals and receivers. If you plan to journal objects on an independent disk pool, you must use a library-capable independent disk pool. You cannot journal objects on a UDFS independent disk pool.

IFS objects: A library-capable independent disk pool can have IFS objects. You can also journal IFS objects on a library-capable independent disk pool.

If the objects are library-capable, you can journal them on either switchable or non-switchable independent disk pools.

4.10.3 Journaling and disk pool groups

You can group switchable independent disk pools into *disk pool groups*. Disk pool groups consist of one primary disk pool and one or more secondary disk pools. If you are going to journal an object in a disk pool group, the object and the journal must be in the same disk pool. The journal receiver can be in a different disk pool, but must be in the same disk pool group as the journal and journaled object.

Use the following rules when journaling objects on independent disk pools:

- The disk pool must be available on the system on which you are working.
- The disk pool must be a library-capable disk pool. You cannot journal an object on a UDFS independent disk pool.
- In a disk pool group, the journaled object and the journal must be in the same disk pool.
- In a disk pool group, the journal receiver can be in a different disk pool, but must be in the same disk pool group.

4.11 Subsystem considerations

Subsystem descriptions can exist as objects in an IASP or an IASP group. However, they cannot be started or run from there. Subsystem descriptions can only be started and run from the *SYSBAS.

The subsystem monitor job runs with a name space of *SYSBAS. As a result, it cannot find such objects as job descriptions, programs, and classes if they are in an IASP. Therefore, the objects referenced from a subsystem description should also be in *SYSBAS.

We recommend that all subsystem descriptions and objects referenced by subsystem descriptions reside in the system ASP and that only backup copies of the descriptions should reside in independent disk pools.

4.12 DRDA considerations

There are certain DRDA-related objects that cannot be contained in user databases. DDM user exit programs must reside in libraries in the system database, as must any ARD programs.

You should be aware that the process of varying on a user database causes the RDB directory to be unavailable for a period of time. This can cause attempts by a DRDA application requester or application server to use the directory that is to be delayed or to time out. The exposure to having directory operations time out due to unavailability caused by varying on a database is much greater if multiple databases are varied on at the same time. As noted below, the first time a user database is varied on, an attempt is made by the server to add a directory entry for that database. If the directory is unavailable due to a concurrent vary on operation, the addition will fail, in which case, you must manually add the entry.

Another consideration for using user databases is in regard to configuration of entries in the RDB directory. One of the rules for naming user databases is that user RDB names cannot match the system name specified in the network attributes (as displayed by the Display Network Attributes (DSPNETA) command).

Local user database entries in the RDB directory are added automatically the first time that the associated databases are varied on. They are created using the *IP protocol type and with the remote location designated as LOOPBACK. LOOPBACK indicates that the database is on the same server as the directory. We recommend that you configure the user databases, which are intended to be switched among servers, to have a dedicated IP address associated with them. If the switchable database does not have a dedicated IP address, then whenever it is switched, you must manually update its directory entry on all the servers that reference that database.

There are two ways to associate the application takeover IP address with an application cluster resource group (CRG) managed. The easiest way, which is the default, is for Cluster Resource Services to manage the IP address. This method directs Cluster Resource Services to create the IP address on all nodes in the recovery domain, including nodes that were subsequently added to the recovery domain. When this method is selected, the IP address cannot currently be defined on any node in the recovery domain.

The alternative way is to manage the IP addresses yourself. This method directs Cluster Resource Services not to take any steps to configure the IP address. The user is responsible for the configuration. You must add the takeover IP address on all nodes in the recovery domain (except on replicate nodes) before you start the CRG.

You must configure the IP address of any node to add to the recovery domain of an active CRG before you add it.

4.13 Commitment control considerations

Independent disk pools and independent disk pool groups can each have a separate OS/400 SQL database. Commitment control can be used with these databases. However, since each independent disk pool or independent disk pool group has a separate SQL database, we make the recommendations offered in the following sections.

4.13.1 Commitment definitions

When commitment control is started, the commitment definition is created in the QRECOVERY library. Each independent disk pool or independent disk pool group has its own version of a QRECOVERY library. On an independent disk pool, the name of the QRECOVERY library is QRCY*xxxxx*, where *xxxxx* is the number of the independent disk pool. For example, the name of the QRECOVERY library for independent disk pool 39 is QRCY00039. Furthermore, if the independent disk pool is part of a disk pool group, only the

primary disk pool has a QRCYxxxxx library. When you start commitment control, the commitment definition is created in the QRECOVERY library of the independent disk pool that is associated with that job. This makes commitment control active on the independent disk pool.

Using the Set ASP Group (SETASPGRP) command while commitment control is active on an independent disk pool has the following effects:

- If you switch from an independent disk pool, and resources are registered with commitment control on the disk pool, the SETASPGRP command fails with message CPDB8EC, reason code 2, "`The thread has an uncommitted transaction`." This message if followed by message CPFB8E9. If you switch from an independent disk pool and no resources are registered with commitment control, the commitment definitions are moved to the independent disk pool to which you are switching. If you switch from the system disk pool (ASP group *NONE), commitment control is not affected. The commitment definitions stay on the system disk pool.
- If you use a notify object, the notify object must reside on the same independent disk pool or independent disk pool group as the commitment definition. If you move the commitment definition to another independent disk pool or independent disk pool group, the notify object must also reside on that other independent disk pool or independent disk pool group. The notify object on the other independent disk pool or independent disk pool group is updated if the commitment definition ends abnormally. If the notify object is not found on the other independent disk pool or independent disk pool group, the update fails with message CPF8358.

Recovery of commitment definitions residing on an independent disk pool is performed during independent disk pool vary on processing. It is similar to IPL recovery. Commitment definitions in an independent disk pool are not recovered during the system IPL. The vary off of an independent disk pool has the following effects on commitment definitions:

- Jobs associated with the independent disk pool end.
- No new commitment definitions are allowed to be created on the independent disk pool.
- Commitment definitions residing on the independent disk pool become unusable.
- If the same definitions are not attached to a job, the release transaction that is scoped locks.

You cannot use an LU6.2 SNA connection (protected conversations or DUW) to connect to a remote database from an independent disk pool database. You can use unprotected SNA conversations to connect from an independent disk pool database to a remote database.

When commitment control is active for a job or thread, access to data outside the independent disk pool or disk pool group to which the commitment definition belongs is only possible remotely, as if it were data that resides on another system. When you issue an SQL CONNECT statement to connect to the RDB on the independent disk pool, the system makes the connection a remote connection. The system disk pool and basic disk pools do not require a remote connection for read only access to data that resides on an independent disk pool. Likewise, an independent disk pool does not require a remote connection for read only access to data that resides on the system disk pool or a basic disk pool.

4.13.2 Considerations for XA transactions

In the XA environment, each database is considered a separate resource manager. When a transaction manager wants to access two databases under the same transaction, it must use the XA protocols to perform a two-phase commit with the two resource managers. Since each independent disk pool is a separate SQL database, in the XA environment, each independent disk pool is also considered a separate resource manager. For an application server to perform a transaction that targets two different independent disk pools, the transaction manager must also use a two-phase commit protocol.

4.13.3 Commitment control recommendations

We recommend that you code commitment control in any applications that use switchable disk pools, since the failover processing is similar to abnormal IPL database processing. You may have to resort to restoring your database from your last set of save tapes. This can help you regain database consistency after disk pool failover if commitment control is not used in your database applications.

4.14 Exit programs

Exit programs are a special consideration when dealing with independent disk pools. If your independent disk pools contain primarily data, the concern isn't as great. However, you may have multiple independent disk pools on your system, containing application programs. In this case, it may be worthwhile considering an application library in *SYSBAS where commonly used application programs and exit programs can reside. This makes those programs accessible even when one or more of the independent disk pools are unavailable.

Clustering brings another special consideration into play. An exit program parameter is available from the CRG. If you use this parameter, never store the exit program in an independent disk pool. In case of failover, this program needs to be available, and, if it is in the independent disk pool, it is one of the entities being switched. During the time the pool is offline, the program cannot be accessed.

4.15 System libraries

Avoid or eliminate the use of system libraries, such as QGPL, QSYS, QSYS2, and essentially any library beginning with the letter Q, whenever possible. Applications that are being written or moved to independent disk pools should *not* use system libraries. Those libraries remain in the system ASP. Then the objects created or used in those libraries do not switch with the independent disk pool when it is moved to another system.

Consider a user-created library in the independent disk pool first when eliminating your use of system libraries. If this does not work for you, then consider using a separate user library in the system ASP.

4.16 System ASP and all basic user ASPs (*SYSBAS)

The ASP group of applications that are being written or moved to independent disk pools is designated only as permanent storage. The exception is if the application is written or designed to currently work across multiple systems or partitions.

Although permanent storage for the system ASPs is available to the applications running from an independent disk pool or ASP group, avoid using this storage whenever possible. You must view *SYSBAS (system ASP and all basic user ASPs) as the domain and storage for operating system code, LPP code, objects that are not supported in independent disk pools, and temporary working storage for applications.

4.17 Other system considerations

Consider the additional items in the following sections when planning application migration to independent ASPs.

System-wide is no longer system-wide

Designers and developers must be aware that jobs that used to accomplish work on a system-wide basis now only accomplish such work on a name-space basis or database basis. Or the jobs may accomplish this work across the set of available (varied on) independent disk pools and *SYSBAS. This is expressed in the parameters used for several keywords. Name space-wide basis is referred to as *ALL for all available independent disk pools. *SYSBAS is referred to as *ALLAVL.

May need ASP group, library, or object to identify an object

The object name and library name may not be enough to uniquely identify an object any longer. Suppose you have a job that provides services for other jobs. Also consider that your server takes work requests for several different independent disk pools. Someone requests a task with LIBNAME/OBJNAME in one independent disk pool, and the next user of this job's services asks for a task with LIBNAME/OBJNAME in a different independent disk pool. These are different objects if the library is in the independent disk pools. However, they are the same object if the library is in *SYSBAS.

To the application, this means:

- You need the independent disk pool identification in addition to the library name and object name.
- Various places where data may be cached must be redone to include the independent disk pool as part of the object designation. Keep in mind that the independent disk pool may go offline at any time.

Locking

Locking no longer prevents an object from changing or being deleted in some cases. When you lock an object that is in an independent disk pool group and then change the name space for the job or thread to work with *SYSBAS or a different independent disk pool group, you still have a lock on the object.

Note that Deallocate Object (DLCOBJ) running in the new name space cannot deallocate something that it cannot find. It cannot find anything outside of the current name space, so you cannot get rid of a lock unless you are running in the correct name space.

Consider this problem. While you are not running in the name space for the locked object, no jobs may be running in that name space. Therefore, there is no reason for the system to prevent the independent disk pool from being taken offline and switched to a different system. You may hold an exclusive lock, but the object is on a different system that may be modified or deleted.

Unable to change a name space

There are times when you cannot change a name space. SETASPGRP rejects the request if there is an uncommitted transaction or if an operation in progress blocked the name space change. You cannot change the name space when:

- Running in a program invoked as a result of pressing the attention key (SETATNPGM)
- Running in a pre-attention exit program
- Running in a pre-system-request exit program
- Running in a message queue's break handling program
- Any user exits from certain system jobs

The other side of this is true. That is, it is a really bad idea to produce an open, uncommitted transaction from a user exit program. You can prevent whatever is running (in the environment from which this user exit was invoked) from a name space switch.

Workflow design and control for use of independent disk pools

Consider doing workflow design and controlling which jobs use the independent disk pool at certain times. Having an independent disk pool in the name space of a job or thread prevents the independent disk pool from being made unavailable. Therefore, some servers should avoid waiting for new work while the independent disk pool is in the name space. This is particularly true for servers that span independent disk pool boundaries. You don't want to optimize out name space switches by waiting to see if a switch is needed. In doing so, you interfere with make unavailable, administrative switchback, and other scenarios.

Moreover, if your server works for different users, you need to swap users and then try the SETASPGRP command so that you get proper security checking. The new user is required to have *USE authority to the device descriptions for each ASP in the group.

Prestart jobs may need to avoid waiting for work in a state that holds the independent disk pool active. On the other hand, you can avoid this requirement by shutting down the subsystem before you vary off the independent disk pool.

When a subsystem takes work from a JOBQ and the jobs have an associated ASP group, you have two considerations:

- For one-at-a-time batch processing, no jobs may be using the independent disk pool for a short time between jobs. This allows the independent disk pools to be made unavailable before you really wanted that done. The "jobs using an IASP" interfaces don't tell you about jobs that want to use the independent disk pool in the immediate future.
- When the independent disk pool is taken down, job initiation fails for jobs that were submitted to run with that independent disk pool. An amazing number of jobs can be removed from a JOBQ and killed before you realize that anything is wrong.

5

Configuration and management

This chapter explains how to create, configure, and manage an independent disk pool. In doing so, it looks at the various functions that are available within the IASP.

5.1 The heart of an independent disk pool

When an independent disk pool is created, a new relational database is also created. Several objects are created as well in the independent disk pool at this time. At vary on, these objects are synchronized and accessed by jobs in the QSYSWRK subsystem. This section explains this further. It also shows you some of the displays that you will see as you create, configure, and manage an independent disk pool.

5.1.1 From the green-screen side

The following sections show the independent auxiliary storage pool (IASP) configuration and maintenance displays that are available from a 5250 session. The displays list and define the following elements:

- The device description
- The database directory entry
- Libraries or collections
- Subsystem jobs
- Files in libraries

The device description

Before you create the disk pool, you enter the following command:

```
WRKCFGSTS *DEV *ASP
```

Then you see Work with Configuration Status display (Figure 5-1).

```
                     Work with Configuration Status
                                                          01/11/03  23:29:10
Position to  . . . . .               Starting characters

Type options, press Enter.
  1=Vary on   2=Vary off   5=Work with job   8=Work with description
  9=Display mode status    13=Work with APPN status...

Opt  Description       Status              -------------Job--------------



                                                                      Bottom
Parameters or command
===>
F3=Exit   F4=Prompt   F12=Cancel   F23=More options   F24=More keys
```

Figure 5-1 WRKCFGSTS display before creating the disk pool

You create the two IASPs using iSeries Navigator. Then after you create the pools, you enter the same command again. Now the Work with Configuration Status display (Figure 5-2) shows DBITSC and EUROPE as the ASP devices. Now the pool is available as part of the process.

```
                      Work with Configuration Status
                                                              01/11/03  23:29:10
Position to  . . . . .               Starting characters

Type options, press Enter.
  1=Vary on   2=Vary off   5=Work with job   8=Work with description
  9=Display mode status    13=Work with APPN status...

Opt  Description       Status               -------------Job--------------
     DBITSC           AVAILABLE
     EUROPE           AVAILABLE



                                                                        Bottom
Parameters or command
===>
F3=Exit   F4=Prompt   F12=Cancel   F23=More options   F24=More keys
```

Figure 5-2 WRKCFGSTS display after creating the disk pools

The database directory entry

Before you create the disk pool and database, you enter the command:

```
WRKRDBDIRE
```

Then you see the Work with Relational Database Directory Entries display (Figure 5-3).

```
               Work with Relational Database Directory Entries

Position to  . . . . . .

Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display details   6=Print details

        Relational          Remote
Option  Database            Location                  Text

        DIEAS450            9.36.149.12
        AS20                9.5.92.28
        AS01                asm01
        XTREME              9.36.240.19
        RCHASE5C            *LOCAL                    Entry added by system


                                                                        Bottom
F3=Exit   F5=Refresh   F6=Print list   F12=Cancel
(C) COPYRIGHT IBM CORP. 1980, 2002.
```

Figure 5-3 WRKRDBDIRE before creating the disk pool and database

Then you create the database and disk pool named DBCOOK and DBEUOPS. The database is created at the same time as the disk pool. The two are essentially synonymous.

You enter the WRKRDBDIRE command again. Now the Work with Relational Database Directory Entries display appears as shown in Figure 5-4. In this example, the *LOCAL entry is for the current system database name.

```
                Work with Relational Database Directory Entries

Position to  . . . . . .

Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display details   6=Print details

        Relational          Remote
Option  Database            Location                  Text

        DIEAS450            9.36.149.12
        AS20                9.5.92.28
        AS01                asm01
        XTREME              9.36.240.19
        DBCOOK              LOOPBACK                  Entry added by system
        DBEUOPS             LOOPBACK                  Entry added by system
        RCHASE5C            *LOCAL                    Entry added by system



                                                                        Bottom
F3=Exit   F5=Refresh   F6=Print list   F12=Cancel
(C) COPYRIGHT IBM CORP. 1980, 2002.
```

Figure 5-4 WRKRDBDIRE after creating the disk pool and database

The libraries (collections)

Before you create the disk pool and database, you enter the command:

```
WRKLIB LIB(*ALL) ASPDEV(DLB1)
```

Then you see the message "`Device DLB1 not found`" on the Work with Libraries display.

When you enter the same command again after you create the disk pool and database, the Work with Libraries display appears as shown in Figure 5-5.

```
                           Work with Libraries

Type options, press Enter.
  1=Create   2=Change   3=Copy    4=Delete   5=Display   6=Print
  8=Display library description   9=Save     10=Restore
  11=Save changed objects         12=Work with objects   14=Clear

                           ASP
Opt  Library     Attribute   Device      Text

     QRCL00033   PROD        DLB1        Reclaim storage library.
     QRCY00033   PROD        DLB1        Recovery Library
     QRPL00033   PROD        DLB1        Library for REPLACE(*YES) processing.
     QSYS00033   PROD        DLB1        System Library
     QSYS200033  PROD        DLB1        System Library for CPI's
     SYSIB00033  PROD        DLB1        System Library for DB2



                                                                    Bottom
```

Figure 5-5 WRKLIB display after creating the disk pool and database

In addition to creating the pool, and making it available, the libraries or collections shown in Figure 5-5 were created and physically located on or in the independent disk pool. We discuss the nature of each of these items later.

Notice that the names of all the libraries end in “33”. This was the first independent disk pool created on this system, so the internal pool ID of this disk pool was assigned 33. As these libraries were created, they were assigned the pool number as part of the object name.

Subsystem QSYSWRK entries

When the independent disk pool is made available, several tasks are run and several jobs are started in QSYSWRK subsystem. You enter the following command:

```
WRKACTJOB SBS(QSYSWRK) JOB(QDB*)
```

We entered the subset of QDB so we can see only the jobs that relate to database services. Then you see the Work with Active Jobs display as shown in Figure 5-6.

```
                          Work with Active Jobs                    AAAAS02
                                                          06/12/02  15:04:53
CPU %:     .0     Elapsed time:   00:00:00     Active jobs:   183

Type options, press Enter.
  2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
  8=Work with spooled files   13=Disconnect ...

Opt  Subsystem/Job  User        Type  CPU %  Function        Status
       QDBSO33V01   QSYS        BCH      .0                  EVTW
       QDBSO33V02   QSYS        BCH      .0                  DEQW
       QDBSO33V03   QSYS        BCH      .0                  DEQW
       QDBSO33V04   QSYS        BCH      .0                  DEQW
       QDBSO33V05   QSYS        BCH      .0                  DEQW
       QDBXO33XR    QSYS        BCH      .0                  DEQW
       QDBXO33XR2   QSYS        BCH      .0                  DEQW


                                                                  Bottom
```

Figure 5-6 WRKACTJOB after the disk pools are made available

Files in the libraries

When the libraries are created in the independent disk pool, files are also created and initialized in some of those libraries. The following examples show the files created in each library. They also include a brief description of the purpose of the library and some of the files.

QRCL00033: Reclaim storage library

Figure 5-7 shows the Reclaim storage library when the independent disk pool was created. Notice that no objects were in this library at the time the independent disk pool was created.

```
5722SS1 V5R2M0  020719                 Display Library    6/12/02 15:14:47 Page    1
Library . . . . . . . . . . . . . . . :   QRCL00033
Type  . . . . . . . . . . . . . . . . :   PROD
Number of objects . . . . . . . . . . :           0
Library ASP number  . . . . . . . . . :        33
Library ASP device  . . . . . . . . . :   DLB1
Create authority  . . . . . . . . . . :   *SYSVAL
Text description  . . . . . . . . . . :   Reclaim storage library.
  Object      Type      Attribute            Size  Description
  (Library has no objects)
                * * * * *   E N D   O F   L I S T I N G   * * * * *
```

Figure 5-7 DSPLIB of QRCL00033: Reclaim storage library

QRCY00033: Recovery library

Figure 5-8 shows the Recovery library when the independent disk pool was created. Six objects were in this library when the independent disk pool was created.

```
5722SS1 V5R2M0  020719                 Display Library     6/12/02 15:14:47 Page    1
Library . . . . . . . . . . . . . . . :   QRCY00033
Type  . . . . . . . . . . . . . . . . :   PROD
Number of objects . . . . . . . . . . :           6
Library ASP number  . . . . . . . . . :        33
Library ASP device  . . . . . . . . . :   DLB1
Create authority  . . . . . . . . . . :   *SYSVAL
Text description  . . . . . . . . . . :   Recovery Library
  Object      Type      Attribute            Size  Description
  QDBJNC0001  *JRNRCV                      978944
  QDBJNF0001  *JRNRCV                      327680
  QSQJRN0001  *JRNRCV                      389120
  QDBJRNCOLM  *JRN                           8192
  QDBJRNFILE  *JRN                           8192
  QAUGDBLL    *FILE     PF                  36864
                    Total size :          1822720
                * * * * *   E N D   O F   L I S T I N G   * * * * *
```

Figure 5-8 DSPLIB for QRCL00033: Recovery library

*QRPL00033: Library for REPLACE(*YES) processing*

Figure 5-9 shows the Library for REPLACE(*YES) processing when the independent disk pool was created. No objects were in this library when the independent disk pool was created.

```
5722SS1 V5R2M0  020719                 Display Library 6/12/02 15:14:47     Page    1
Library . . . . . :   QRPL00033
Type  . . . . . :   PROD
Number of objects . . . . . :           0
Library ASP number  . . . . . :       33
Library ASP device  . . . . . :   DLB1
Create authority  . . . . . :   *SYSVAL
Text description  . . . . . :   Library for REPLACE(*YES) processing.
  Object      Type      Attribute            Size  Description
  (Library has no objects)
                * * * * *   E N D   O F   L I S T I N G   * * * * *
```

*Figure 5-9 DSPLIB for QRPL00033: Library for REPLACE(*YES) processing*

QSYS00033: System library

Figure 5-10 shows the System library when the independent disk pool was created. Notice that 35 objects were in this library when the independent disk pool was created.

```
5722SS1 V5R2M0  020719                 Display Library           6/12/02 15:14:48       Page    1
Library . . . . . . . . . . . . . . . :   QSYS00033
Type  . . . . . . . . . . . . . . . . :   PROD
Number of objects . . . . . . . . . . :          35
Library ASP number  . . . . . . . . . :       33
Library ASP device  . . . . . . . . . :   DLB1
Create authority  . . . . . . . . . . :   *SYSVAL
Text description  . . . . . . . . . . :   System Library
  Object      Type      Attribute            Size  Description
  QADBCCST    *FILE     PF                  77824  Constraint Field Usage Information
  QADBCKCL    *FILE     LF                  36864  LF for QADBCCST by Constraint Library and File
  QADBFCST    *FILE     PF                 102400  File Level Constraint Cross Reference File
  QADBFDEP    *FILE     PF                 208896  Cross reference dependency file
  QADBIATR    *FILE     LF                2646016  Cross reference logical file by attribute
  QADBIFLD    *FILE     PF               19763200  Cross reference physical file
  QADBILFI    *FILE     LF                2654208  Cross reference logical file by long name
  QADBILLB    *FILE     LF                1335296  Cross reference logical file by long library name
  QADBIUDT    *FILE     LF                1073152  CROSS-REFERENCE LOGICAL BY UDT IDENTIFIER
  QADBKATR    *FILE     LF                  12288  Cross reference logical file over QADBKFLD
  QADBKFLD    *FILE     PF                 524288  Cross reference physical file
  QADBLDEP    *FILE     LF                  81920  Dependency logical file by dependency
  QADBLDNC    *FILE     LF                 102400  Dependency logical multiple-format file
  QADBLPKG    *FILE     LF                  36864  SQL Package logical file
  QADBPKG     *FILE     PF                  65536  SQL Package physical file
  QADBRKCL    *FILE     LF                  49152  LF for QADBFCST by Constraint Library and File
  QADBRKPA    *FILE     LF                  81920  LF for QADBFCST by Parent Library and File
  QADBRKPN    *FILE     LF                  81920  LF for QADBFCST by Unique Constraint Name & Lib
  QADBXATR    *FILE     LF                 122880  Cross reference logical file by attribute
  QADBXDIC    *FILE     LF                  90112  Cross reference logical file by dictionary
  QADBXFIL    *FILE     LF                 188416  Cross reference logical file by file
  QADBXLFI    *FILE     LF                 131072  Cross reference logical file by long name
  QADBXLFN    *FILE     LF                 188416  Cross reference logical file by long name
  QADBXLLB    *FILE     LF                 131072  Cross reference logical file by long library name
  QADBXREF    *FILE     PF                 884736  Cross reference physical file
  QADBXSFKEY  *FILE     LF                 155648
  QADBXSFLD   *FILE     PF                  69632
  QADBXTBFIL  *FILE     LF                 172032
  QADBXTBFSN  *FILE     LF                 172032
  QADBXTCDEF  *FILE     LF                 155648
  QADBXTDDEF  *FILE     LF                 155648
  QADBXTDDEP  *FILE     LF                 155648
  QADBXTRIGB  *FILE     PF                 356352
  QADBXTRIGC  *FILE     PF                  61440
  QADBXTRIGD  *FILE     PF                  90112
                  Total size :            32305152
                * * * * *   E N D   O F   L I S T I N G   * * * * *
```

Figure 5-10 DSPLIB QSYS00033: System library

QSYS200033: System library for CPIs

Figure 5-11 shows the System library for CPIs when the independent disk pool was created. Notice that 35 objects were in this library when the independent disk pool was created.

```
5722SS1 V5R2M0  020719                 Display Library       6/12/02 15:14:48        Page    1
Library . . . . . :   QSYS200033
Type  . . . . . :   PROD
Number of objects . . . . . :          31
Library ASP number  . . . . . :       33
Library ASP device  . . . . . :   DLB1
Create authority  . . . . . :   *SYSVAL
Text description  . . . . . :   System Library for CPIs
  Object      Type      Attribute            Size  Description
  QSQJRN      *JRN                            8192
  PROCEDURES  *FILE     LF                   49152
  QASQDRDP    *FILE     LF                  155648
  QASQRESL    *FILE     LF                   61440
  QASQSPDP    *FILE     LF                  155648
  QSQPTABL    *FILE     PF                   28672
  SYSCHKCST   *FILE     LF                   24576
  SYSCOLUMNS  *FILE     LF                   90112
  SYSCST      *FILE     LF                   49152
  SYSCSTCOL   *FILE     LF                   40960
  SYSCSTDEP   *FILE     LF                   36864
  SYSFUNCS    *FILE     LF                   53248
  SYSINDEXES  *FILE     LF                   69632
  SYSJARCONT  *FILE     PF                   32768
  SYSJAROBJ   *FILE     PF                   40960
  SYSKEYCST   *FILE     LF                   53248
  SYSKEYS     *FILE     LF                   53248
  SYSPACKAGE  *FILE     LF                   77824
  SYSPARMS    *FILE     PF                  217088
  SYSPROCS    *FILE     LF                   45056
  SYSREFCST   *FILE     LF                   49152
  SYSROUTINE  *FILE     PF                  241664
  SYSRTNDEP   *FILE     PF                   49152
  SYSTABLES   *FILE     LF                   49152
  SYSTRIGCOL  *FILE     LF                   45056
  SYSTRIGDEP  *FILE     LF                   45056
  SYSTRIGGER  *FILE     LF                   94208
  SYSTRIGUPD  *FILE     LF                   40960
  SYSTYPES    *FILE     PF                  192512
  SYSVIEWDEP  *FILE     LF                   86016
  SYSVIEWS    *FILE     LF                   45056
                  Total size :             2371584
              * * * * *   E N D   O F   L I S T I N G   * * * * *
```

Figure 5-11 DSPLIB QSYS200033: System library for CPIs

SYSIB00033: System library for DB2

Figure 5-12 shows the System library for DB2 when the independent disk pool was created. No objects were in this library when the independent disk pool was created.

```
5722SS1 V5R2M0  020719                 Display Library     6/12/02 15:14:48 Page    1
Library . . . . . . . . . . . . . . . :   SYSIB00033
Type  . . . . . . . . . . . . . . . . :   PROD
Number of objects . . . . . . . . . . :            0
Library ASP number  . . . . . . . . . :         33
Library ASP device  . . . . . . . . . :   DLB1
Create authority  . . . . . . . . . . :   *SYSVAL
Text description  . . . . . . . . . . :   System Library for DB2
  Object      Type      Attribute            Size  Description
  (Library has no objects)
                 * * * * *   E N D   O F   L I S T I N G   * * * * *
```

Figure 5-12 DSPLIB of SYSIB00033: System library for DB2

5.2 Disk pool operation

A system can bring an IASP online and make it active at any time during IPL after the QSYSWRK subsystem is active. The system can bring IASP online during IPL if the IASP is varied on using the Vary Configuration (VRYCFG) command in the startup program. After the IASP is online and active, the objects within the IASP are accessible and usable.

Important: An IASP can go offline and yet the rest of the system remains functional. However, after 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.

A system in which an IASP is online can vary that IASP offline. The system can do this explicitly by a user request. Or it can do this implicitly through system termination.

5.2.1 Disk pool and disk pool group

To help you understand what a disk pool and disk pool group are, complete the following steps using iSeries Navigator:

1. Expand a system under **My Connections**.
2. Expand **Configuration and Service**.
3. Expand **Hardware**.
4. Expand **Disk Units**.
5. At the sign-on prompt, enter the DST user profile and password for QSECOFR for your system.

You see two subheadings: Disk Pools and Disk Pools Groups. Disk Pools shows all available ASPs on the system, regardless of hierarchy. Disk Pool Groups shows *only* IASPs, listed in order of group, primary IASP, and secondary IASP.

5.2.2 Making an independent disk pool available

You must make an independent disk pool available (by varying it on) to access and work with its data. A switchable disk pool can be made available on any node in the recovery domain of the cluster resource group, if it is switched to that node first.

Note: If you make a primary or secondary disk pool available, all of the disk pools in the disk pool group are also made available at the same time.

To make an independent disk pool available, follow these steps:

1. Open iSeries Navigator and expand **My Connections**.
2. Expand the iSeries server on which your IASP is located. After the initial creation, the IASP is allocated to the cluster's primary system.
3. Expand **Configuration and Service-> Hardware-> Disk Units**.
4. When the Service Device Sign-on window appears, sign on with a QSECOFR level DST user ID and password. Remember, this password *is* case sensitive.
5. Expand **Disk Pools**. Right-click the disk pool to be made available to the system, and select **Make Available** (Figure 5-13). You can select multiple disk pools to make available at the same time.

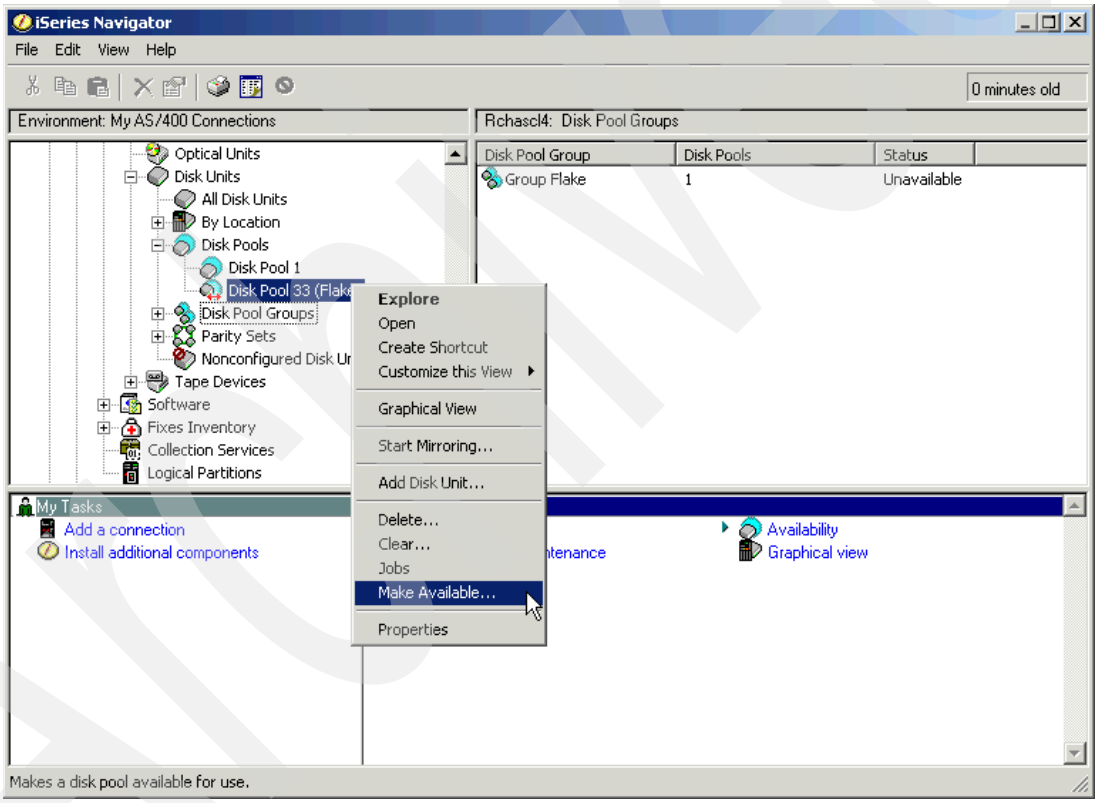


Figure 5-13 iSeries Navigator: Choosing the Make Available option

6. Confirm the option by clicking the **Make Available** button on the confirmation panel (Figure 5-14). A confirmation window confirms the success.

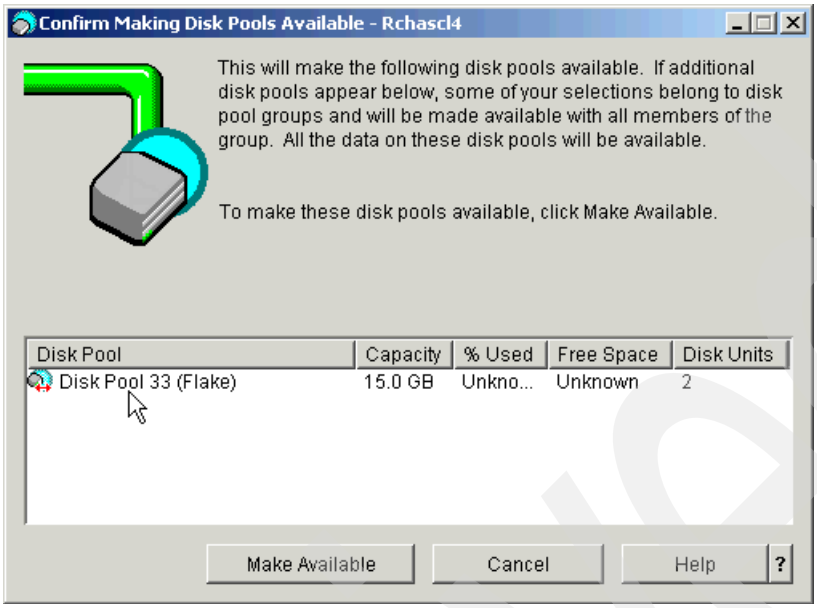


Figure 5-14 iSeries Navigator: Confirming the Make Available option

7. On a 5250 display, enter the command:

   ```
   WRKDSKSTS
   ```

8. Observe the disk units beginning with a “4” in the thousands position. These are the newly available IASP disk units. You can use this IASP by adding it into your jobs name space by entering the following command:

   ```
   SETASPGRP
   ```

After the IASP is available and within your jobs name space, you can manipulate it, as any of the other 32 ASPs, by commands that support an ASP device or ASP number.

5.2.3 Duration of the Make Available option

The duration of the Make Available (vary on) option is sensitive to the number of objects in *SYSBAS. If the Make Available time is more that two minutes, many application functions may start to time out and fail.

On systems where the number of objects in *SYSBAS is relatively small (OS/400 and licensed programs only installed), the Make Available time is relatively quick, under a minute in test environments. On systems where the number of objects in *SYSBAS is extremely large, the Make Available time has taken quite a bit longer.

5.2.4 Making an independent disk pool unavailable

You can select an independent disk pool to make it unavailable (varied off) in the same way it was made available (varied on). You cannot access any of the disk units in the independent disk pool until it is made available (varied on) again. The pool can be made available again on the same system or another system in the recovery domain of the cluster resource group (CRG), after a switch occurs.

To make an independent disk pool unavailable, follow these steps:

1. In iSeries Navigator, expand **My Connections**.
2. Expand any iSeries server.
3. Expand **Configuration and Service-> Hardware-> Disk Units**.
4. When the Service Device Sign-on display opens, sign on with a QSECOFR level DST user ID and password. Remember, this password *is* case sensitive.

   Important: Make sure that no active jobs are using the disk pool prior to making the pool unavailable.

5. Expand **Disk Pools**. Right-click the available independent disk pool and select **Make Unavailable**.
6. On the panel that follows, confirm the selection to make the disk pool unavailable. A panel opens that indicates the success of the unavailability.

5.2.5 Independent disk pool overflow

There is a difference between user ASPs and independent ASPs when it comes to data overflow. That is, user ASPs overflow, and independent ASPs don't.

An overflow of a basic user 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 considered independent or switchable. An IASP is allowed to fill up, and the application that is responsible for filling it up simply halts. There is no message that says the IASP is full, and there is no automatic cancellation of the responsible job. If this job is running from a single-threaded JOBQ, in a single-threaded subsystem, all further processing is stopped until user action is initiated.

5.2.6 Independent disk pool IPL considerations

An IASP is activated by varying on the *ASP device on the system or by the Make Available option in iSeries Navigator. After a system IPL, disk pools are not automatically activated. This requires manual intervention to vary on the device or use of a CL program such as the startup program.

In addition to the IASP availability, you must cluster resources after you perform an IPL. To learn more about this, see *Clustering and IASPs for Higher Availability on the IBM @server iSeries Server*, SG24-5194.

5.2.7 Switching independent disk pools

A planned switch of an independent disk pool to its designated backup can be performed at any time, provided Cluster Resource Services are available and the cluster nodes are synchronized. Cluster software performs the process of switching independent disk pools between nodes within a domain. If the independent disk pool is to be switched between logical partitions, then the disk units can be either internal or external components.

If the IASPs are to be switched between machines (that is, separate Central Electronic Complexes (CECs)), then the disk units must be external, such as those in a switchable tower or a storage area network (SAN).

A logical system power control network (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.

5.2.8 Planned disk pool switch

To perform a planned disk pool switch, follow these steps:

1. End all jobs using objects within the IASP.
2. Place on hold the job queues for jobs using the disk pool.
3. Make the disk pool unavailable. You can use iSeries Navigator to do this as explained in 5.2.4, “Making an independent disk pool unavailable” on page 64. Or you can use Cluster Resource Services to make the disk pool unavailable before switching or failing it over to the target (new primary) node.
4. Expand **Clusters-> *your cluster name*-> Switchable Hardware**.
5. Select your **Switchable Hardware group**, right-click, and select **Switch** as shown in Figure 5-15.

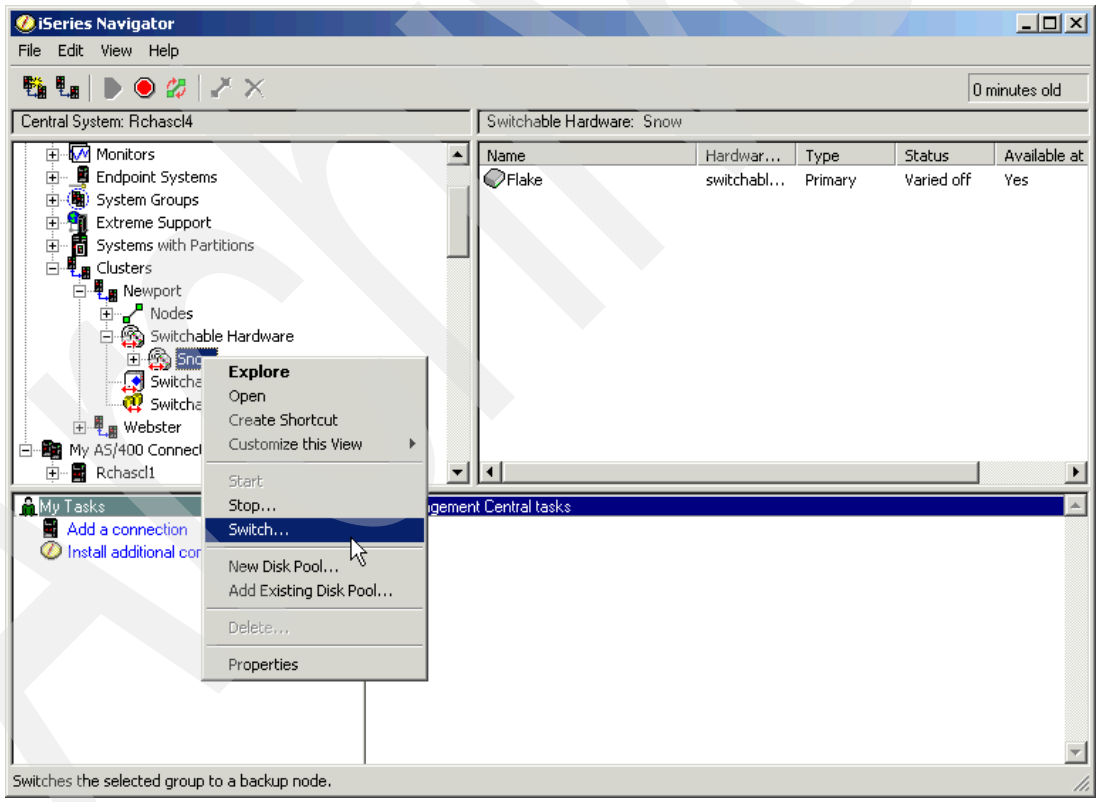


Figure 5-15 iSeries Navigator: Selecting the Switch option

6. A window (Figure 5-16) opens that prompts you to confirm your selection. Select **Yes** in response to the prompt.

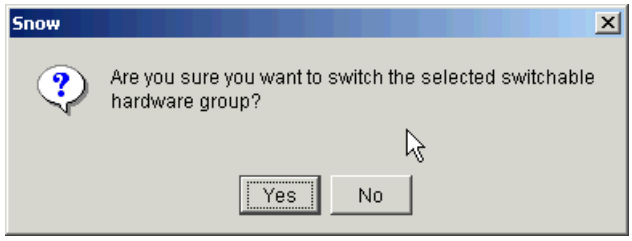


Figure 5-16 Switchable hardware confirm

7. Observe the actions of the switch process on the target machine. As the ASP device status changes, enter the following command:

   ```
   WRKCFGSTS *DEV *ASP
   ```

8. When you see *Available*, the iSeries Navigator is freed and the IASP is ready for use.

Keep in mind that an online attribute field is associated with each disk pool device description defined in the switchable hardware group (device CRG). This tells Cluster Resource Services whether to vary on (make available) the disk pool on the target node at switchover or failover. Using the cluster GUI, by default, the online attribute is set, but can be changed. If the online attribute is not set, the disk pool is left in a varied off (unavailable) state.

5.2.9 Unplanned IASP switch

A system failure or other major outage may require an unplanned switch of an IASP. This is handled in the same way as a planned switch. However, there are added delay factors due to the same abnormal IPL considerations for rebuilding database access paths that are encountered during a system IPL. Consider using systems-managed access-path protection (SMAPP) and setting it to the shortest rebuild time that is possible.

5.2.10 IASP save/restore

Refer to Chapter 11, “Backup and recovery” on page 159, for a complete discussion about saving and restoring IASPs.

5.2.11 ASP and disk unit numbering

The system assigns an ASP number to the system ASP and IASPs. The user assigns ASP numbers to user ASPs. The numbering scheme follows these guidelines:

- The system ASP is always number 1.
- User ASPs are assigned by the user in the range from 2 through 32.
- IASP numbers range from 33 through 255.
- Disk drive numbers in the system ASP and user ASPs range from 1 through 2047.
- Disk drive numbers in the range of 2048 through 4000, and greater than 6047, are reserved for future use.

5.2.12 Device domains

The construct known as a *device domain* is a subset of cluster nodes that share device resources. A device resource may be a disk pool.

A function of a device domain is to prevent conflicts that can cause the failure of an attempt to switch a resilient device between systems. The resources involved in a device domain include the structures used to identify and manage the content of the structures across the multiple systems that are involved in the domain.

Across an IASP device domain, numbers assigned to disk are unique. The separate ranges provide additional disk unit numbers. Disk numbering follows these standards:

- Disk drive numbers in IASPs are assigned numbers in the range of 4001 through 6047.
- Each node in the cluster can have up to 2047 disk units in its system and user ASPs.
- A device domain can have up to 2047 units in all IASPs within the device domain.
- There can be 128 nodes in a cluster, and each node can be its own device domain. This means that there may be as many as 128 device domains in a cluster.

Figure 5-17 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.

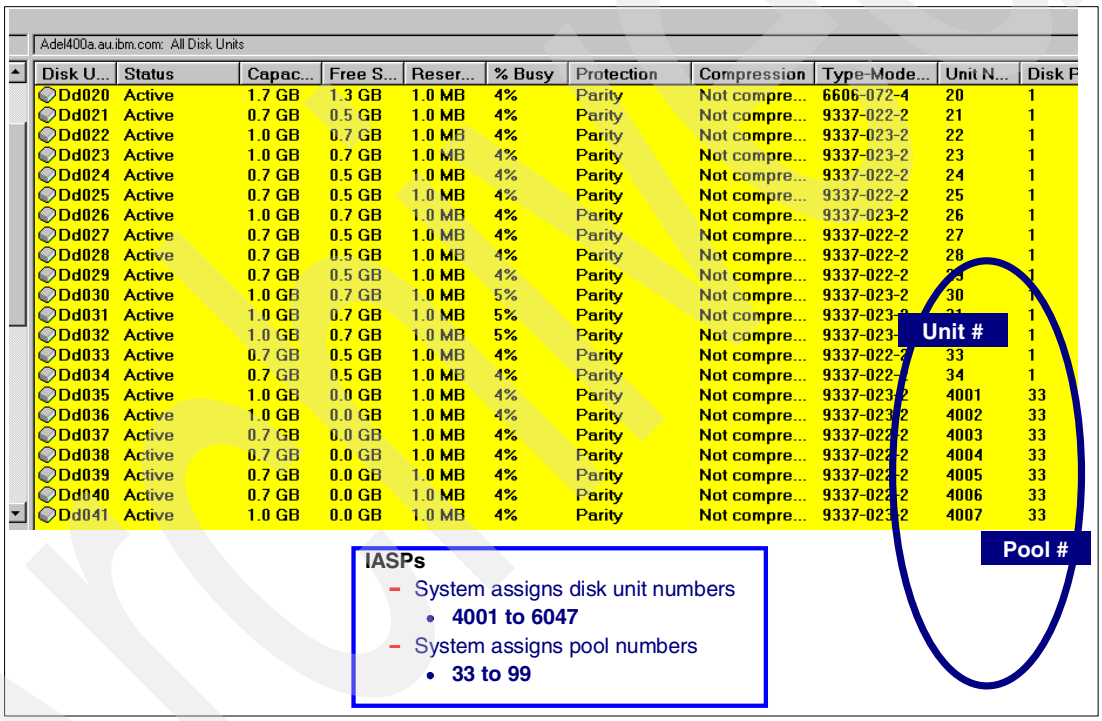


Figure 5-17 Disk unit numbering for IASPs

5.2.13 Selecting all disk units to add to a pool

An alternate path is available to create IASPs on a system. This is the *All Disk Units* path. For a new system or new environment, select this path to work with more than one pool in the same operation. Follow this alternative GUI path to create disk pools.

5.3 Disk pool management

Disk pools need to be managed as do any other objects on the system. Systems with only a system ASP do not have to worry about such tasks. Remember that with IASPs, you are dealing with multiple databases.

5.3.1 Creating a disk pool

Before you create a disk pool, be sure to complete the steps in Appendix A, “Prerequisite steps” on page 175. Then refer to Chapter 6, “Stand-alone IASP setup” on page 73, to learn how to create a disk pool.

5.3.2 Clearing the data from a disk pool

You can clear the data from a disk pool from your system. When you clear a disk pool, you destroy all data on the disk units in the pool.

If you want to clear an independent disk pool that is unavailable, you can do so when your system is fully restarted. For all other disk pools, you must restart your system using Dedicated Service Tools (DST) before you clear them.

To clear a disk pool, follow these steps:

1. In iSeries Navigator, expand **My Connections** (or your active environment).
2. Expand any iSeries server.
3. Expand **Configuration and Service-> Hardware-> Disk Units**.
4. Expand **Disk Pools**.
5. Select the disk pools you want to clear.
6. Right-click a highlighted disk pool and select **Clear**.
7. Follow the instructions on the window that opens.

5.3.3 Recovering the disk pool group

If the primary disk pool for a secondary disk pool is deleted, or if the primary disk pool is not aware of the secondary disk pool, the secondary disk pool must be re-associated with a primary disk pool. You can recover the disk pool group using iSeries Navigator.

To recover a disk pool group, follow these steps:

1. In iSeries Navigator, expand **My Connections** (or your active environment).
2. Expand any iSeries server.
3. Expand **Configuration and Service-> Hardware-> Disk Units**.
4. Double-click **Disk Pools**.
5. In the right pane, select one or more secondary disk pools that need to be associated with a primary disk pool. Right-click and select **Recover Group**.
6. On the Confirm Recover Disk Pool Group window, select the primary disk pool that you want to associate with the secondary disk pools. Only the primary disk pools that are currently owned by the system are available for selection. You cannot change the primary disk pool after you perform this action.
7. Click **Recover Group**.

5.3.4 Balancing a disk pool

You can balance the data on a disk pool in your system. Balancing a disk pool improves system performance by balancing disk capacity across all the disk units in a disk pool.

There are two ways to balance a disk pool using iSeries Navigator:

- Use the Add Disk Unit wizard when you add disk units to a pool.
- Use the Add Disk Unit wizard when you create a new disk pool.

After you start the wizard, follow the instructions provided by the wizard. If you add disk units to an existing disk pool that contains data, one of panels in the wizard asks whether you want to balance the disk units you are adding.

5.3.5 Deleting a disk pool

You can delete a disk pool from your system. When you delete a disk pool, you remove all disk units from the pool. All data on the disk units in a deleted disk pool is destroyed.

If you want to delete an independent disk pool that is unavailable, you can do so when your system is fully restarted. For all other disk pools, you must restart your system to DST before deleting them.

To delete a disk pool, follow these steps:

1. In iSeries Navigator, expand **My Connections** (or your active environment).
2. Expand any iSeries server.
3. Expand **Configuration and Service-> Hardware-> Disk Units**.
4. Expand **Disk Pools**.
5. Select the disk pools you want to delete.
6. Right-click a highlighted disk pool and select **Delete**.
7. Follow the instructions on the window that opens.

5.3.6 Converting a user-defined file system (UDFS) disk pool to a primary or secondary disk pool

You can convert UDFS disk pools to library-capable primary and secondary disk pools. Library-capable disk pools support library-based objects. You must convert UDFS disk pools if you want them to participate in a disk pool group. Before you create a secondary disk pool, you must create its primary disk pool first.

Important: You cannot reverse this action.

To create a new disk pool and add disk units to it, follow these steps:

1. In iSeries Navigator, expand **My Connections** (or your active environment).
2. Expand any iSeries server.
3. Expand **Configuration and Service-> Hardware-> Disk Units**.
4. Double-click **Disk Pools**.
5. In the right pane, select one or more UDFS disk pools to convert at the same time. Right-click the desired UDFS disk pool or pools and select **Convert to --> Secondary disk pool**.
6. On the Confirm Convert to Secondary Disk Pool window, select the primary disk pool that you want to associate with the secondary disk pools. Only the primary disk pools that are currently owned by the system are available for selection. You cannot change the primary disk pool after you perform this action.
7. Click **Convert Disk Pool**.

5.3.7 Setting the threshold of a disk pool

You can set the threshold of a disk pool by following these steps:

1. In iSeries Navigator, expand **My Connections** (or your active environment).
2. Expand any iSeries server.
3. Expand **Configuration and Service-> Hardware-> Disk Units**.
4. Expand **Disk Pools**.
5. Right-click the disk pool for which you want to change the threshold and select **Properties**.
6. Select the **Threshold** tab. On this page, specify whether you want to increase or decrease the threshold for the disk pool.

5.3.8 What to do when a disk pool fills up

When a disk pool fills up, the job that generates the data that filled up the disk pool may not be complete. The system generates an MCH2814 message indicating this condition as shown in Figure 5-18.

```
                        Additional Message Information

Message ID . . . . . :   MCH2814      Severity . . . . . :   30
Message type . . . . . :   Escape
Date sent  . . . . . :   03/22/02      Time sent  . . . . . :   17:35:06

Message . . . . :   ASP resources exceeded.
Cause . . . . . :   Resources of ASP  &1, ASP number 34, were exceeded.
    The reason code is 2.  The reason codes are:
    0 - Unspecified reason code.
    1 - ASP addresses are unavailable.
    2 - The storage space of the ASP has been exceeded.
Recovery  . . . :   For reason code 2, either destroy some objects that reside
  in the ASP or add an additional disk unit to the ASP.
```

Figure 5-18 ASP resources exceeded

This may have serious ramifications. Cancelling the offending job relieves the problem in most cases. The system does not automatically cancel the offending job. If the job is from a single-threaded JOBQ or a single threaded subsystem, other jobs behind it are held up until the offending job is handled. Possible scheduling impacts may occur.

5.3.9 Removing a disk unit from an IASP

You can remove a disk unit from an IASP when it is unavailable by following these steps:

1. In iSeries Navigator, expand **My Connections** (or your active environment).
2. Expand any iSeries server.
3. Expand **Configuration and Service-> Hardware-> Disk Units**.
4. Expand **Disk Pools**.
5. Select the disk unit to be removed, right-click, and select **Remove**.
6. Confirm the action.
7. A window opens that indicates successful completion.

5.3.10 Adding a disk unit to an existing IASP

You can add a non-configured disk unit to an IASP by following these steps:

1. In iSeries Navigator, expand **My Connections** (or your active environment).
2. Expand any iSeries server.
3. Expand **Configuration and Service-> Hardware-> Disk Units**.
4. Expand **Disk Pools**.
5. Right-click the disk pool to which to add a unit and select **Add Unit**.
6. Confirm the action.
7. If you chose to balance the data during the disk unit add, a warning window opens that indicates the balancing cannot take place until the IASP is made available.
8. A window opens that indicates successful completion.

5.3.11 Reclaim Storage and IASPs

With the introduction of IASPs, you can run Reclaim Storage (RCLSTG) on an IASP while the rest of the system keeps running. This means that multiple IASP RCLSTG processes can run concurrently, one for each IASP on the system.

V5R1 functional changes to the RCLSTG command added in support of 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. The system *must be* in a restricted state to run this.

  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 run without the system being in restricted state. In fact, because the IASP must be varied on to be seen, the system *cannot* be in a 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 ASP device requires that there can be no active users of the ASP device that is the subject of the reclaim.

6

Stand-alone IASP setup

Independent auxiliary storage pools (IASPs) support three distinct types of configurations.

- Single system, non-switchable (otherwise referred to as a stand-alone IASP)
- Multiple logical partition (LPAR), single system, switchable input/output processor (IOP)
- Multiple system, multiple LPAR, switchable tower

This chapter explains how to build a stand-alone IASP.

6.1 IASP creation prerequisites

Prior to creating a stand-alone IASP, you must complete the following prerequisites:

- iSeries Navigator must have a connection to the system or systems that are using the independent disks to create the IASP.
- The user must have *IOSYSCFG and *ALOBJ authority.

Important: Be sure to follow the steps in Appendix A, “Prerequisite steps” on page 175, prior to using iSeries Navigator.

6.2 Creating a primary disk pool as a stand-alone resource

Creating the simplest form of IASP begins with defining the disk pool and the disk units to be included in the independent ASP. You can create a disk pool and add disk units to it by using the New Disk Pool wizard. If you want to include existing user-defined file system (UDFS) disk pools in a disk pool group, see 5.3.6, “Converting a user-defined file system (UDFS) disk pool to a primary or secondary disk pool” on page 70.

To create a new disk pool group, follow these steps:

1. In iSeries Navigator, expand **My Connections** (or your active environment) and the iSeries Server desired.
2. Expand **Configuration and Service-> Hardware-> Disk Units** (see Figure 6-1).

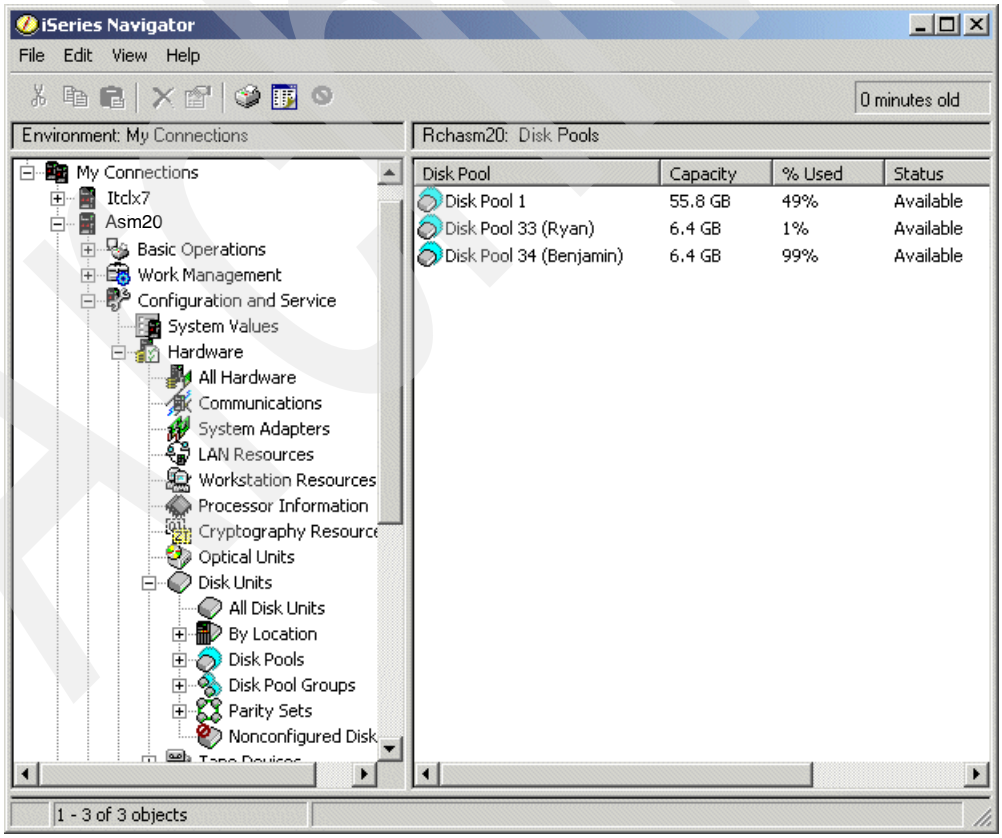


Figure 6-1 iSeries Navigator: Expanding Configuration and Service-> Hardware-> Disk Units

3. When the Service Device Sign-on display (Figure 6-2) opens, sign on using the DST QSECOFR password or equivalent.

Figure 6-2 Service Device Sign-on display

4. Right-click **Disk Pools** and select **New Disk Pool**.
5. The New Disk Pool wizard begins. Click **Next** on the New Disk Pool - Welcome window (Figure 6-3).

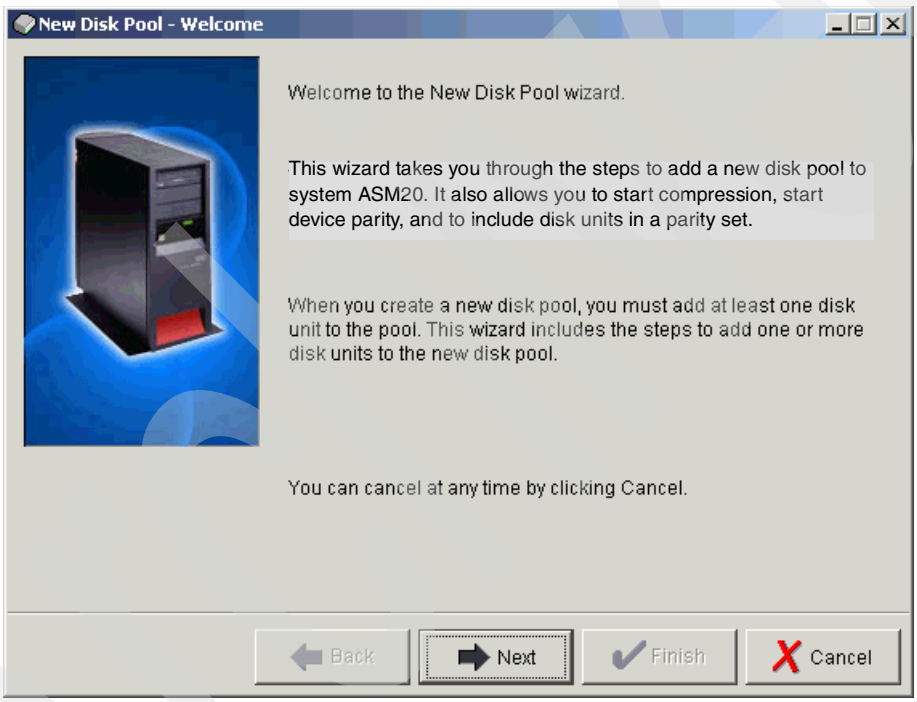


Figure 6-3 New Disk Pool: Welcome window

6. On the New Disk Pool window (Figure 6-4), select **Primary** for the Type of Disk Pool field. Then enter a name for the new primary IASP in the Disk pool field. If your disks are using either RAID or mirroring, be sure to select the **Protect the data in this disk pool** box.

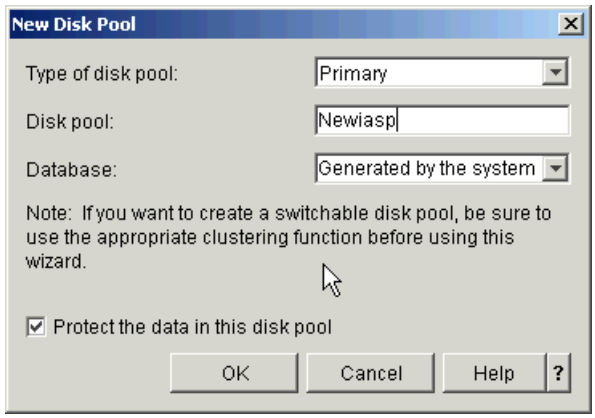


Figure 6-4 New Disk Pool window

7. Click **Next** to confirm selection of the disk pool that you just created and to add disk units to it.

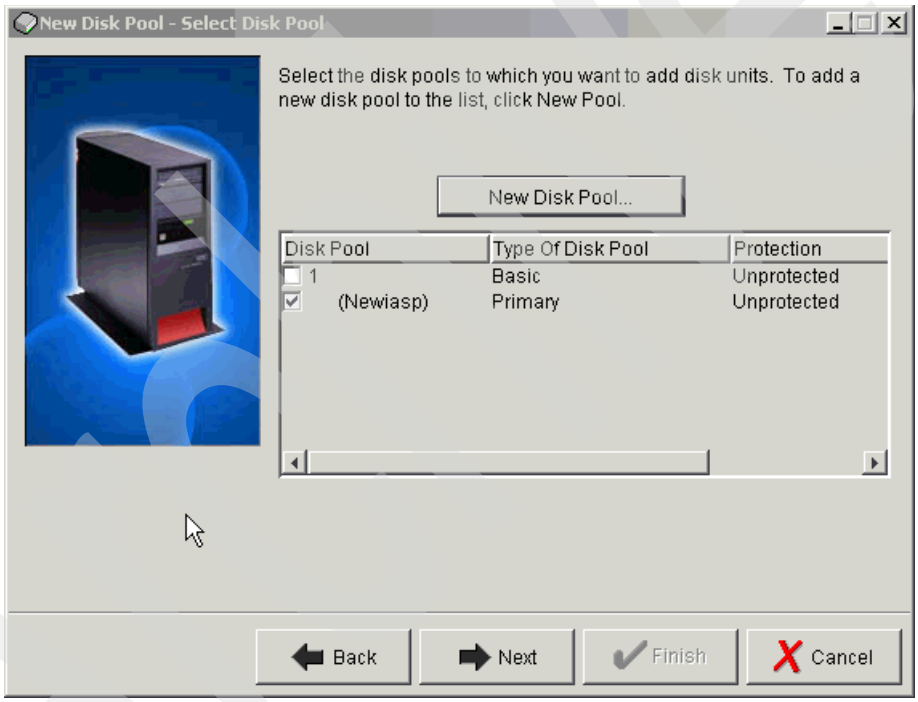


Figure 6-5 New Disk Pool: Confirming the disk pool selection

8. On the New Disk Pool - Add to Disk Pool Newiasp window (Figure 6-6), select the appropriate disk protection scheme used by the non-configured disks. Click **Next**.

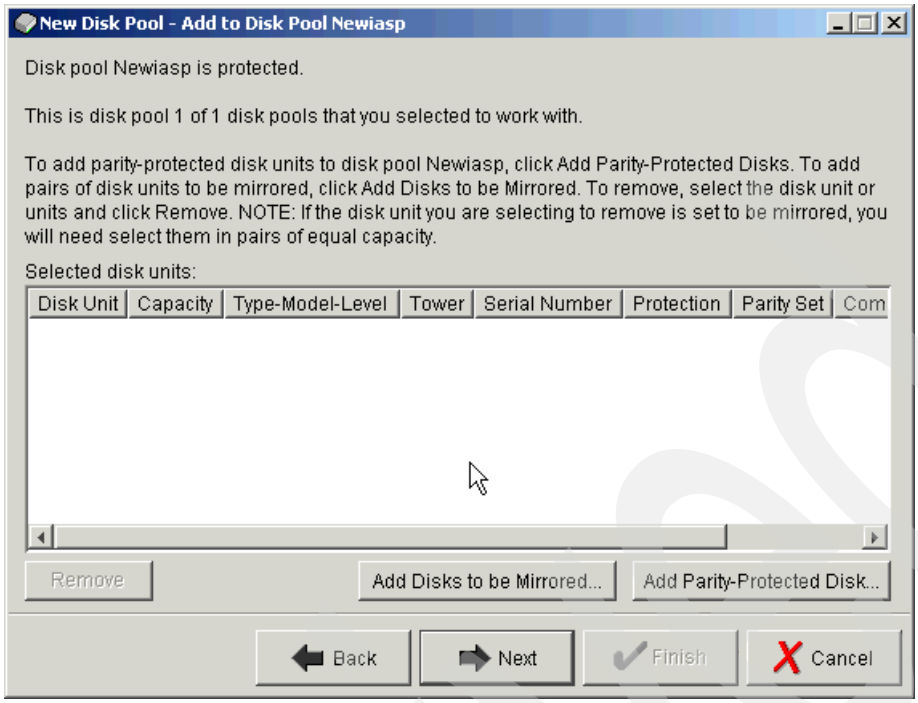


Figure 6-6 New Disk Pool protection

9. When the disk list appears in the Disk Pool Newiasp - Add Disks window (Figure 6-7), select the units to be included in the IASP. Click **Add**.

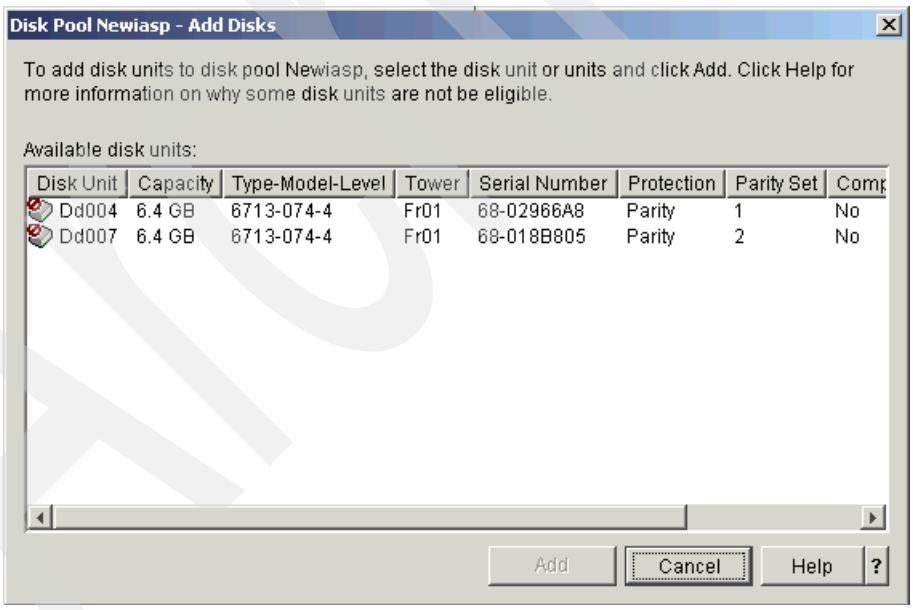


Figure 6-7 New Disk Pool Newiasp: Adding disks

10. On the New Disk Pool - Summary window (Figure 6-8), click **Finish** to confirm the addition of the disks.

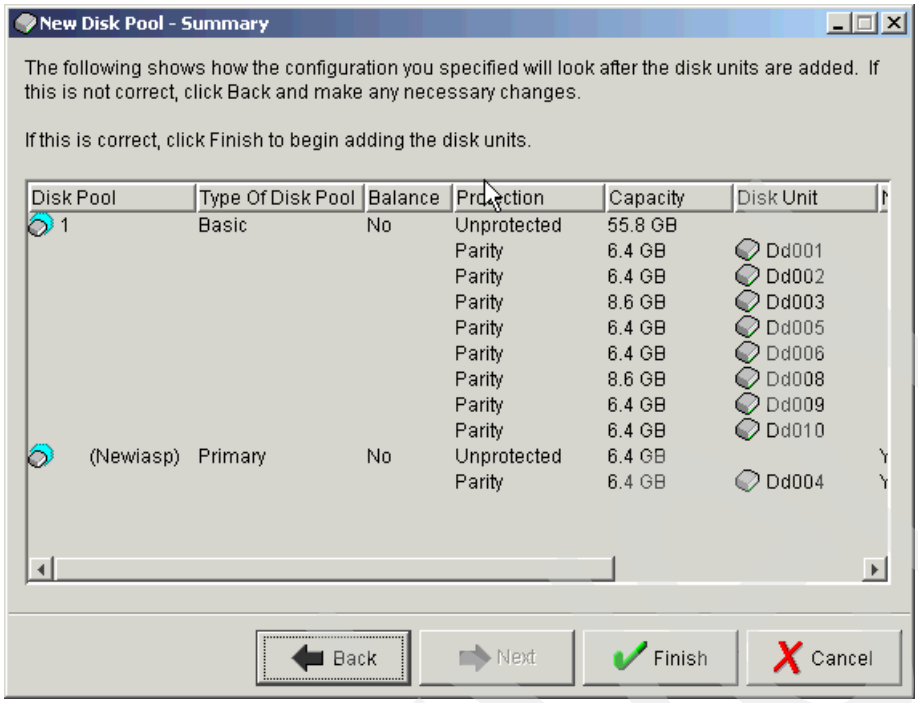


Figure 6-8 New Disk Pool – Summary

11. When the Messages window opens, observe and respond to any warnings. Then click **Continue**. The message in Figure 6-9 simply alerts you to the fact that a stand-alone disk pool (IASP) is being created.

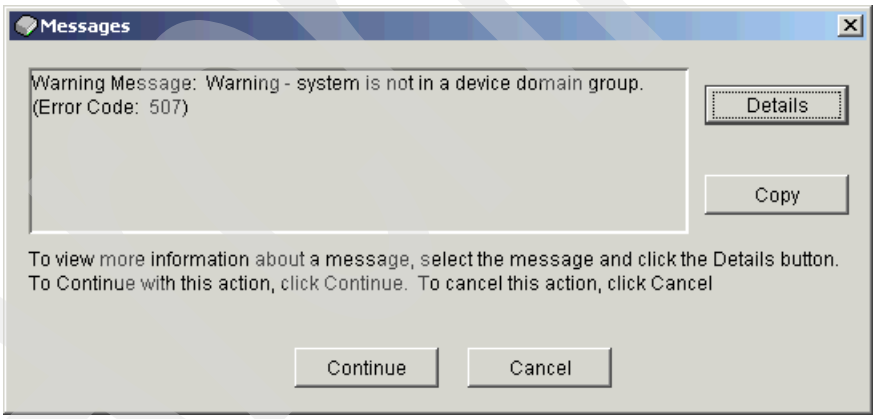


Figure 6-9 New Disk Pool: Warning message

12. As indicated by the New Disk Pool Status window (Figure 6-10), the disk pool is created.

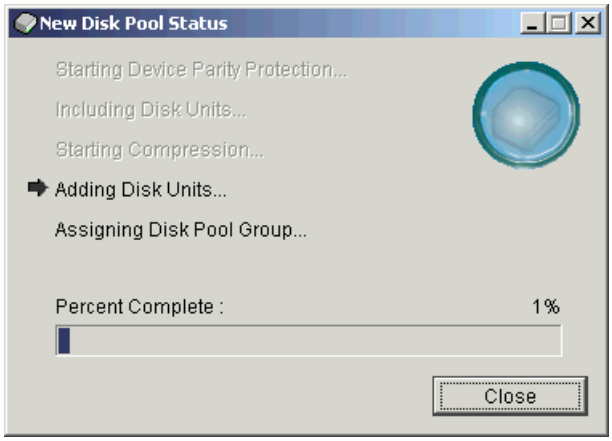


Figure 6-10 New Disk Pool Status

13. Click **OK** to confirm the successful creation of the IASP.

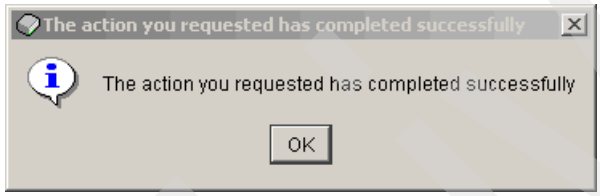


Figure 6-11 New Disk Pool completion message

6.3 Creating a new secondary disk pool

A secondary disk pool is simply another IASP. It is attached to a primary IASP and then made unavailable and available with the primary. To create the secondary disk pool, follow the same steps for creating a primary disk pool (see 6.2, “Creating a primary disk pool as a stand-alone resource” on page 74). However on the New Disk Pool window, for Type of disk pool, select **Secondary**. Then assign it to an existing primary IASP. See the example shown in Figure 6-12.

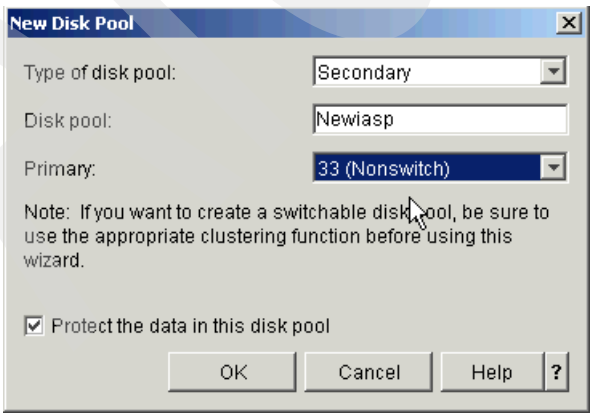


Figure 6-12 New Disk Pool: Creating a secondary disk pool

6.4 Creating a new UDFS disk pool as a stand-alone resource

A UDFS disk pool was introduced at V5R1 of OS/400. It support integrated file system (IFS) object constructs only. This type of IASP is useful if QSYS.LIB type objects are not required.

To create the UDFS disk pool, follow the same steps that are explained in 6.2, “Creating a primary disk pool as a stand-alone resource” on page 74. When you reach the New Disk Pool window, this time select **UDFS** for Type of disk pool as shown in Figure 6-13.

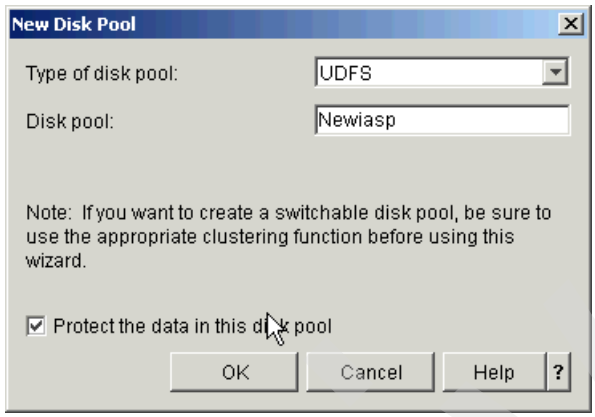


Figure 6-13 New Disk Pool: Creating a UDFS disk pool

7

Switchable setup

As explained in Chapter 6, “Stand-alone IASP setup” on page 73, there three types of configurations for independent auxiliary storage pools (IASPs). Chapter 6 covers the first type of configuration (single system, non-switchable). This chapter examines the following types:

- Multiple logical partition (LPAR), single system, switchable input/output processor (IOP)
- Multiple system, multiple LPAR, switchable tower

This chapter also shows how to use Cluster Resource Services when switching an IASP to another system.

7.1 Prerequisites for creating IASPs

To accomplish any tasks in regard to hardware using iSeries Navigator, you *must* complete the steps in Appendix A, “Prerequisite steps” on page 175.

You must also complete the following requirements:

- The non-configured disk units which make up the new IASP must be owned by a system. For a multiple LPAR, single system, switchable IOP, the disk unit’s IOP must be defined as *own bus shared*. For a multiple system, multiple LPAR, switchable tower, the disk unit’s IOPs within the tower must be defined as *own dedicated* if they are defined to an LPAR.
- iSeries Navigator must have connections to the systems that are using the independent disks. Management Central’s Central System must point to the owning system of the non-configured disk units.
- Product Option 41 - OS/400 - HA Switchable Resources (Licensed Program Product 5722-SS1) is required when you set up simple clustering, for switching between two systems.
- Cluster Resource Services are used to switch independent disk pools between multiple systems. For a full explanation of clustering, consult *Clustering and IASPs for Higher Availability on the IBM @server iSeries Server*, SG24-5194, or look in the iSeries Information Center on the Web at:

  `http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html`

  For the purposes of this redbook, we create a simple cluster to facilitate a switchable IASP.

7.2 Installing iSeries Navigator component logical systems

This optional feature of iSeries Navigator is required to work with switchable IASPs. After iSeries Navigator is installed on the PC and Option 41 of 5722-SS1 is installed on the iSeries machine, you can install the component as explained here:

1. Open iSeries Navigator.
2. Select **My AS/400 Connections**.
3. Right-click and select **Install Options-> Selective Setup** as shown in Figure 7-1.

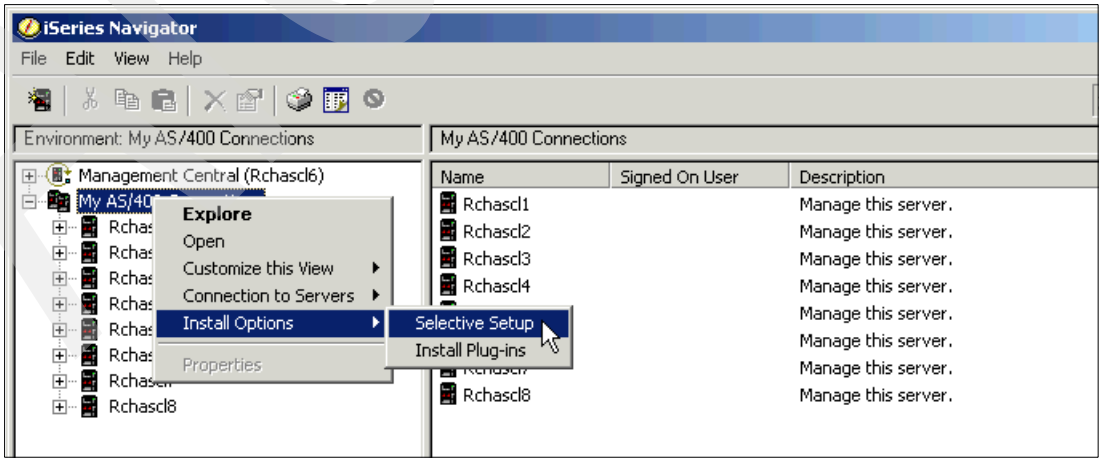


Figure 7-1 iSeries Navigator: Selecting Selective Setup

4. On the Selective Setup window (Figure 7-2), select the **Server** option and in the drop-down list, select the server that is the management CENTRAL SYSTEM. Click **OK**.

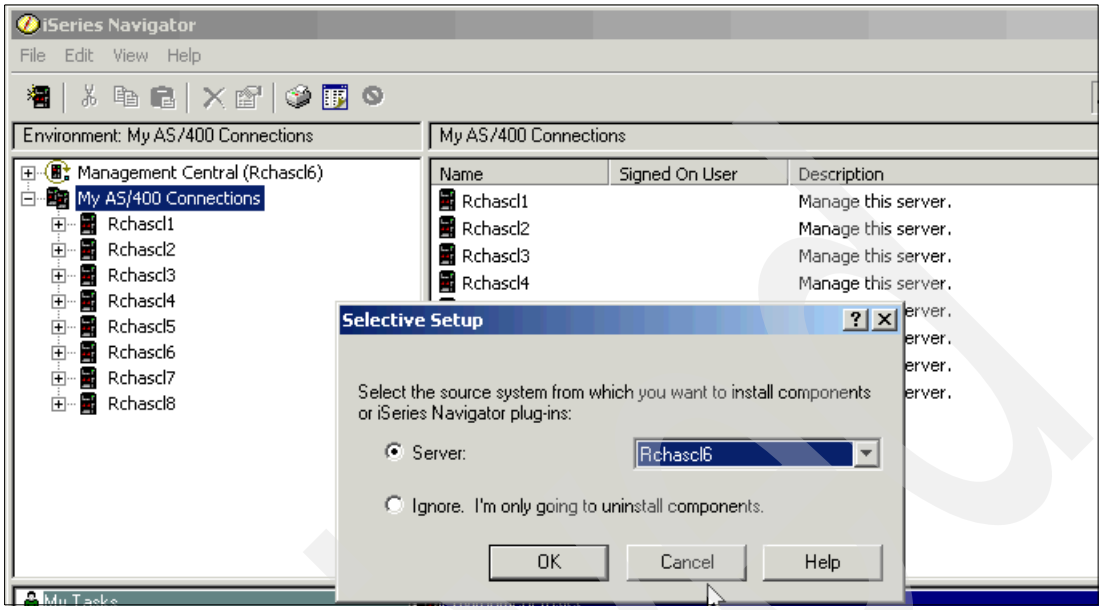


Figure 7-2 iSeries Navigator: Selective Setup window

5. On the Signon to iSeries window (Figure 7-3), sign on using QSECOFR. Then, click **OK**.

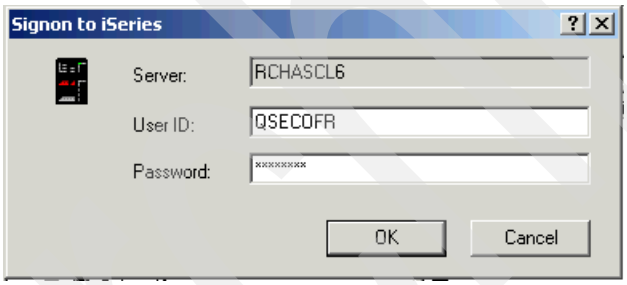


Figure 7-3 iSeries Navigator: Signing on to the iSeries server

6. The Selective Setup panel opens. Click **Next**.

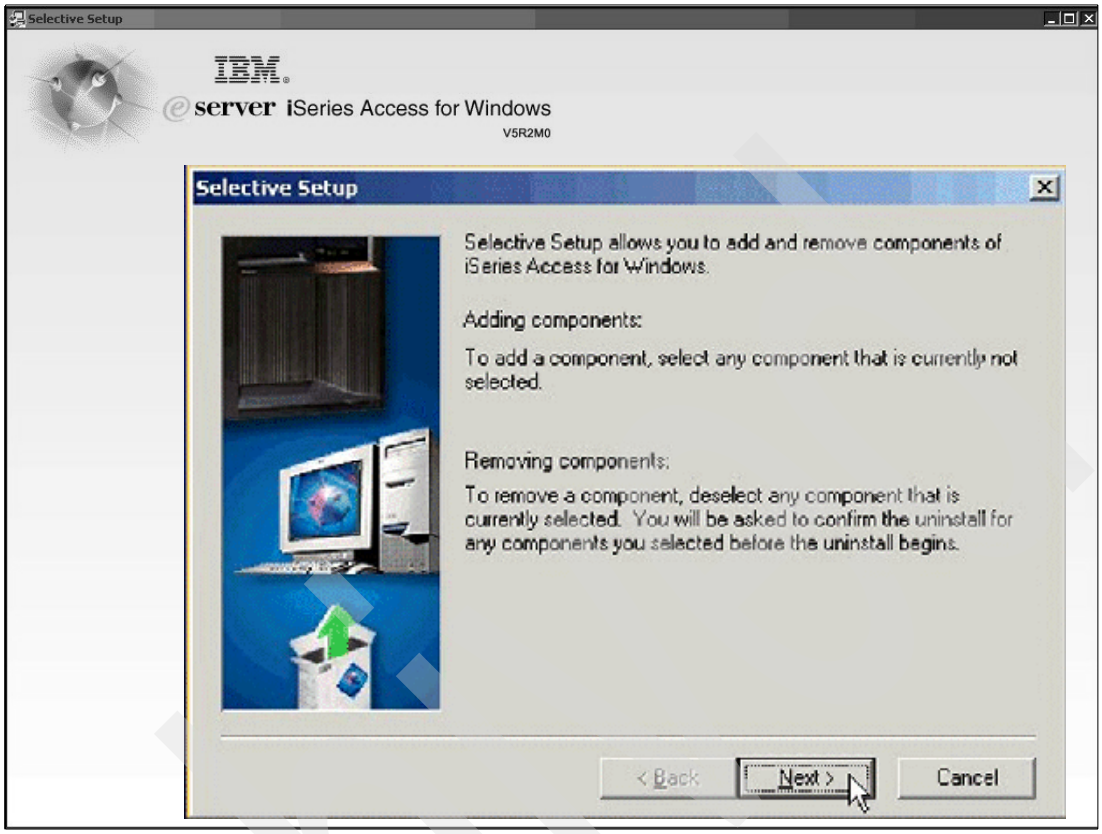


Figure 7-4 iSeries Navigator: Selective Setup window

7. The Signon to iSeries Netserver window (Figure 7-5) opens. Sign on to the iSeries NetServer using QSECOFR. Click **OK**.

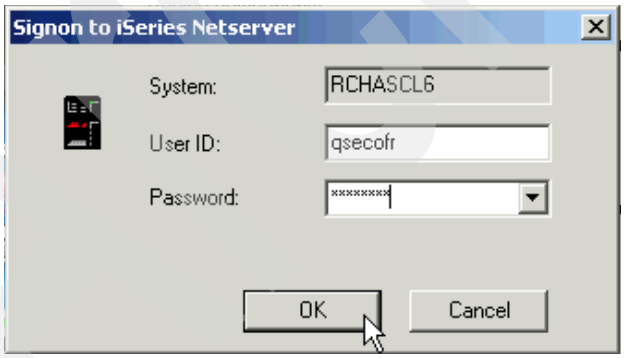


Figure 7-5 iSeries Navigator: Signing on to the iSeries Netserver

8. On the Components Cannot Be Installed window (Figure 7-6), click **Next** since this condition is irrelevant to the operation we are performing.

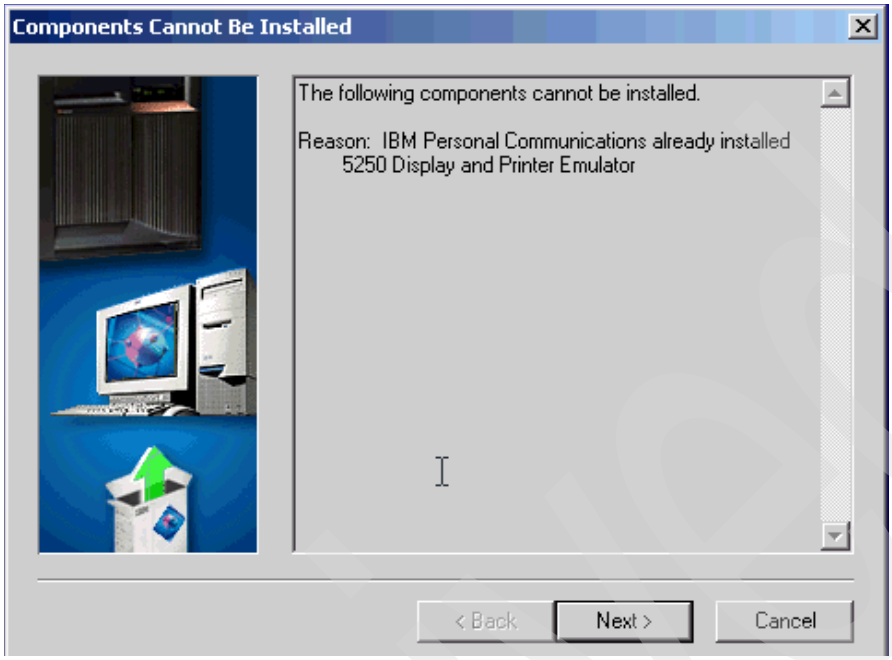


Figure 7-6 iSeries Navigator: Components Cannot Be Installed window

9. On the Component Selection window (Figure 7-7), scroll down and select the **Logical Systems** check box. This causes the cluster support for iSeries Navigator to download. Then click **Next**.

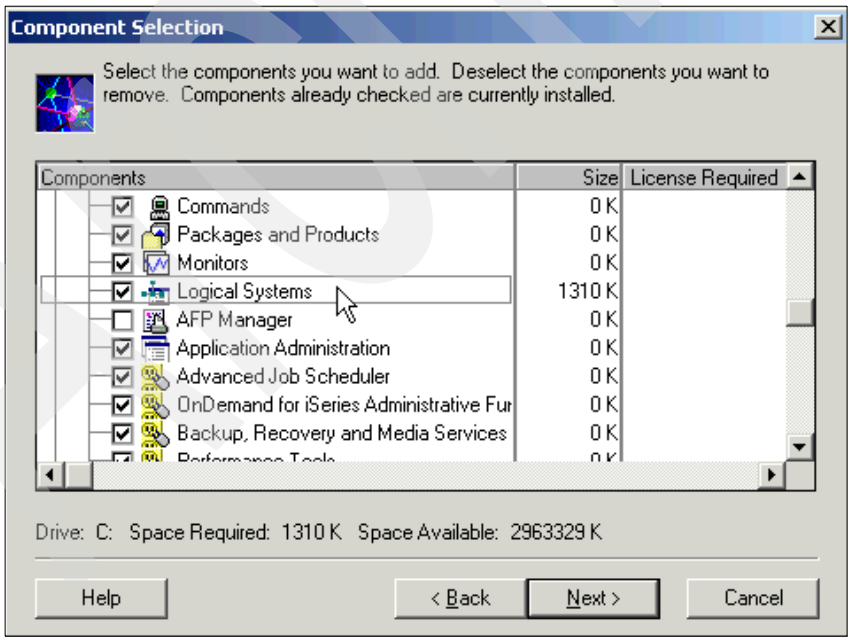


Figure 7-7 iSeries Navigator: Component Selection window

10. On the Start Copying Files window (Figure 7-8), click **Next** to confirm the selection.

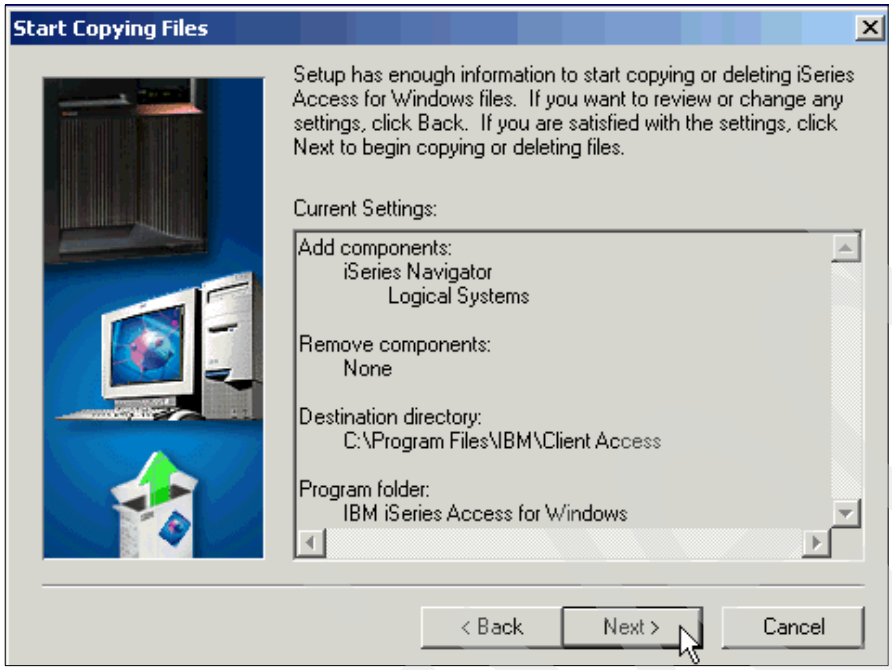


Figure 7-8 iSeries Navigator: Start Copying Files window

11. As shown in Figure 7-9, the software download proceeds.

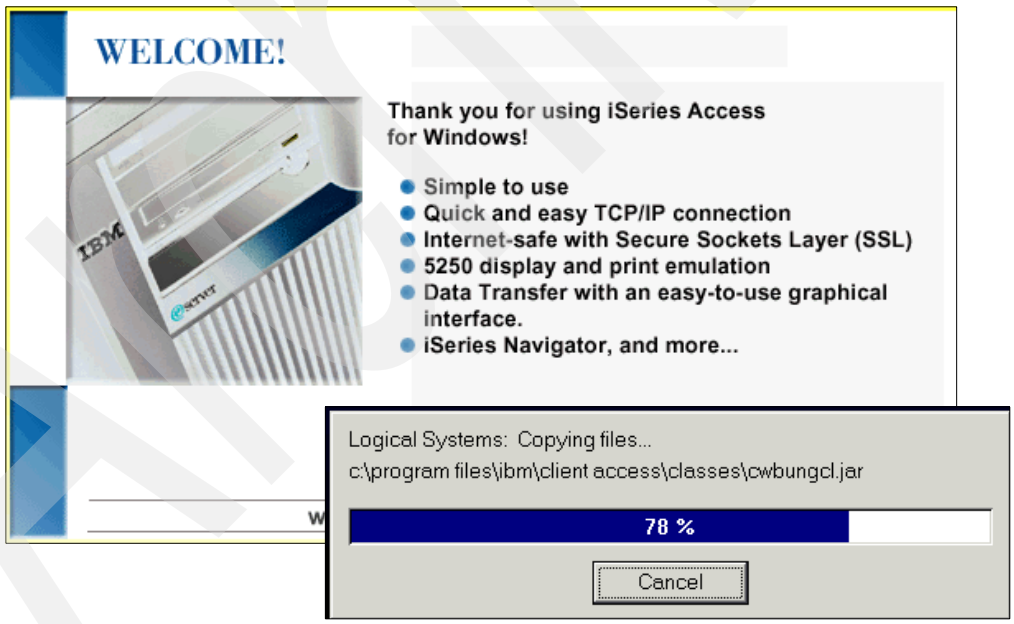


Figure 7-9 iSeries Navigator: Download in progress

12. The Install Completed window (Figure 7-10) opens. You may choose to view the readme file or select the option to add shortcut to desktop box if it is not already selected. Then click **Next**.

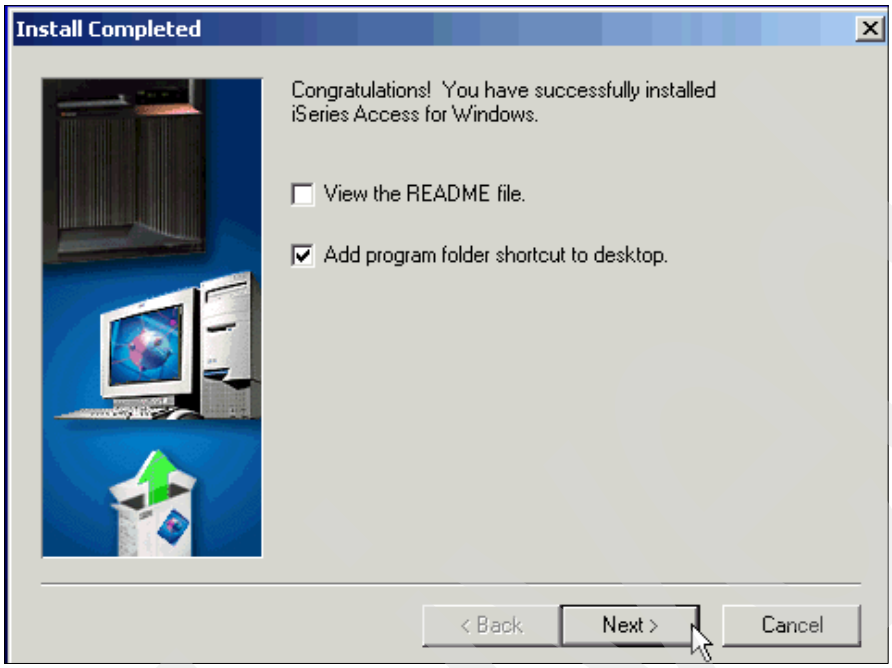


Figure 7-10 iSeries Navigator: Install Completed window

13. On the Setup Completed window (Figure 7-11), click **Finish** and the installation is complete.

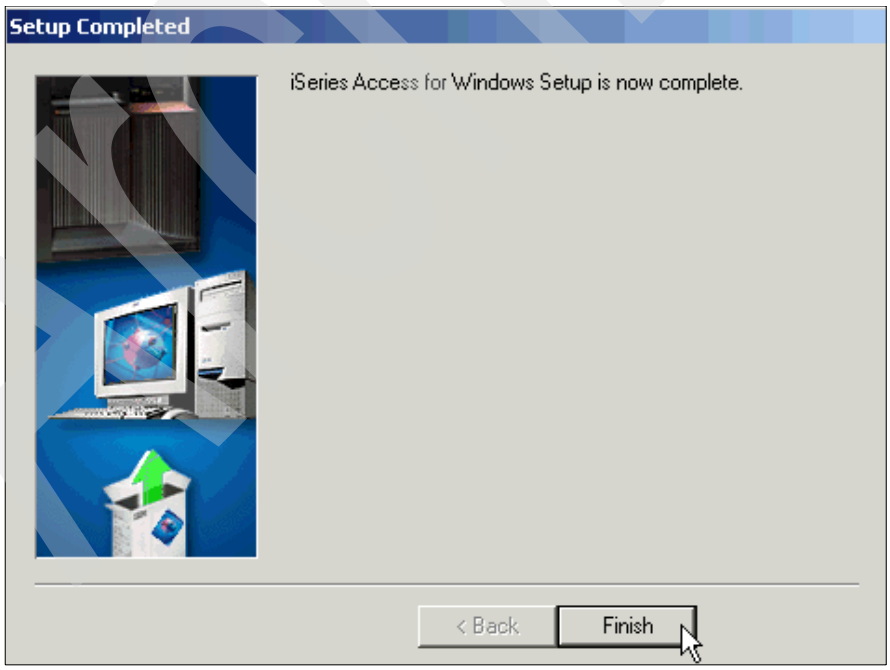


Figure 7-11 iSeries Navigator: Setup Completed window

7.3 Creating a two-node cluster

This section shows you how to create a simple two primary/backup cluster using iSeries Navigator. The TCP server *INETD must be running, and Option 41 of 5722-SS1 must be installed with its license key applied to continue. Follow these steps:

1. Open iSeries Navigator and expand **Management Central**.

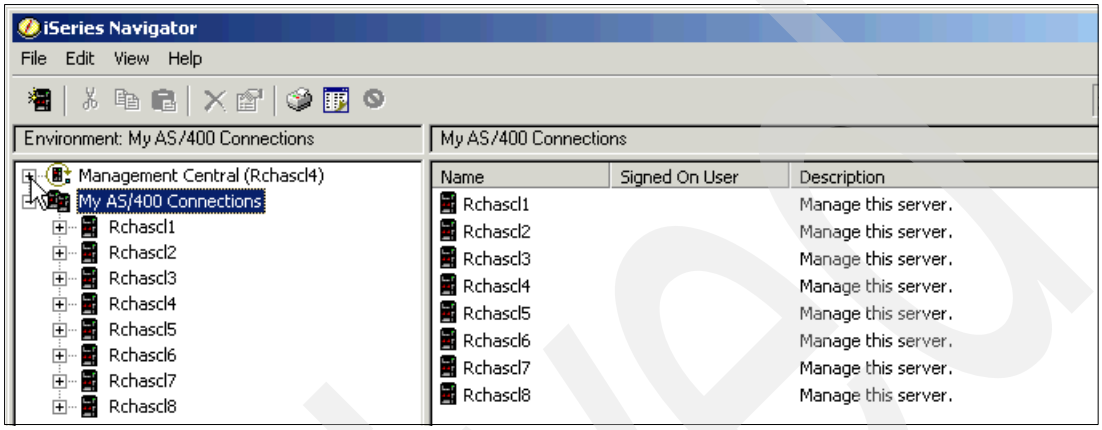


Figure 7-12 Expanding Management Central

2. Sign on to the Management Central “Central System” as shown in Figure 7-13.

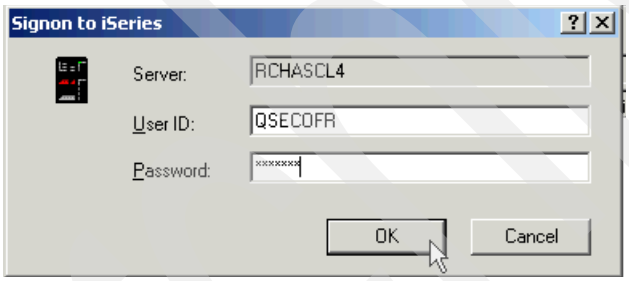


Figure 7-13 iSeries Navigator: Signing on to the central system

3. Select **Clusters**, right-click, and select **New Cluster** as shown in Figure 7-14.

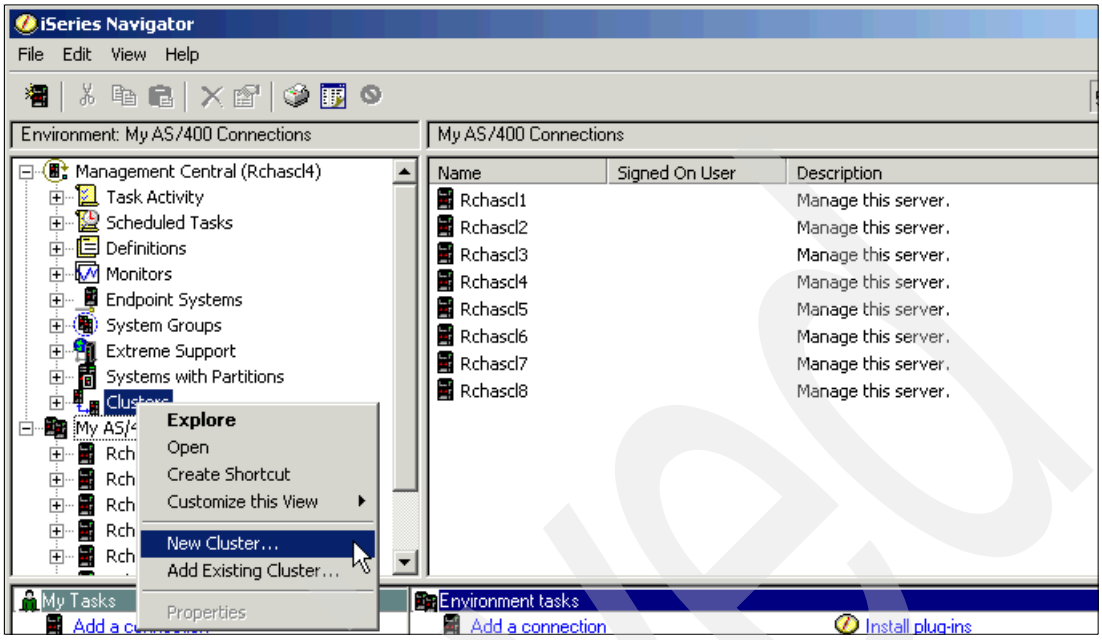


Figure 7-14 iSeries Navigator: Selecting a new cluster

4. On the New Cluster - Welcome window (Figure 7-15), click **Next**.

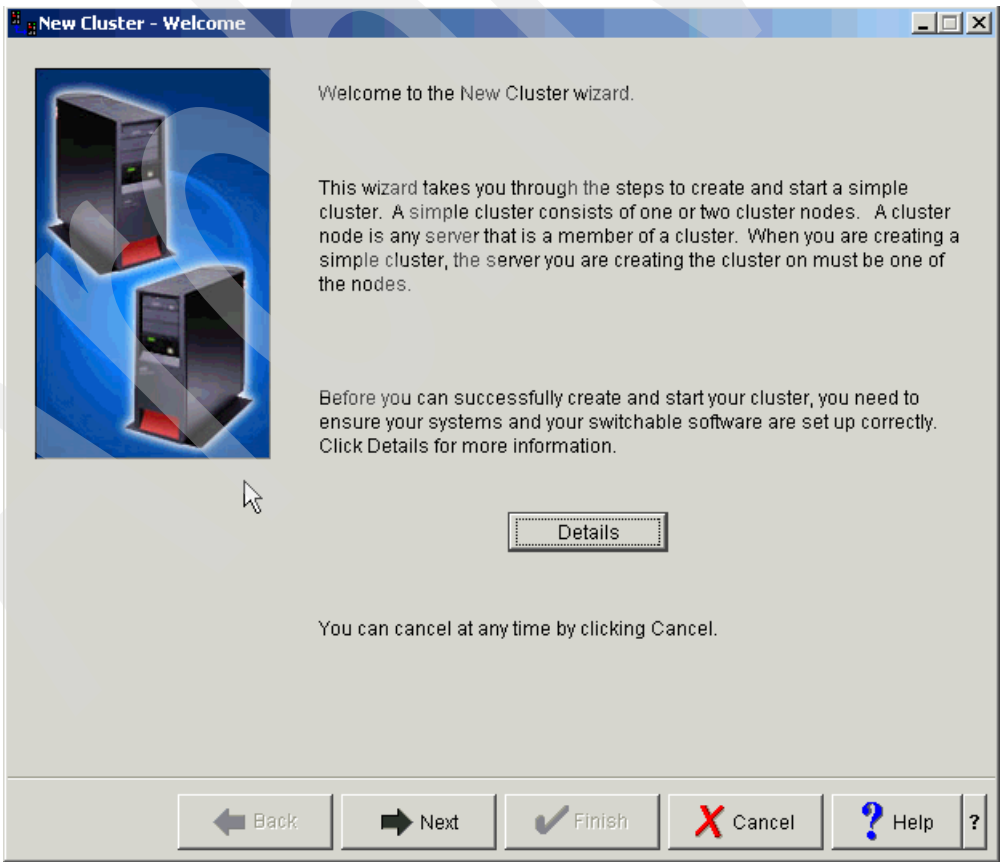


Figure 7-15 New Cluster - Welcome window

5. On the New Cluster - Specify Cluster Name window (Figure 7-16), type a name for your cluster and click **Next**.

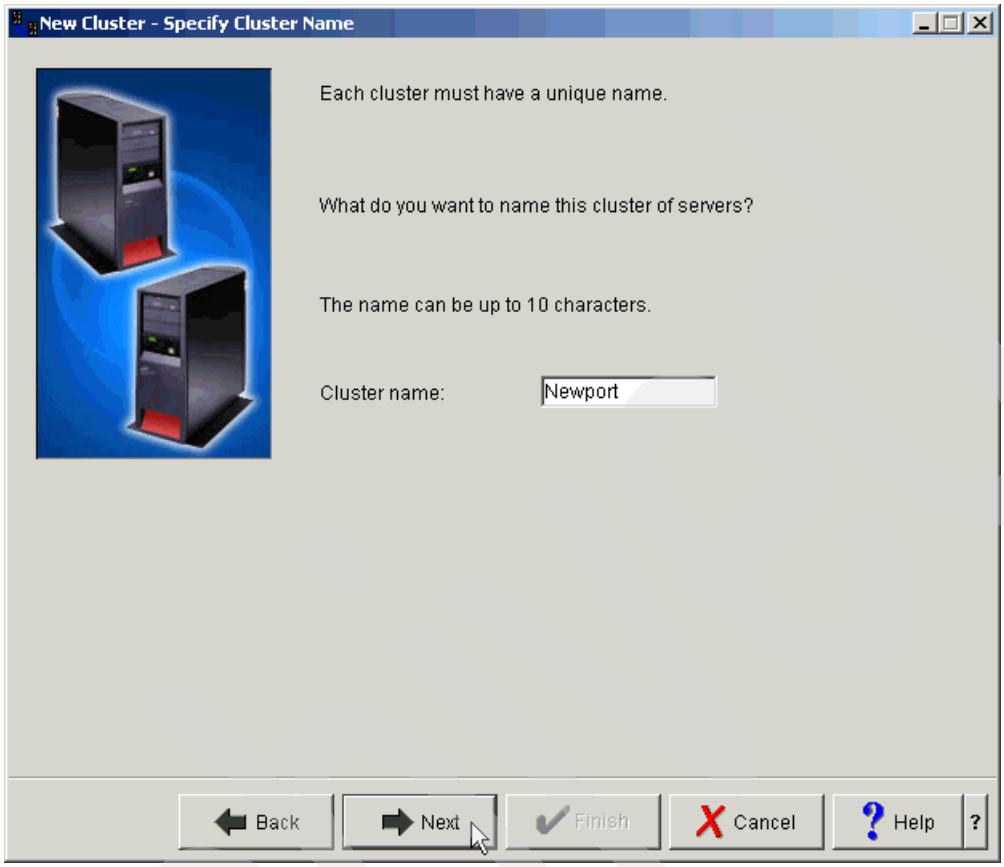


Figure 7-16 New Cluster - Specify Cluster Name window

6. On the New Cluster - Specify Node window (Figure 7-17), follow these steps:
   a. Under Node name and Server, enter the name of the system that is to be the primary system in the cluster. Although the node name may be unique, for simplicity we make it the same as the system or server name.
   b. Enter up to two IP addresses to be used for cluster heartbeat and high-speed message passing. The second address is optional, but recommended, since it is used if “Cluster interface IP address 1” becomes unavailable.
   c. Click **Next**.

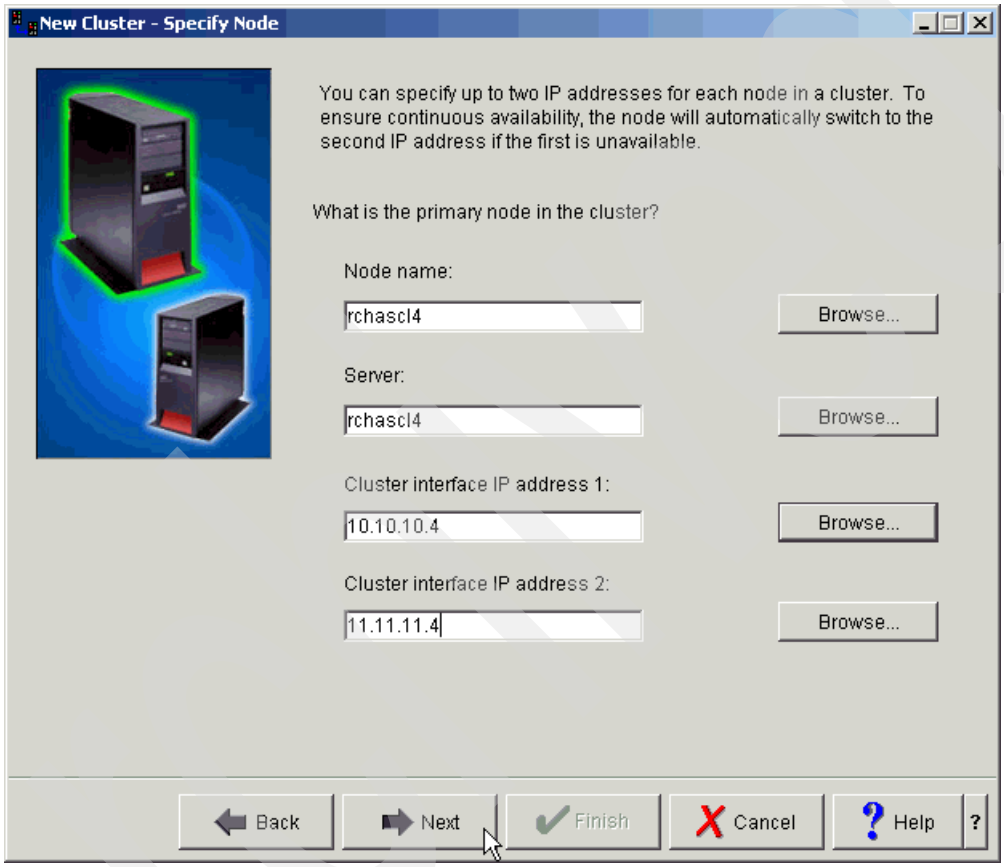


Figure 7-17 New Cluster - Specify Node window

7. On the New Cluster - Specify Backup Node window (Figure 7-18), enter the node name, server, and up to two IP addresses for the second node in the cluster for the backup system. We switch the IASP to this system in our examples. Click **Next**.

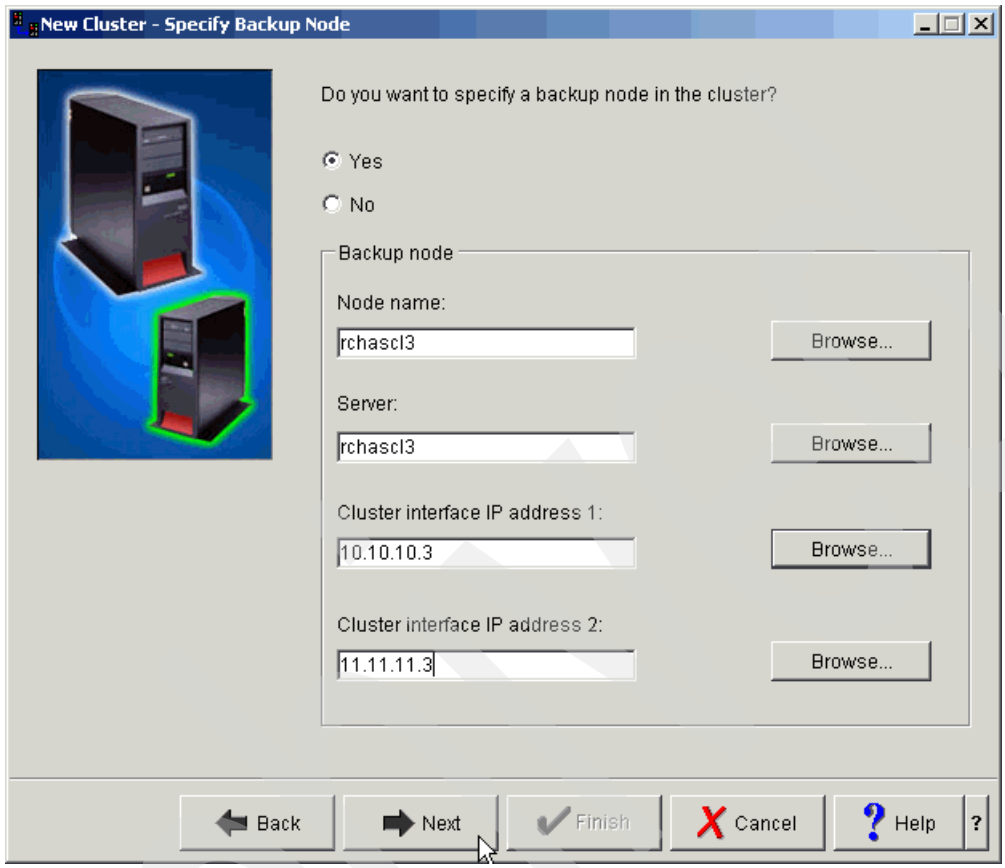


Figure 7-18 New Cluster - Specify Backup Node window

8. The Signon to the Server window (Figure 7-19) opens asking you to sign on to the backup server. This sign-on panel is quite small and often appears in the top left-hand corner of your desktop. If you are performing other tasks and have a busy desktop, it is easy to miss this sign-on panel.

   Sign on to the backup node system with QSECOFR for that system. Click **OK**.

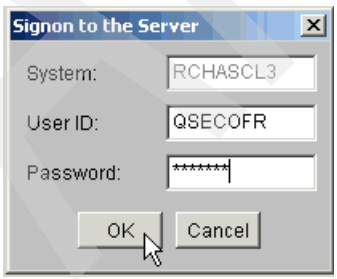


Figure 7-19 New cluster wizard: Signing on to the backup node

9. The New Cluster - No Switchable Software Found window (Figure 7-20) opens indicating that no HABP software solutions are involved. This is a normal window in this process, so there is no need to be alarmed. The primary cluster node checks for any existing switchable software. Click **Next** on this window.

It is possible (and sometimes desirable) to create some disk pools on one, but not more than one, node (system) of the (future) cluster, before the cluster is created. If disk pools already exist on a system used to create a cluster, and the cluster is being created with the GUI wizard, the system that already has disk pools must be entered as the “primary” system. “Primary” on the preceding windows does not mean “primary” in the sense of being the SPCN “owner” of a switchable tower. The owning node (system) of a switchable tower can be set later as shown in Figure 7-28 on page 99.

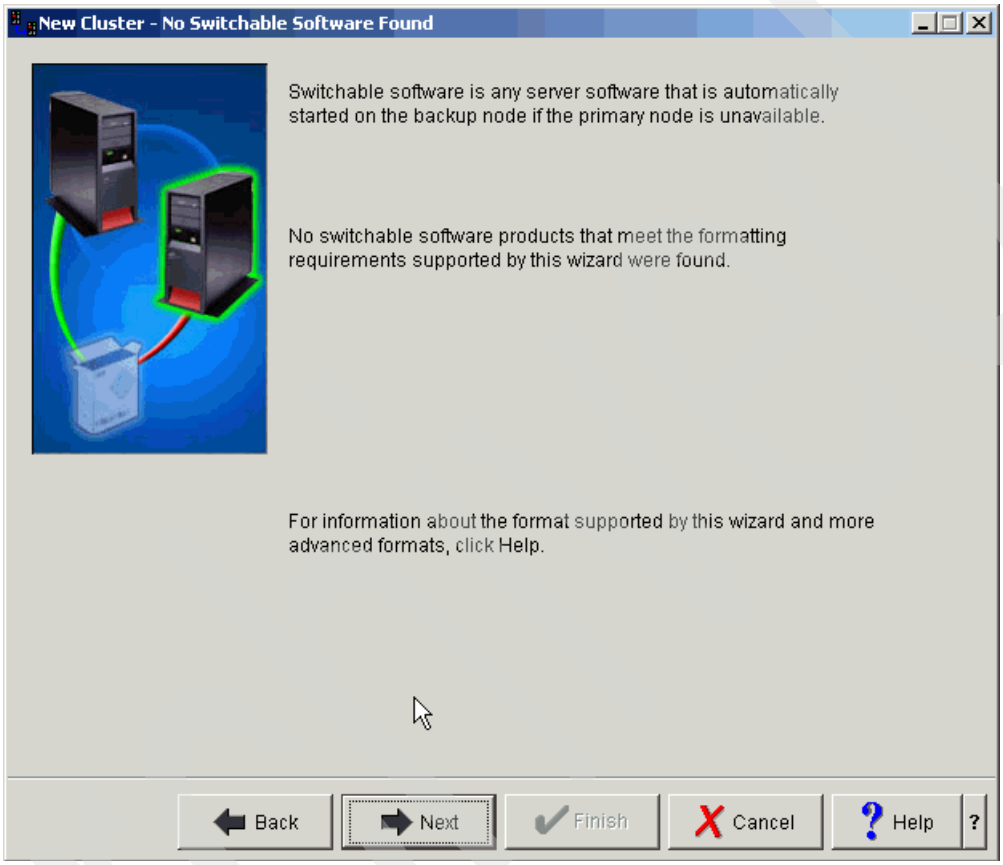


Figure 7-20 New cluster wizard, check for existing switchable software

10. The cluster is created as indicated by the Creating Cluster window (Figure 7-22).

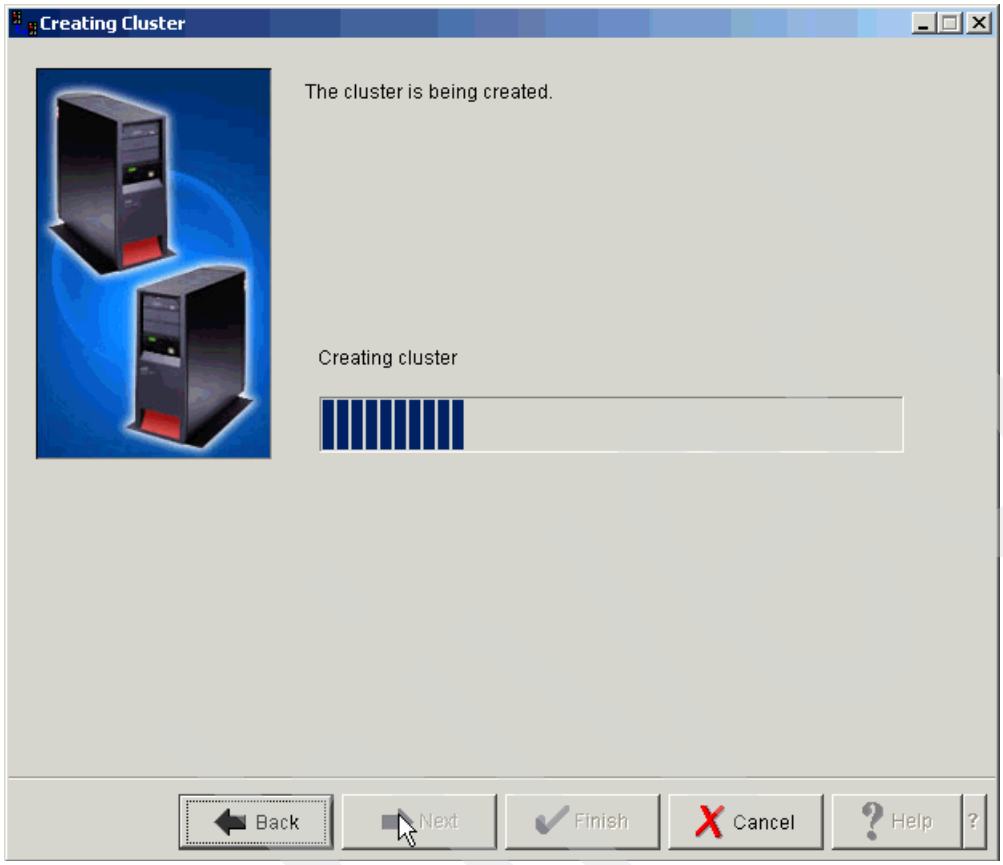


Figure 7-21 Creating Cluster window

11. The next Creating Cluster window (Figure 7-22) indicates that the cluster was successfully created. Click **Next**.

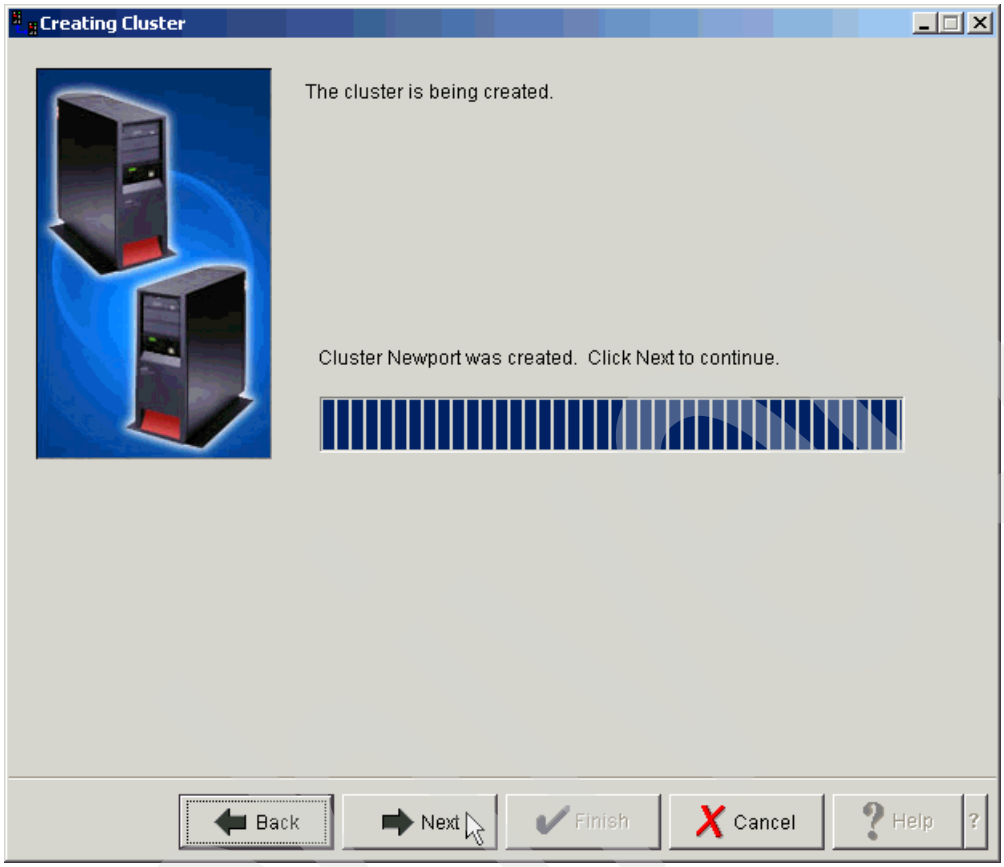


Figure 7-22 Creating Cluster window indicating a successful creation

12. On the New Cluster - Summary window (Figure 7-23), click **Finish**.

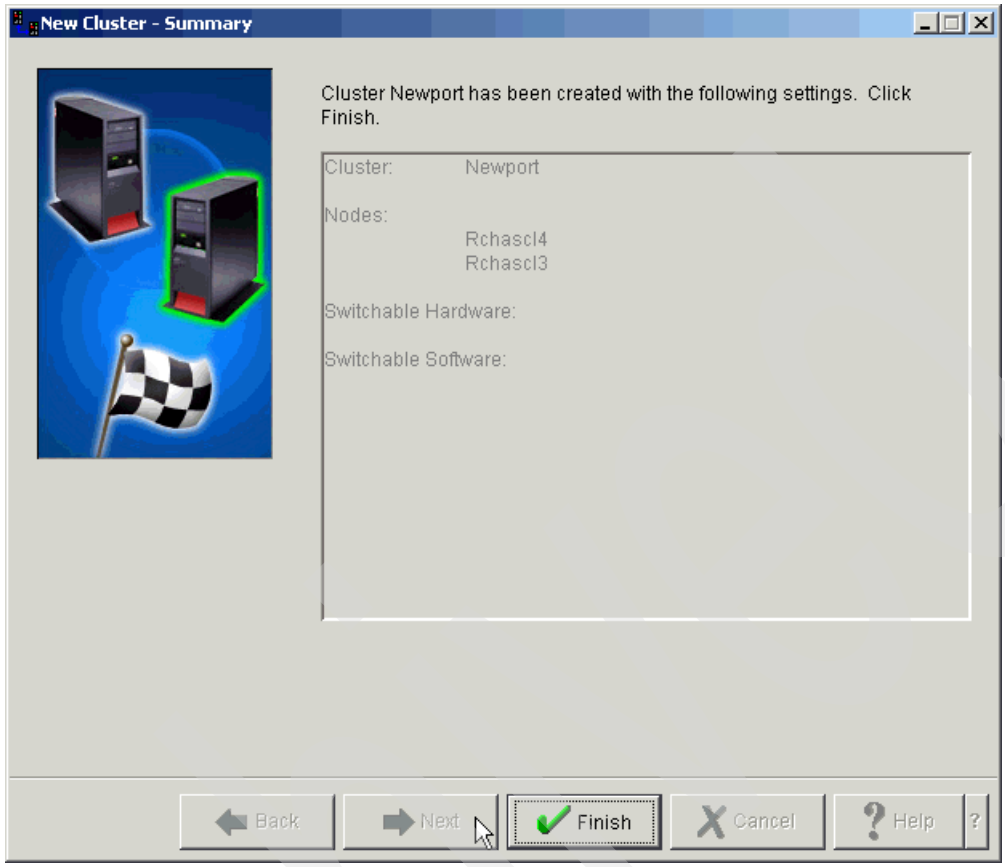


Figure 7-23 New Cluster - Summary window

13. On the iSeries Navigator window, expand **Clusters**. You now see your new cluster name (Figure 7-24).

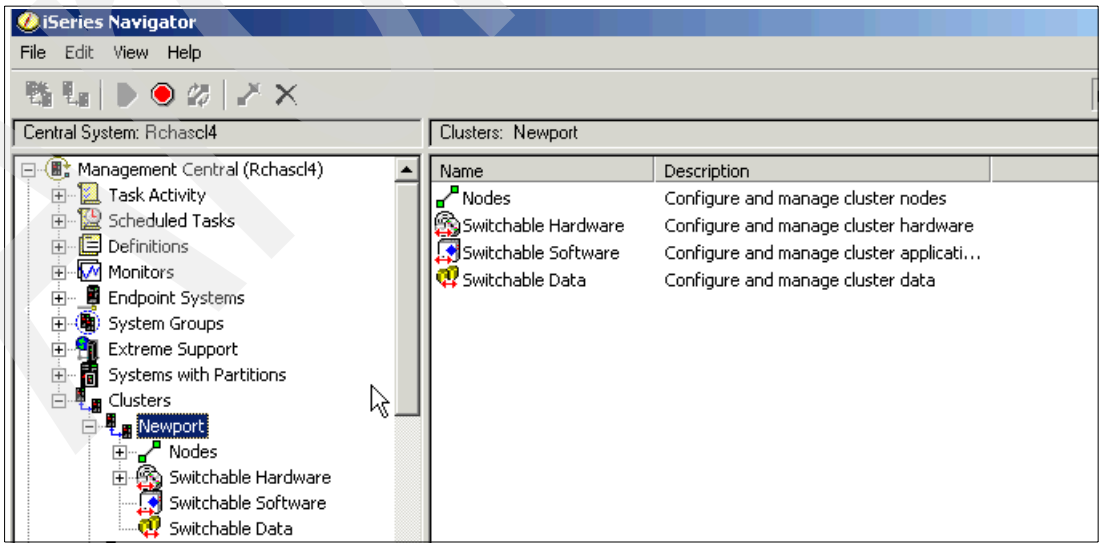


Figure 7-24 iSeries Navigator: Expanding Clusters to see your new cluster name

7.4 Creating a switchable ASP

This section explains the steps for creating a switchable ASP.

7.4.1 Creating an IASP switchable between multiple LPARs of a single system

The multiple LPAR, single system, switchable IOP configuration is supported for any iSeries server that allows logical partitioning. All of the hardware considerations for single system, non-switchable IASP apply. In addition, the original owning system's buses containing the IOPs to be switched must be defined as *own bus shared*. The target LPARs bus definition must be set to *use bus shared*.

For more information about iSeries LPARs, see the IBM Redbook *LPAR Configuration and Management: Working with IBM @server iSeries Logical Partitions,* SG24-6251.

7.4.2 Creating a switchable hardware group

You must create a definition in the cluster to make both system "aware" when using switchable devices. This is called the *switchable hardware group*, also referred to as the *device cluster resource group* (CRG). This section shows you how to create one in the previously created cluster, using iSeries Navigator. Follow these steps:

1. Ensure the non-configured disk IOPs are located on the primary cluster node.
2. Open iSeries Navigator. Expand **Clusters** and the name of your previously created cluster (Figure 7-25).

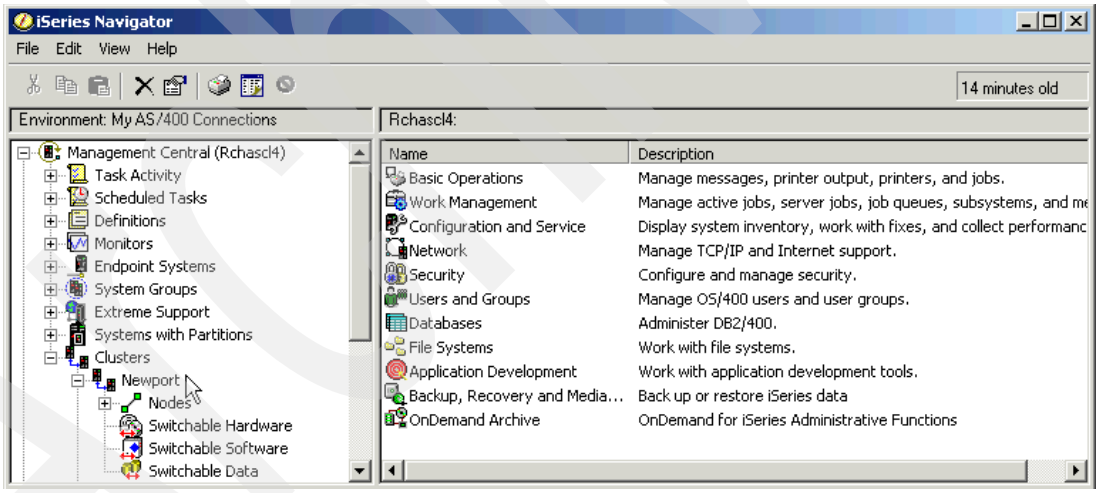


Figure 7-25 iSeries Navigator: Locating Clusters under Management Central

3. Select **Nodes**. Ensure both nodes are started.
4. To start a node, select the node, right-click, and select **Cluster-> Start** (Figure 7-26). The status of nodes changes to *Started*.

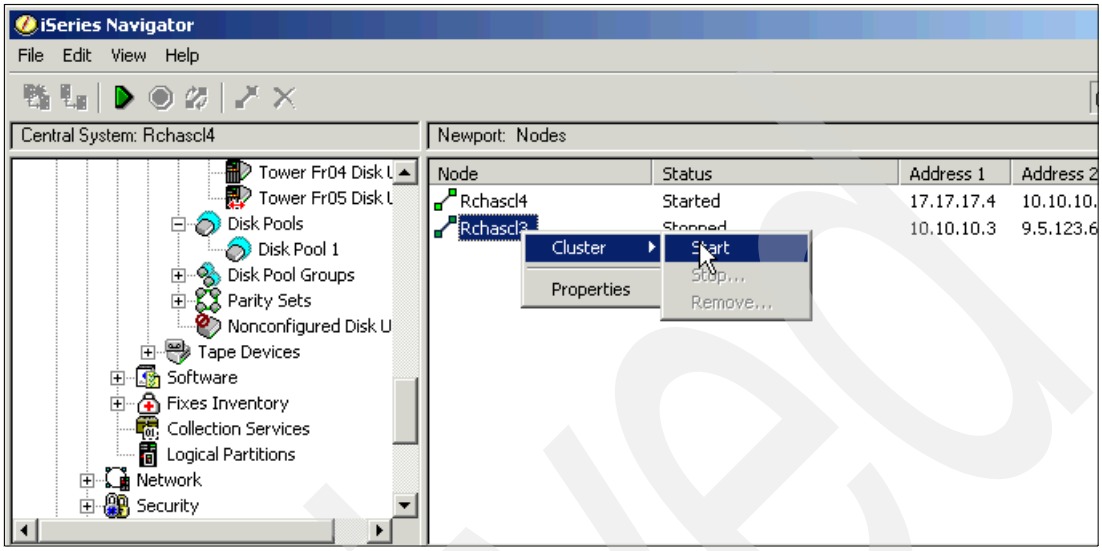


Figure 7-26 iSeries Navigator: Starting the cluster

5. Select **Switchable Hardware**, right-click, and select **New Group** (Figure 7-27).

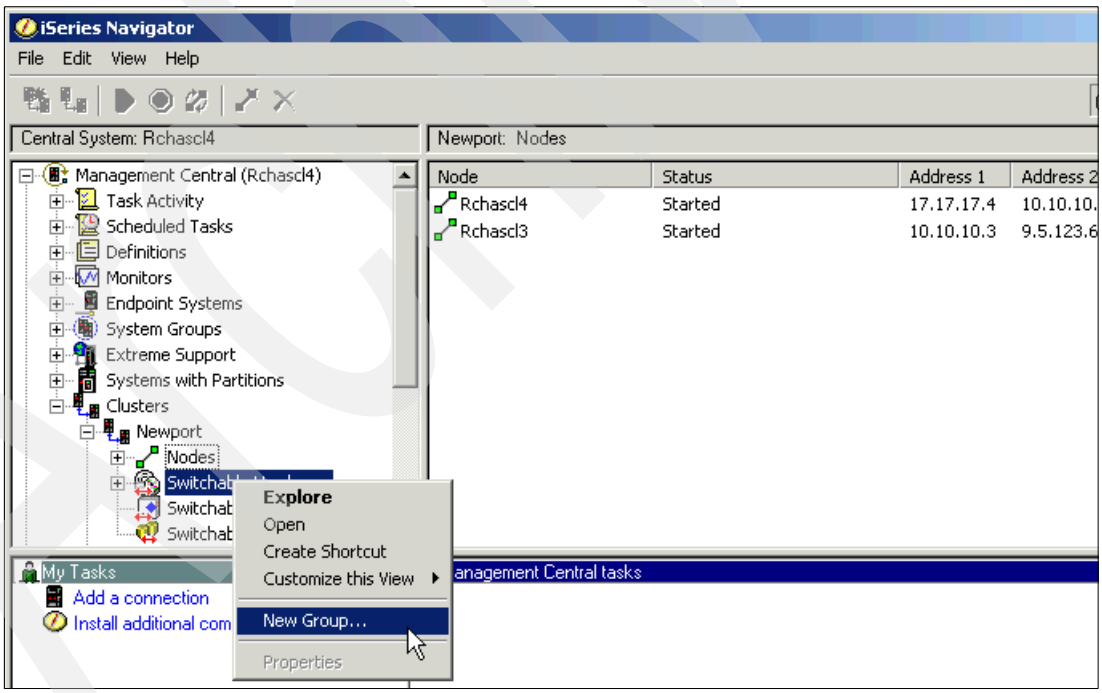


Figure 7-27 iSeries Navigator: Selecting New Group

6. On the New Group - Welcome window (Figure 7-28), click **Next**.

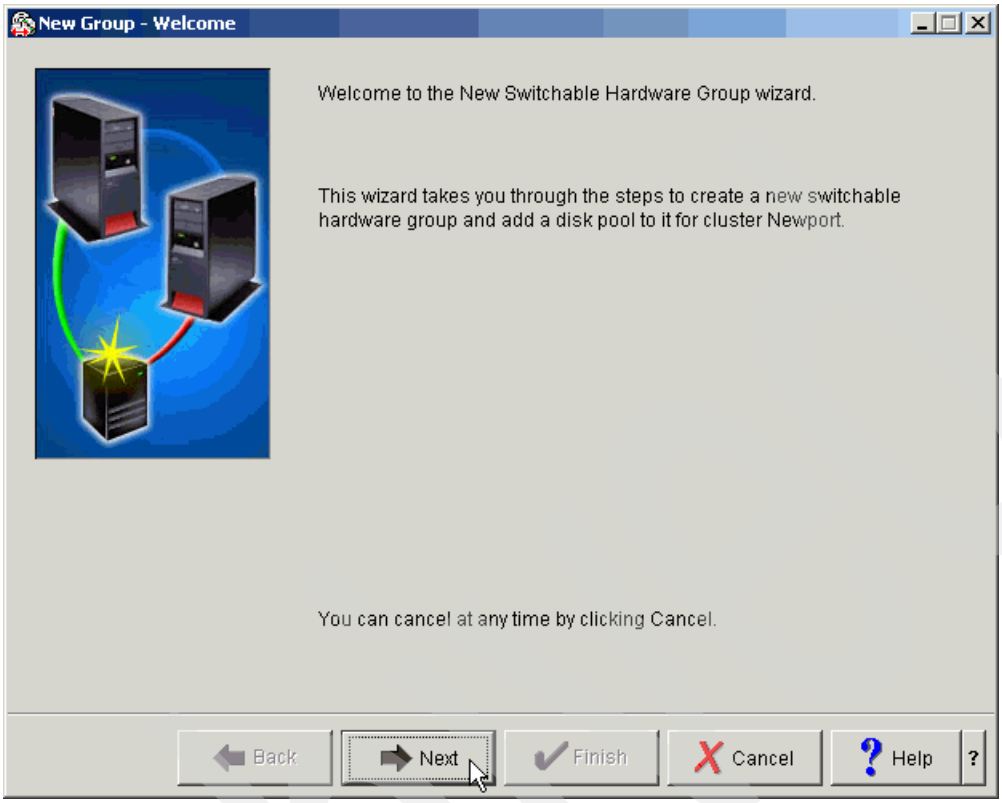


Figure 7-28 New Group - Welcome window

7. On the New Group - Specify Primary Node window (Figure 7-29), select the primary node of the cluster from the drop-down list. Click **Next**.

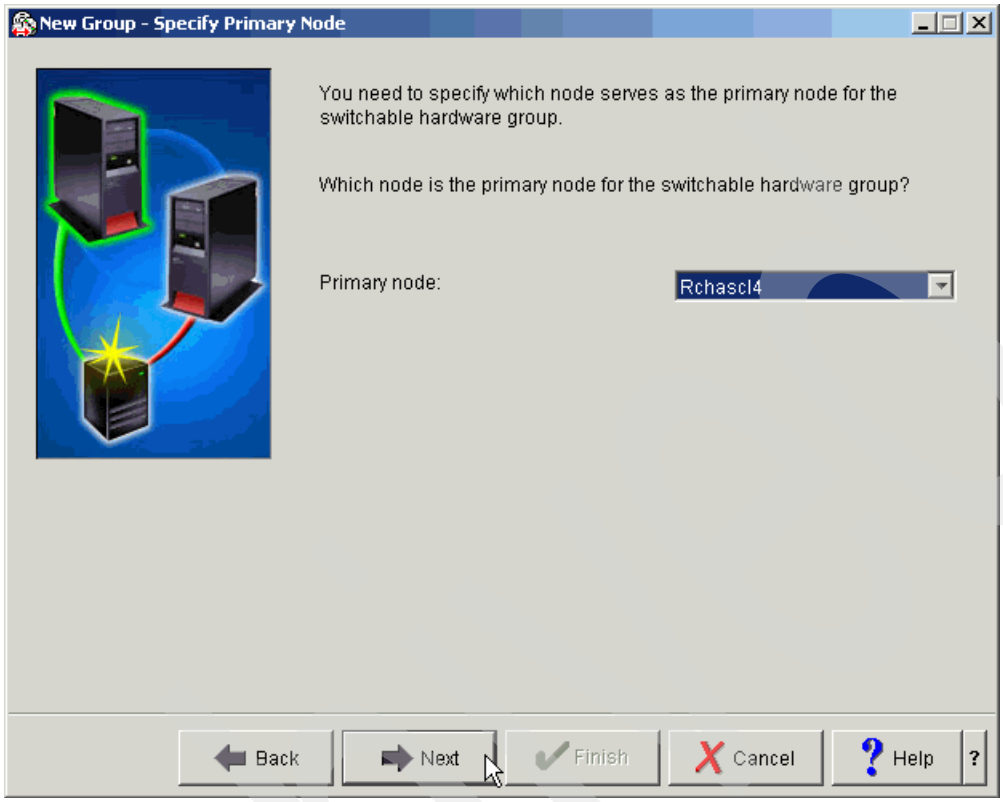


Figure 7-29 New Group - Specify Primary Node window

8. On the New Group - Specify Primary Name window (Figure 7-30), specify a name for the switchable hardware group. Supply a description if desired. Click **Next**.

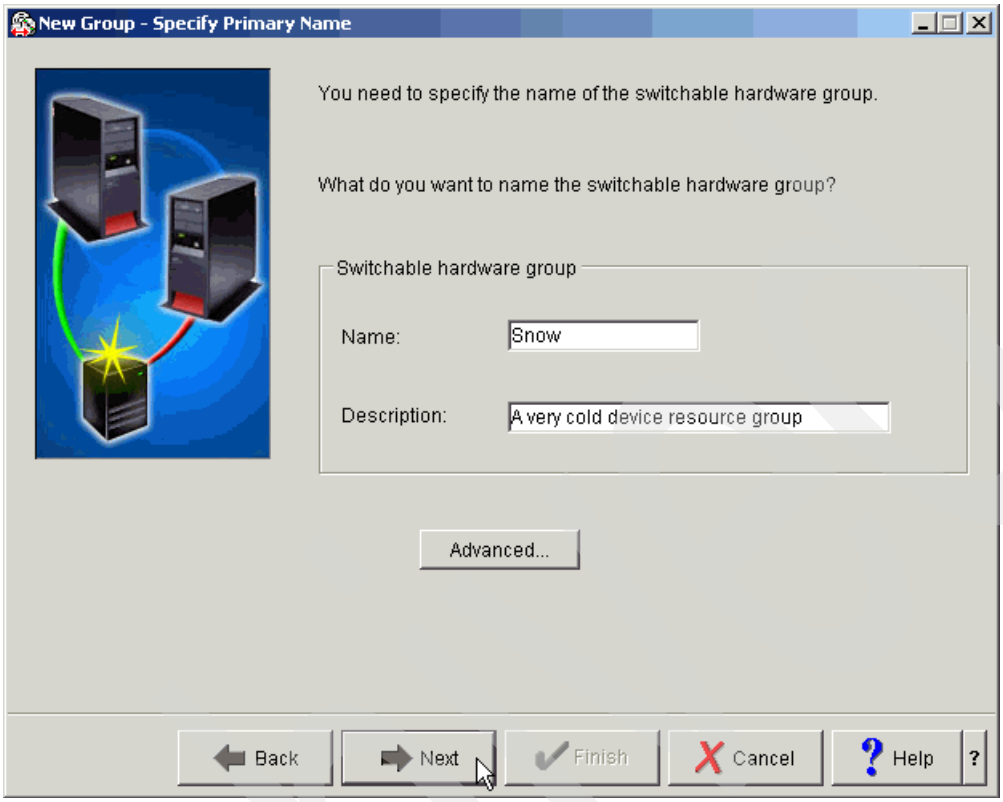


Figure 7-30 New Group - Specify Primary Name window

9. On the New Group - Create New or Add Existing Disk Pool window (Figure 7-31), you can either create a new disk pool (switchable) or add an existing disk pool (previously configured stand-alone). Because some of these panels differ from the Add New Disk Pool wizard when selected from the disk pools section, we step through the process. Select **Yes, create a new switchable disk pool**. Click **Next**.

Figure 7-31 New Group - Create New or Add Existing Disk Pool window

10. On the New Group - Disk Pool Type window (Figure 7-32), select the type of disk pool. Be sure to select **System-generated** from the drop-down list for Database name for Primary. Click **Next**.

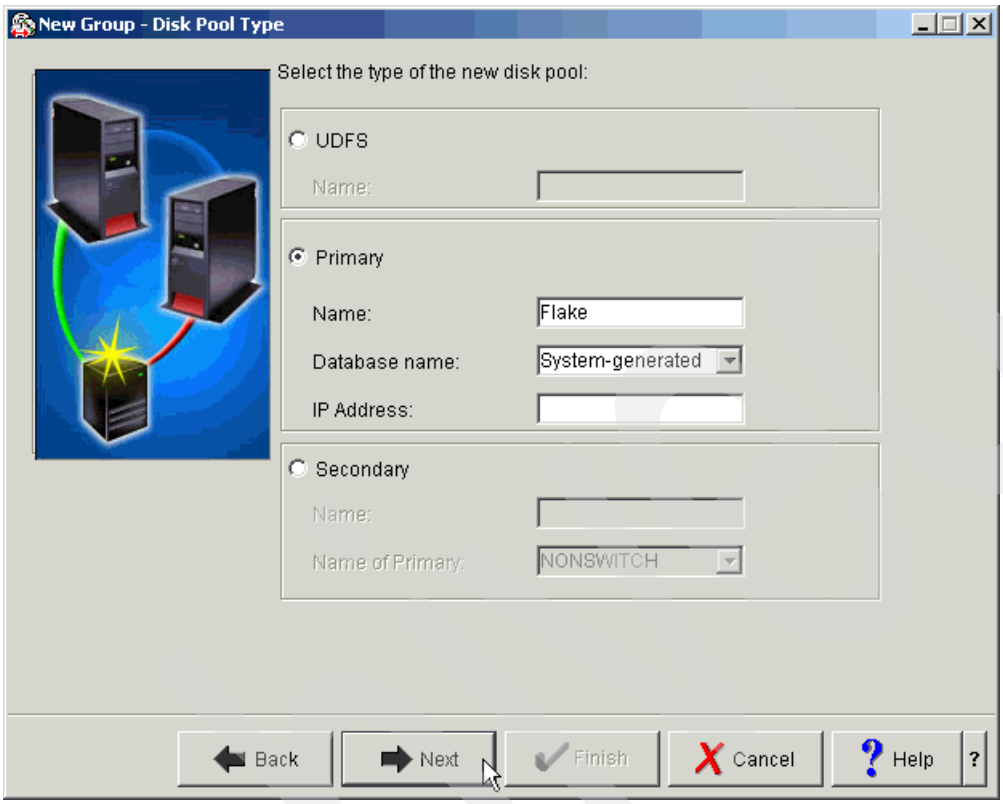


Figure 7-32 New Group - Disk Pool Type window

11. On the New Group - Protect Data in the New Disk Pool window (Figure 7-33), for mirrored or RAID-5 disk units to be added to the IASP, select **Yes, protect the data in the disk pool**. If no protection is desired on the non-configured disk units, select **No**. Click **Next**.

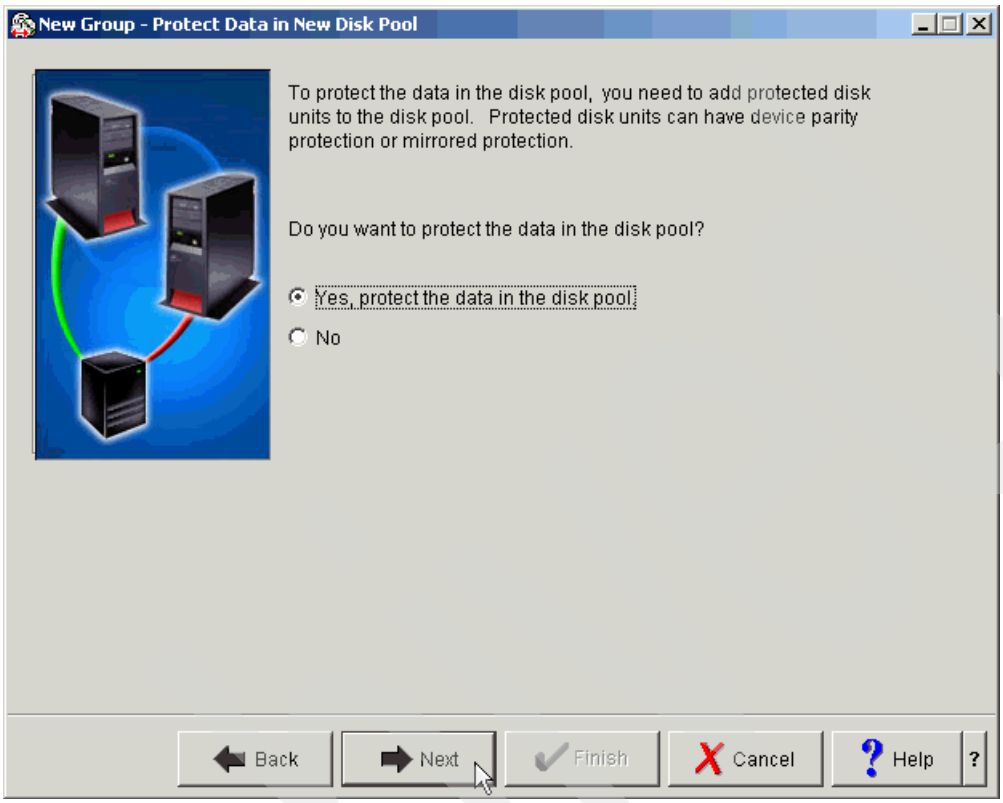


Figure 7-33 New Group - Protect Data in the New Disk Pool window

From this point forward, the windows are displayed from the Add Disk Unit wizard, as explained in 6.2, “Creating a primary disk pool as a stand-alone resource” on page 74. However, when you finish creating the new disk pool, the final window is displayed for the new cluster resource group creation (Figure 7-34).

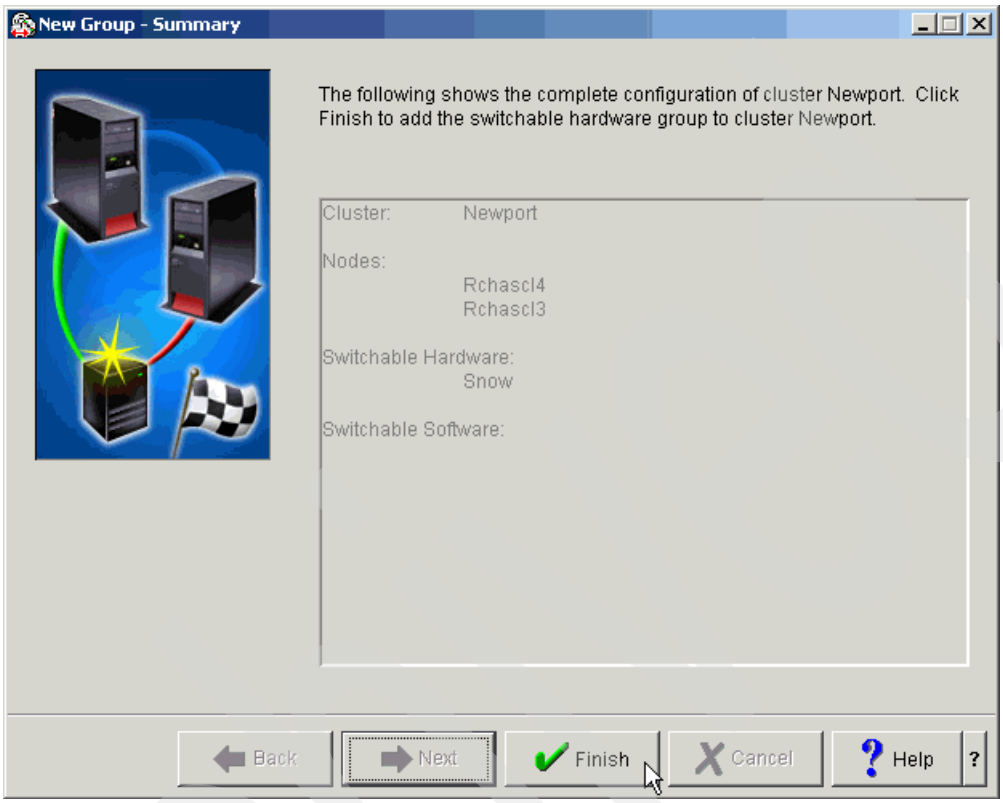


Figure 7-34 New Group - Summary window

7.4.3 Creating an IASP switchable between multiple systems and LPARs

The multiple system with LPAR configuration of the switchable IASP has the most hardware requirements of the three configuration scenarios. This configuration is implemented by the location of the IASP within an isolated tower, which switches in its entirety to the secondary system. These towers must be connected by HSL (high-speed link) and require 5722-SS1 Option 41 (HA Switchable Resources) to be installed.

See Chapter 3, “Configuration examples” on page 27, for a more detailed explanation of the requirements.

7.4.4 Making a tower switchable

The tower containing the disk units to be switched from one system to another must be defined as switchable prior to creating the switchable IASP. Here are the steps to accomplish this:

1. Open the iSeries Navigator and expand the towers initial owning system under **My Connections**.
2. Expand **Configuration and Service-> Hardware, Disk Units**.
3. When the Service Device Sign-on window opens, sign on with your DST/SST QSECOFR user ID and password. Remember, this password *is* case-sensitive.

4. Expand **By Location** and right-click the tower that contains the non-configured disks. Select the **Make Switchable** option. See Figure 7-35.

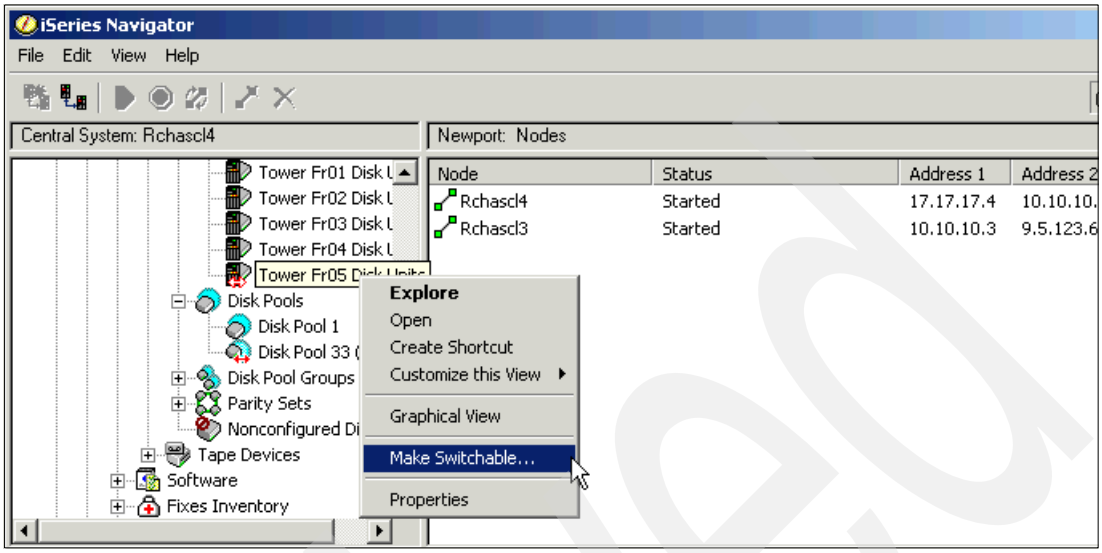


Figure 7-35 Making a tower switchable

5. On the Make Tower FR05 Switchable window (Figure 7-36), confirm your choice by clicking **OK**. When the action has completed, a confirmation panel opens.

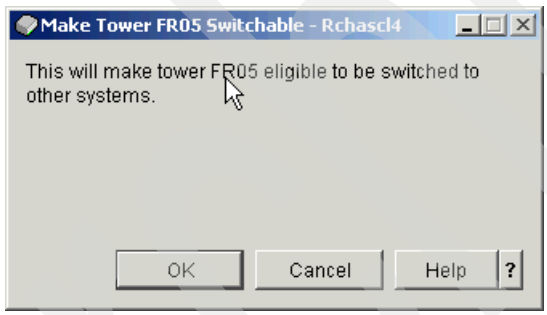


Figure 7-36 Confirming your selection to make a tower switchable

6. The remaining configuration, creating the cluster resource group and disk pool, is detailed under 7.4.2, “Creating a switchable hardware group” on page 97.

8

Installing applications

This chapter explains some considerations for installing applications into an independent disk pool. It considers both the non-switchable and switchable disk pool scenarios.

In general, the required libraries and objects for the application may not be available in the system auxiliary storage pool (ASP) and are known to exist in an independent disk pool. Making that disk pool available and executing the Set Auxiliary Storage Pool Group (SETASPGRP) command makes those libraries and objects available to the current job or thread.

To explain better how the current installation processes may need to be changed to facilitate independent ASPs (IASPs), Phoenix Software has allowed us to use their system as an example. The Spectrum Distribution System from Phoenix Software of Tustin, CA, is a full distribution and accounting system, with job costing capabilities.

To learn more about Phoenix Software, you can find them on the Web at:

http://www.phoenixsoftware.com/

8.1 Phoenix system discussion

A customer decided that implementing the Phoenix system required a two-system, disk pool switching solution. The order entry portion must be available 24-x-7 to allow for one hour of downtime to switch over.

There are several issues to consider here. However, when it comes to making the data accessible, the use of disk pools became the clear answer. Some items at the system level require either replication or restore at the time of the switch, but the placement of the libraries is what we discuss here.

The Phoenix system normally consists of three libraries: system, program, and file libraries. When the entire system is installed, components may be moved into various IBM-provided libraries to improve access. However, most of the provided applications stay in the provided libraries. The initial inclination is to place all three libraries in the IASP. After going through the various object types and seeing how the system works, another installation method was found.

Review of the application provided three basic categories of objects:

- **Category 1**: The object is required to be in the system ASP.

  The objects are not supported by OS/400 in an IASP. In the case of the Phoenix system, these are the JOBQ object types.

- **Category 2**: The application requires objects be at the system ASP level.

  The application usage dictates that the objects are in a centralized common area. This is the system ASP. In the case of the Phoenix system, these were the JOBDs and the programs and files used by the menu system. Because the JOBD allows for designation of an ASPGRP as a parameter, even though they are allowed in an IASP by OS/400, they should be placed in the system ASP.

- **Category 3**: The object can be in any ASP.

  The objects are not in the other two categories. In the case of the Phoenix system, they are the bulk of the application programs and data files or objects.

The objects in categories 1 and 2 must be installed on both sides of the cluster. They must also be maintained at the same level. Hopefully these objects change slowly, and a simple save and restore process is adopted to maintain currency. Rapidly changing objects remaining in *SYSBAS requires a replication technique for maintenance.

Note: If an application update is applied to the primary side and the underlying IASP based database changes, you *cannot* switch to the back system until the application objects on the backup match those on the primary.

8.2 Current Phoenix Software installation instructions

Phoenix Software requires only a small set of installation instructions as explained here:

1. Create the user profile to load everything. Enter the following command:

   ```
   CRTUSRPRF USRPRF(SPECTRUM) INLPGM(*LIBL/ZZMENUCL) INLMNU(*SIGNOFF) TEXT('Spectrum Master
   Profile') JOBD(DMOSYS240/SPECTRUM)
   ```

2. Restore the appropriate libraries by entering the following commands:

   ```
   RSTLIB SAVLIB(DMOFIL240) DEV(TAP01) ENDOPT(*LEAVE)
   RSTLIB SAVLIB(DMOSYS240) DEV(TAP01) ENDOPT(*LEAVE)
   ```

```
RSTLIB SAVLIB(SDSPGM240) DEV(TAP01) ENDOPT(*LEAVE)
RSTLIB SAVLIB(SDSSYS240) DEV(TAP01) ENDOPT(*LEAVE)
```

3. Change or add specific routing entries to the batch subsystem using the following command:

```
ADDRTGE SBSD(QSYS/QBATCH) SEQNBR(600) CMPVAL('PGMEVOKE') PGM(*RTGDTA) CLS(QSYS/QBATCH)
```

4. Add specific job queue entries to the batch subsystem description. Enter the following commands:

```
ADDJOBQE SBSD(QSYS/QBATCH) JOBQ(SDSSYS240/SPECTRUM_1) SEQNBR(20)
ADDJOBQE SBSD(QSYS/QBATCH) JOBQ(SDSSYS240/SPECTRUM_2) SEQNBR(30)
ADDJOBQE SBSD(QSYS/QBATCH) JOBQ(SDSSYS240/SPECTRUM_3) SEQNBR(40)
```

That's it. The Phoenix system is installed. Logging in as user SPECTRUM lets you select various menu options that complete the setup of the Phoenix system.

8.3 Changing the installation to a non-switchable disk pool

It was determined that Phoenix is to be installed to an IASP with a control library in the system ASP rather than installing it entirely to the system ASP. This section explains how to do this.

8.3.1 Phoenix system library installation

The system library is normally restored to the system ASP, so there are no changes. Therefore, the Restore System Library (RSTLIB) command, as shown here, remains unchanged. It identifies the library that we do not restore.

```
RSTLIB SAVLIB(DMOSYS240) DEV(TAP01) ENDOPT(*LEAVE)
```

8.3.2 Creating the disk pool

The name of the IASP is *Phoenix*. The database name is *Phoenix1*. You can find the steps for creating an IASP in Appendix A, "Prerequisite steps" on page 175, and in Chapter 7, "Switchable setup" on page 81.

8.3.3 User profile/job description considerations

If an IASP is primary or secondary, it is assigned an ASP group when it is created. The name of the ASP group is always the name of the primary IASP. The name of the IASP is usually the same as the database name that is assigned.

If an IASP is a user-defined file system (UDFS), it is not assigned an ASP group when it is created. Another way to understand this is that if libraries must exist in the IASP, the IASP must be created as a primary or secondary or converted to a primary or secondary from a UDFS.

8.3.4 Subsystem considerations

To become active, a subsystem description must reside in either the system ASP or a user ASP. Subsystem descriptions can exist in an IASP. However, to go active, they must reside in the system ASP.

Phoenix Software uses its own subsystem description named SPECTRUM. This must be placed in the system ASP to implement Phoenix Software. An archival version can be loaded into the IASP.

8.3.5 Job queue considerations

For the Phoenix system, three job queues are provided in what is considered the program library. These must remain in the system ASP.

8.3.6 Output queue and printing considerations

The *OUTQ object type is not supported in an IASP under V5R2. In a simple configuration where the IASP is always available for use by an application, this is of little concern. However, when the IASP is switched to another system, any spooled files in an output queue on the parent system that pertain to the data in the IASP are not switched to the second system along the IASP. You must plan to save and restore the spooled files, replicate them, rerun them, or ignore them, as the business requires.

In addition, if you choose to store external resources for spooled files in an IASP, you must be aware of the possible issues. Such objects as *FNTSRC, *FORMDFN, *OVL, *PAGDFN, and *PAGSEG can be stored in an IASP. Using the SETASPGRP command to bring the resources in the IASP into the name space allows the jobs to print correctly.

For the Phoenix system, no output queues are provided. Therefore, you do not need to make any changes to the system. The Phoenix system uses IBM-provided output queues, which all exist in the system ASP on both sides of the cluster.

8.3.7 Switchable disk pool

The user profile refers to a particular job description. The job description in V5R2 supports the Initial ASP Group (INLASPGRP) keyword.

The Initial ASP Group keyword specifies the initial setting for the ASP group name for the initial thread of jobs using this job description. A thread can use the SETASPGRP command to change its library name space. When an ASP group is associated with a thread, all libraries in the independent ASPs in the ASP group are accessible. Objects in those libraries can be referenced using regular library-qualified object name syntax. The libraries in the independent ASPs in the specified ASP group plus the libraries in the system ASP (ASP number 1) and basic user ASPs (ASP numbers 2 to 32) form the library name space for the thread.

There are two possible parameters for the keyword:

- ***NONE**: This value specifies that the initial thread of jobs using this job description is started without an ASP group. The library name space does not include libraries from any ASP group. Only the libraries in the system ASP and any basic user ASPs are in the library name space.
- **auxiliary-storage-pool-group-name**: This value specifies the name of the ASP group to be set for the initial thread of jobs using this job description. The ASP group name is the name of the primary ASP device within the ASP group. All libraries from all ASPs in this ASP group are included in the library name space.

For the Phoenix system, the user profile *Spectrum* is provided, which uses a job description of *Spectrum*. This job description is provided in a program library. The job description is where the ASP group is set for the libraries to be found. Because of this, leave the job description in the system ASP.

8.4 Moving applications from *SYSBAS to an independent disk pool

We recommend that you delete all objects that the application creates (especially the unqualified objects that may go in surprising places) before you run an application in *SYSBAS and later in an independent disk pool (or visa versa). Failure to do this may cause the independent disk pool to be ineligible to be varied on.

For example, create a collection while running in independent disk pool X1. Then make X1 unavailable and run the same coded in *SYSBAS. The collection is created in *SYSBAS. Because duplicate collection names are not allowed in the same name space, independent disk pool X1 cannot be made available.

However, the system is somewhat forgiving. Error messages are provided that point to the offending object or objects. You can remove or rename them as a means of repairing the problem. Keep in mind that this can be a time consuming process. We recommend that you perform thorough testing when moving an application from one disk pool to another.

9

An implementation example

This chapter provides several implementation and usage examples of independent disk pools. They range from the ultra simple to the extremely complex. They demonstrate the flexibility and possibilities of independent disk pools in the day-to-day iSeries world.

The samples covered in this chapter are:

- Non-switchable simple independent disk pool
- Switchable independent disk pool
- Multiple versions of software with independent disk pools
- Integrated xSeries servers and independent disk pools
- Partitions with independent disk pools
- Linux with independent disk pools
- Remote and local journaling with independent disk pools

9.1 Non-switchable simple independent disk pool

To get used to the idea of independent disk pools, we created a simple example of a single system with a single non-switched independent disk pool. We installed the application on the independent disk pool to show the types of changes that are required for an application to move to an independent disk pool.

We used Phoenix Software's Spectrum Distribution System to show a simple independent disk pool. The environment is a single iSeries server with a single independent disk pool configured. The Spectrum software is loaded on the independent disk pool. Figure 9-1 shows the physical setup.

Figure 9-1 Spectrum Distribution System environment: Non-switchable independent disk pool

Spectrum Distribution System is a simple ERP application that was created before the concept of independent disk pools was designed. Spectrum is distributed as a set of four libraries that contain all the necessary objects. The original installation instructions for the Spectrum Distribution System are:

1. Create the Spectrum user profile under which all the objects load. Enter the command:

   ```
   CRTUSRPRF USRPRF(SPECTRUM) INLPGM(*LIBL/ZZMENUCL) INLMNU(*SIGNOFF) TEXT('Spectrum Master Profile') JOBD(DMOSYS240/SPECTRUM)
   ```

2. Restore the product libraries from the install tape. The four libraries are SDSSYS240, DMOSYS240, SDSPGM240, and DMOFIL240. Enter the following command for each library:

   ```
   RSTLIB SAVLIB(library) DEV(TAPxx)
   ```

3. Change or add program evoke routing entry to the QBATCH subsystem. By default, this routing entry is already added to the QBASE subsystem. Enter the following command:

   ```
   ADDRTGE SBSD(QBATCH) SEQNBR(600) CMPVAL(PGMEVOKE 29) PGM(*RTGDTA)
   ```

4. Add specific job queue entries to the batch subsystem description. The job queues are SPECTRUM_1, SPECTRUM_2, and SPECTRUM_3 in library SDSSYS240. Enter the following command for each job queue:

   ```
   ADDJOBQE SBSD(QBATCH) JOBQ(SDSSYS240/SPECTRUM_1)
   ```

5. Sign on as SPECTRUM and select the appropriate menu option to complete the setup.

9.1.1 Installing Spectrum on an independent disk pool

To install the Spectrum Distribution software on an independent disk pool, create an independent disk pool and then change the installation to use the created disk pool. You can create the independent disk pool using iSeries Navigator.

Installation changes to Spectrum

To install the Spectrum software on an independent disk pool, you must make several changes. First, *JOBQ, *JOBD, and *OUTQ objects are installed as part of the Spectrum software. These object types are either not supported or cannot be referenced in independent ASPs. For these objects, you must create a new library, which must be in the system ASP. The remaining objects in the original libraries can be placed in the independent disk pool. Then, you must change the job descriptions to specify an initial ASP group. This ASP group is the ASP group where the Spectrum software is loaded.

The new installation instructions are outlined here:

1. Create the Spectrum user profile under which all the objects load. Enter the following command:

   ```
   CRTUSRPRF USRPRF(SPECTRUM) INLPGM(*LIBL/ZZMENUCL) INLMNU(*SIGNOFF) TEXT('Spectrum Master Profile') JOBD(DMOSYS240/SPECTRUM)
   ```

2. Restore the libraries with the unsupported object types. The libraries are DMOSYS240 and SDSSYS240. Enter the command:

   ```
   RSTLIB SAVLIB(library) DEV(TAPxx)
   ```

3. Restore the product libraries from the installation tape to the IASP. The remaining libraries are SDSPGM240 and DMOFIL240. Enter the following command for each library:

   ```
   RSTLIB SAVLIB(library) DEV(TAPxx) RSTASPDEV(IASPdevice)
   ```

4. Change or add program evoke routing entry to the QBATCH subsystem. By default, this routing entry is already added to the QBASE subsystem. Enter the command:

   ```
   ADDRTGE SBSD(QBATCH) SEQNBR(600) CMPVAL(PGMEVOKE 29) PGM(*RTGDTA)
   ```

5. Add specific job queue entries to the batch subsystem description. The job queues are SPECTRUM_1, SPECTRUM_2, and SPECTRUM_3 in library SDSSYS240. Enter the following command for each job queue:

   ```
   ADDJOBQE SBSD(QBATCH) JOBQ(SDSSYS240/SPECTRUM_1)
   ```

6. Change all the Spectrum job descriptions to specify the initial ASP group. The job descriptions are 1, 2, and 3. Enter the following command for each job description:

   ```
   CHGJOBD JOBD(IASPSYS240/jobd) INLASPGRP(IASPdevice)
   ```

7. Sign on as SPECTRUM and select the appropriate menu option to complete the setup.

The Spectrum software is now installed on an independent disk pool. All the programs and data are contained in one independent disk pool with the exception of those objects that are not supported in an independent disk pool. When you sign on as SPECTRUM, the SPECTRUM job description automatically adds the independent disk pool that contains all the objects for Spectrum software. To the user, there is no change in how the program looks or behaves.

9.1.2 Single non-switched independent disk pool comments

A reason to install a product in a single non-switched independent disk pool is to isolate the disk resources required for that application from the rest of the system, or to consolidate multiple servers on one system.

The example with Spectrum software installed on a non-switched independent disk pool shows the strength of independent disk pools. From an administrator's perspective, you can install software on an independent disk pool without changing how users use the software. Independent disk pools can be a easy and effective way to isolate users from certain software. Without the independent ASP listed in the job, a user cannot see or run the software loaded on the independent disk pool. If you are in a situation where you need to prove that only authorized users can run software, you may want to use independent disk pools.

9.2 Switchable independent disk pool

The next step takes a single independent disk pool and makes it switchable between two systems. By switching a disk pool, you can improve the overall availability for your application.

To show you how an independent disk can be switched from system to system, we used the IBM Content Manager OnDemand for iSeries product. The system environment is a single independent disk pool that is switched between two systems. Figure 9-2 shows the physical setup.

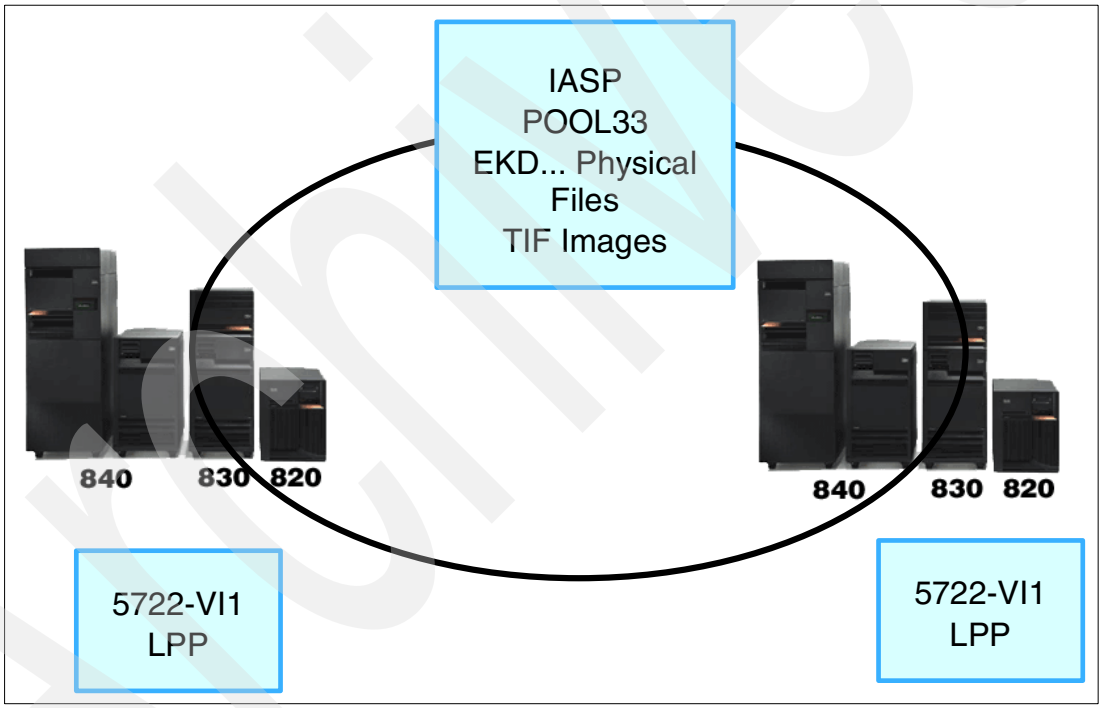


Figure 9-2 Content Manager OnDemand for iSeries environment: Switchable independent disk pool

Content Manager OnDemand for iSeries is a licensed program product that catalogs and stores image files. In our environment, we installed the licensed program on both systems with the data in an independent disk pool that can be switched between the two systems. The data is stored in integrated file system (IFS) files, and all the files are on disk in a switchable tower.

9.2.1 Installing Content Manager OnDemand for iSeries on an independent disk pool

With a shared or switched disk pool, more configuration steps are required to create the independent disk pool. You must cluster the two systems and create a device domain that is shared between the two systems. You also need a switchable hardware group object and an ASP device description. Figure 9-3 shows all the required objects and their relationship with each other.

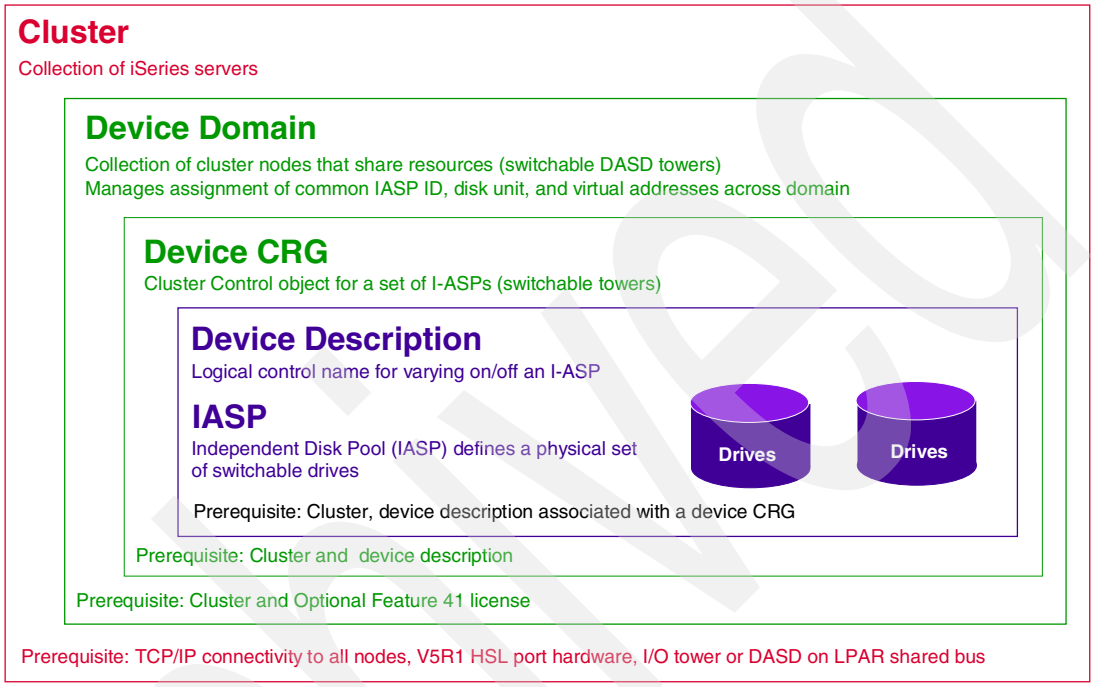


Figure 9-3 Switchable independent disk pool object relationship

Creating the objects

To create a switchable independent disk pool, you must first create a cluster. A cluster is the collection of iSeries servers that will share the independent disk pool. You follow these steps:

1. Create the cluster using iSeries Navigator and the clustering functions under Management Central.
2. Create a device domain. The device domain helps to manage the actual resources that are contained in the cluster and are switchable.
3. Create a switchable hardware group for the specific independent disk pool.
4. Create the independent disk pool with the disk units that are to be switched.

Adapting Content Manager OnDemand for iSeries

The actual changes to Content Manager OnDemand for iSeries are limited to moving the IFS directory that contains the image files from the system ASP to the independent disk pool. On the backup system, you remove the IFS directory for the image files. When you switch the independent disk pool to the backup system, the IFS directory is then available.

9.2.2 Switching the independent disk pool

To actually switch the disk pool, use iSeries Navigator and the clustering functions under Management Central.

9.2.3 Switchable independent disk pool comments

While a switchable independent disk pool does not replace full-blown cluster support with automatic failover, it provides a higher level of availability. Using a switchable independent disk pool in this environment can be a great way to implement planned outages, especially if the amount of data that you need to replicate or copy to the backup system is large. In this case, the data is never copied to the backup system. The disks that contain the data are simply moved to the backup system.

This does *not* work for performing system saves, because the data that is moved to the backup system is not available on the primary system for saving. If you have several applications running in production mode with data on independent disk pools, you may be able to use a switched environment to have only one application down at a time for a save.

In the Content Manager OnDemand for iSeries example, only the data is contained in the independent disk pool. Each system still needs a copy of the product code. There may be cases where separating the data from the code is not possible. In these cases, you can place both the data and code in the independent disk pool. You need to check with your software service provider to determine whether a new license is necessary for the backup system.

9.3 Independent disk pools with multiple versions of software

Supporting multiple versions of a product is another use of independent disk pools. For this, the programs and data for each version of the product are installed on separate independent disk pools. A user signing on to the disk pool that is set determines which version of the product is used. The disk pool can be assigned based on a job description or the initial program running the SETASPGRP command.

Figure 9-4 shows the environment used for a fictitious software product. In this environment, version x of the product is installed on independent disk pool POOL145. This contains all the programs and data required for the application. The next version of the product is on independent disk pool POOL146. All programs and data for this version are located on this independent disk pool. For a user to use version x of the product, either the job description for that user or the initial program for that user sets the ASP group to POOL145. For another user to use version x+1 of the product, the ASP group is set to POOL146.

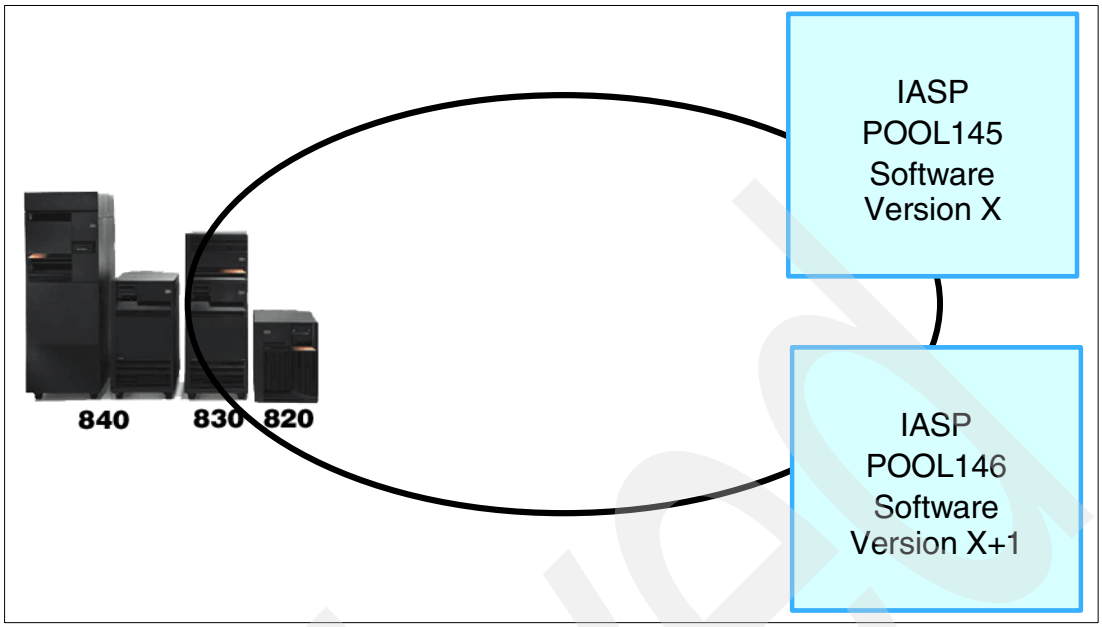


Figure 9-4 Multiple Spectrum versions using multiple independent disk pools for multiple versions

9.3.1 Usage for multiple versions

There are many tasks you can perform with the iSeries now that you can have multiple copies of the same library. Prior to V5R2, to test a new version of a product, you had to have a dedicated system to install the new release and analyze it there, or you needed to have a partition on which to install the new release. These are good approaches and handle all situations for all products.

However, we recommend that you install the product, programs, and data, on independent disk pools to solve this problem. When a newer version of the product comes out, you can install that version on a new independent disk pool and test it there. You can even migrate some users to this new version, while allowing other users to remain on the old release.

You can use independent disk pools to handle two different applications that, otherwise, can interfere with each other at a program or data level. For example, you have two applications that need to touch the same file object. While one application is working on the file, it is locked and the other applications can no longer perform any work. It may be possible to move that file onto two independent disk pools and then structure the applications to use different disk pools. This isolates the file used from the other application. This obviously cannot work if both applications work on the exact same data within the file.

Another possibility for configuring multiple independent disk pools on a single system is for server consolidation when one organization is not allowed to view data of another organization. In this environment, you can install a single version of the product in the system ASP. The independent disk pools are configured to contain the data library for each organization.

When a user from an organization signs on, the ASP group containing the data for their organization is set in their job. This allows a user to view the data they need without any concerns of them accessing data for another organization. The only concern with this environment is how an organization with a headquarters would summarize data from each organization. To solve this concern, summary data from each organization can be copied to a system ASP library. Or a headquarters program with authority to each independent disk pool can loop through each independent disk pool and gather the required summary data.

9.3.2 Multiple independent disk pools for multiple version comments

Using multiple independent disk pools to run multiple version of a product allows you to consolidate multiple system into one. And you don't have to worry about getting all applications to the same version and keeping all applications at the same version.

If you plan to use this approach, there are several considerations that you need to be aware of. For example, serviceability of the multiple versions is more complicated. If you use multiple versions for testing new releases, serviceability issues may be negated since there is little simultaneous service to the two versions.

Note: An important consideration is whether it is possible. If non-supported objects are different in the two releases, then you may not be able to support multiple versions. We experienced this when we looked at doing this with the Spectrum Distribution System. The system objects library needed to be in the system ASP because of unsupported objects. With only one version of this library, we could not separate the two versions.

Another scenario is where you cannot support multiple versions is when you use IFS as the runtime environment for program objects. This is normally done via symbolic links in IFS to the QSYS file system program objects. If the product does not support multiple runtime directories, then it is not possible for the symbolic links to point to two different independent disk pools. We experienced this when we originally tried this example with Domino for iSeries.

9.4 Integrated xSeries Servers and independent disk pools

The Integrated xSeries Servers can also benefit from independent disk pools. Since the storage spaces for Integrated xSeries Server are kept in IFS, you can keep that storage in an independent disk pool. When the primary system is down, the independent disk pool is switched to a backup system, and the storage spaces linked to the Integrated xSeries Server objects on the backup system.

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 disks are:

1. Take the iSeries Server A offline. The disk tower switches to iSeries Server B.
2. Manually link the network server descriptions (NWSDs) to iSeries Server B resource names.
3. Reboot the Windows servers. The Windows servers are back online on iSeries Server B.

See Figure 9-5 for an illustration of this example.

Note: The xSeries servers must have the same configuration.

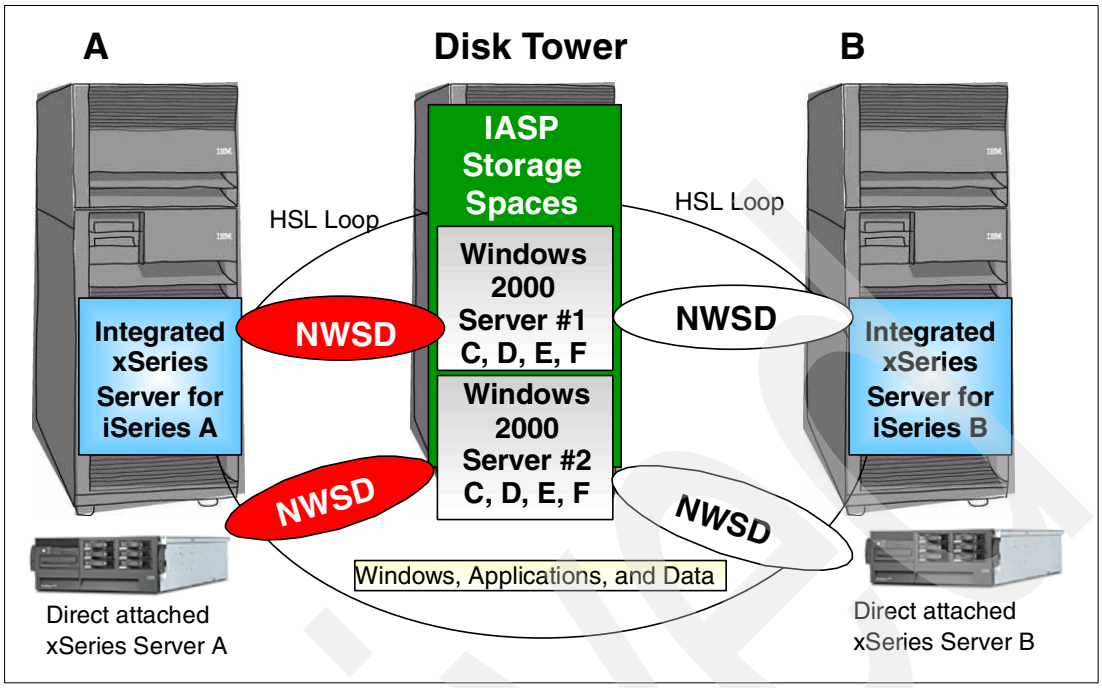


Figure 9-5 Integrated xSeries Server or direct attached xSeries Server using IASP

9.4.1 Real example with Integrated xSeries Adapters

Figure 9-6 shows a real-world example of using Integrated xSeries Adapters and independent disk pools.

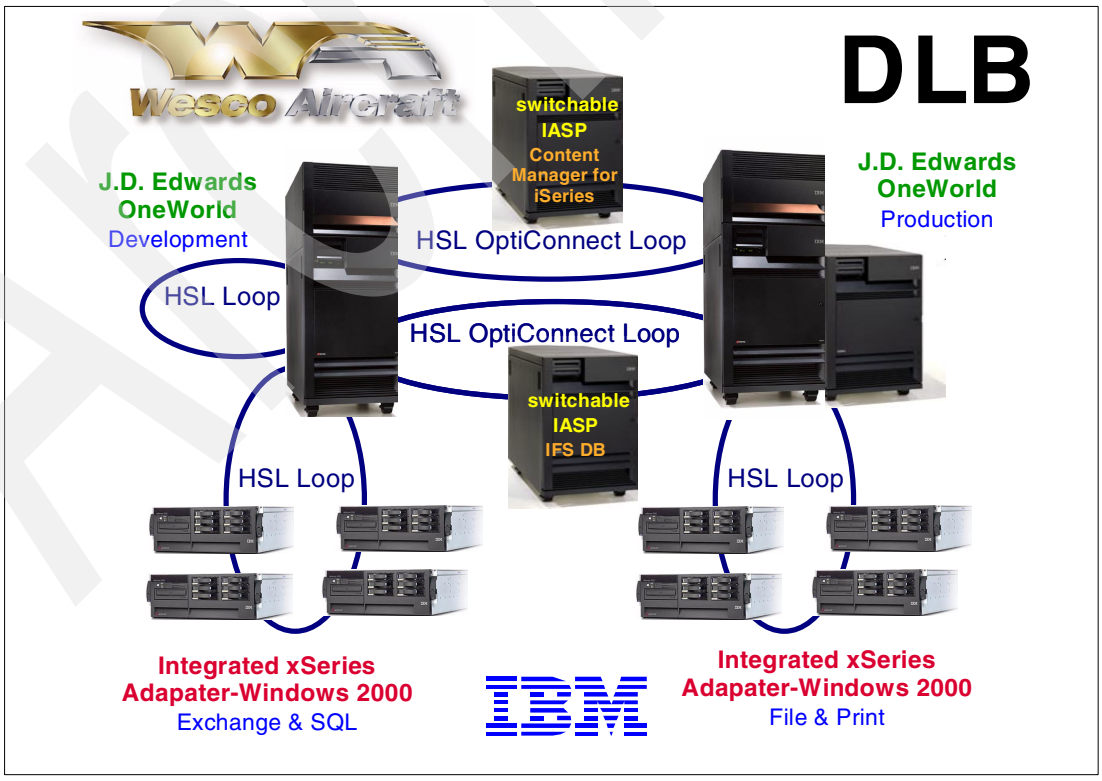


Figure 9-6 Wesco Aircraft example

In this environment, the disks containing the server storage for the Integrated xSeries Adapters and the Integrated xSeries Adapters themselves are in switchable towers and switchable independent disk pools. In the event of a failure or during routine maintenance on the primary system, the independent disk pool and the Integrated xSeries Adapters can be switched to the backup system. For planned outages, it allows the applications running on the Integrated xSeries Adapters to continue to be available. For unplanned outages, you have determinable time for recovery.

9.5 Partitions and independent disk pools

Now that we covered the basics, we can explore more complicated examples involving logical partitions, multiple towers, and multiple disk pools all in the same environment. Figure 9-7 shows a simplified diagram of our environment.

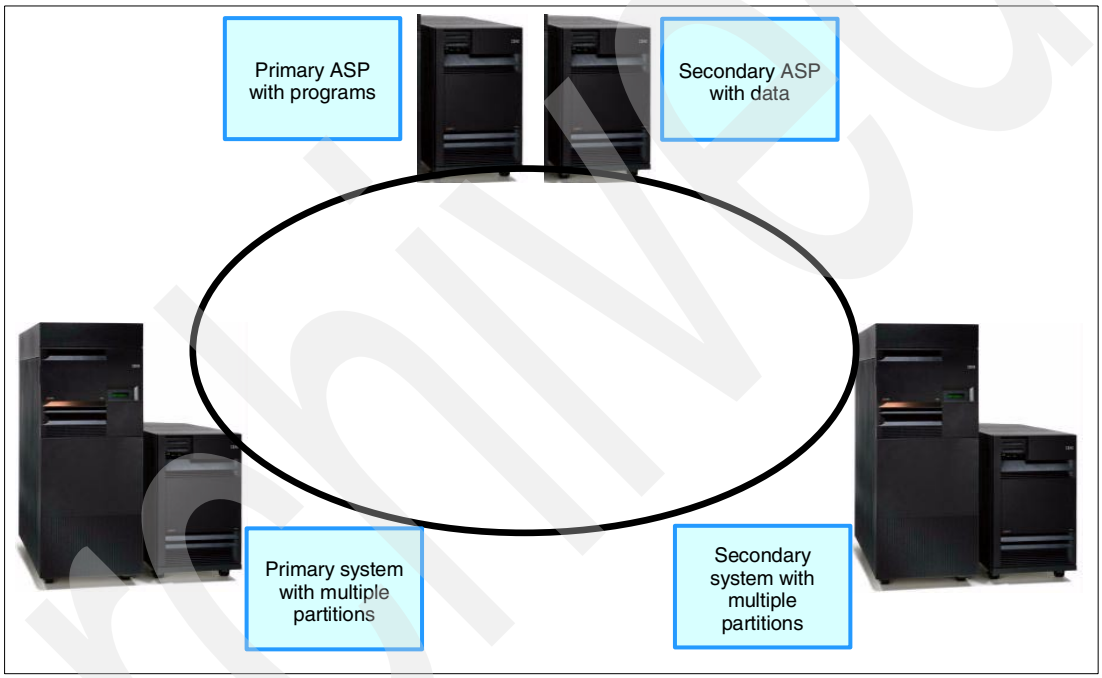


Figure 9-7 Partitions and disk pools: Switchable independent disk pools with partitions

9.5.1 Real example setup

Figure 9-7 shows a real example that is planned to be implemented with the release of V5R2. On the primary system, one partition has basic system availability (disk protection) and recovery (backup). The rest of the partitions already have high availability options setup. The goal is to use independent disk pools to provide a higher availability. The decision to use independent disk pools for higher availability is based on the fact that the applications running on that partition are planned to be moved or removed in a couple of years. The hardware costs involved with the other high availability options were not justified.

The first step in the customer shop is to walk through each application running on the partition to determine what can and cannot go to an independent disk pool. Next the disks required to support the partition are gathered into the required number of towers and cabled to the primary and backup systems. The decision was made to have the program objects in the primary disk pool and the data objects in the secondary disk pool. This allows a disk failure in one of the pools without impacting the other.

After the objects are identified and the disk pools created, the objects can be moved. The final step is to handle those objects that need to remain in the system disk pool. These objects must be replicated to the backup system. If the objects do not change frequently, it may be possible to use backup tapes to get the objects there. Otherwise, a data replication strategy is necessary.

On the backup system, a partition is created with the disk resources needed only for the system disk pool. The processor and memory resources are set to support the method used for getting the system pool data. When a failure occurs on the primary system or when a planned outage occurs, the partition on the backup system is increased to support the needs of the applications, and the independent disk group is switched.

9.5.2 Other uses of partitions and independent disk pools

By using a backup partition on another system, you can use the resources for other activities until the backup capability is needed. This can be a great way to provide for planned outages. Also the amount of time to switch the independent disk pools is less than taking down a system, performing the hardware or software maintenance, and bringing the system back up.

In the case of a hard failure, the amount of time to rebuild all the access paths on the independent disk pool is the same whether the rebuild takes place on the backup or primary system. Using independent disk pools does not give you 100% availability with automatic failover. However, it can give you a determinable recovery time after a failure.

9.6 Linux and independent disk pools

Linux on iSeries was introduced in V5R1. With the introduction of switchable independent disk pools, you now can have a higher available Linux system.

To use Linux with independent disk pools, you need to use hosted or virtual direct access storage device (DASD). The disk space for the Linux partition is created using network server storage spaces.

A selling point for placing Linux on an iSeries partition is the ease of backing up the network server storage spaces and the ability to duplicate a Linux installation by copying the storage space. If that server space is on an independent disk pool, then you can switch that disk pool to another system and bring up that Linux system on another partition. Figure 9-8 shows a possible configuration with Linux and independent disk pools.

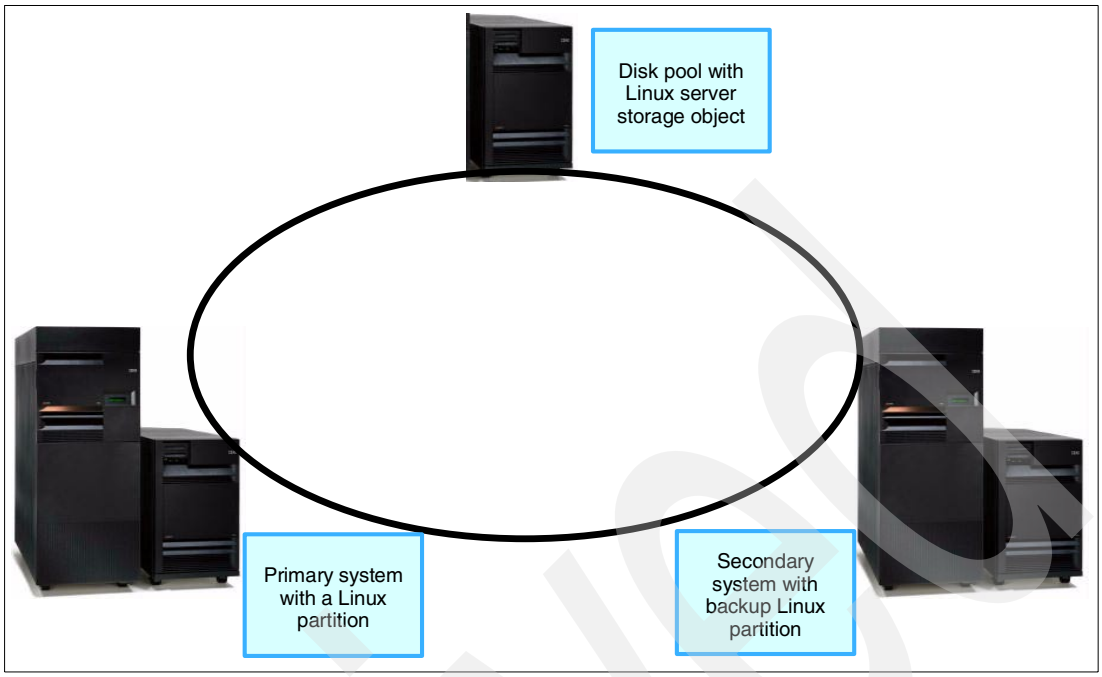


Figure 9-8 Linux and disk pools: Switchable independent disk pools

9.6.1 Linux and independent disk pool comments

If your Linux disk requirements are large, then it is not practical to save the storage spaces and move them to a backup system. In this case, a switchable independent disk pool provides a convenient and effective way to move a Linux system from one system to another.

If the network storage space is the only object on the switchable disk pool, in the event of a hard failure, there should be little recovery time when switching the disk pool to the backup system and bringing it up.

9.7 Remote and local journaling with independent disk pools

There are numerous considerations when trying to determine how, where, and when journaling should take place. When OS/400 and journaling were first introduced, the only question with journaling was when to use it and when not to use it.

With the introduction of user ASPs, another option was introduced. With user ASPs, the journal receiver (*JRNRCV object type) can be placed on a different ASP than the file being journalled. This offers two advantages. First, the disk I/O for the journal receiver is isolated from the file input/output (I/O). This gave better performance since the file I/O and journal I/O can take place at the same time.

The second advantage is the isolation of failure. If a disk failure occurs in one user ASP, it does not necessarily impact the objects in other ASPs. With remote journaling, you can isolate the failure and performance issues even farther away. Now with independent disk pools, you have yet another option.

With independent disk pools, it is possible to isolate journaling issues slightly more than user ASPs without the full hardware requirement of another system for remote journaling. If you use remote journaling, it may be useful to place that journal on a switchable disk pool. When you need to take down the remote system for routine hardware or software maintenance, you can temporarily switch that independent disk pool to either a different remote system or to the local system and then change the journaling to point to the new location.

Figure 9-9, Figure 9-10, and Figure 9-11 show the progression of journaling possibilities.

With local journaling (Figure 9-9), the journal is on the local system in either the system ASP or a user ASP. The advantage of local journaling is the low cost and simplified recovery. The disadvantage is that a disaster can wipe out both the file and the journal. By using a user ASP, you can help isolate the journal from the file for performance reasons and for failures.

Figure 9-9 Local journaling

With remote journaling (Figure 9-10), the journal is physically kept on a different system. The advantage is complete isolation of the journal from the file in a disaster situation. The disadvantage is the cost involved with the additional system.

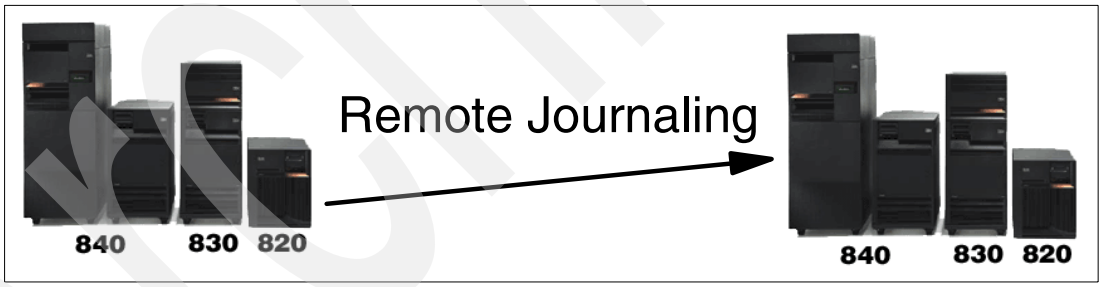


Figure 9-10 Remote journaling

Journaling to an independent disk pool (Figure 9-11) can enhance either local or remote journaling. If the independent disk pool is non-switched, private, then you have the situation of better isolation then local journaling. If the independent disk pool is a switched resource between two systems, remote journaling can be done to an independent disk pool on the remote system. When you need to take the remote system down for maintenance or when a failure occurs, that disk pool can be switched to another system where the journaling can continue.

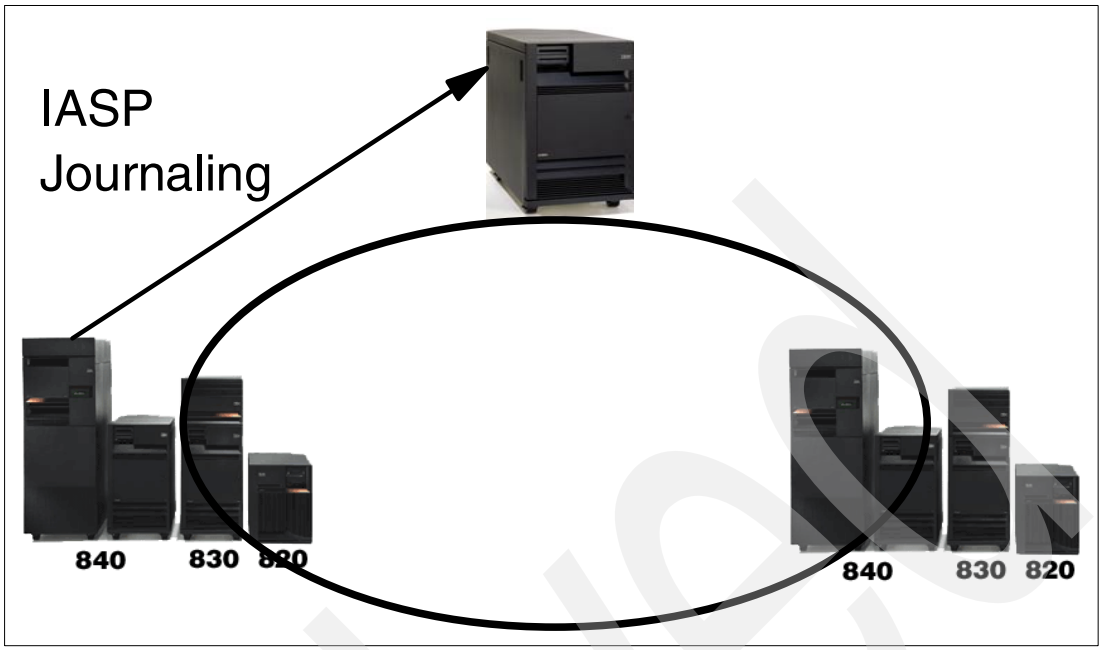


Figure 9-11 Journaling with independent disk pools

10

SAP in an IASP

This chapter explains how you can install the mySAP.com application suite in a cluster-based independent auxiliary storage pool (IASP)-environment. This environment allows you to switch over the SAP database from one system to another without rebooting the target system.

The mySAP.com suite is available for iSeries. It delivers a comprehensive e-business platform designed to help companies collaborate and succeed, regardless of their industry or network environment. For more information about mySAP.com, see:

http://www.sap.com/solutions/

The major components of the mySAP.com suite are SAP systems. An *SAP system* includes one kernel, one database (SQL collection), and the SAP directory structure being placed in the integrated file system (IFS) of the iSeries server. The kernel contains the executable code for the SAP application server. The database includes the customer data and the Advanced Business Application Programming (ABAP) modules. The characteristic of an SAP system is specified by its SAP system profile located in the IFS.

An SAP system has one database server and one or more application servers. The application server contains the executional environment, which is a collection of processes called *SAP instance*. An instance is connected to only one database. Multiple instances can be defined for a single SAP system. An SAP instance is implemented as an iSeries subsystem with SAP work processes running in it. *Work processes* are jobs within the instance subsystem that actually perform the work. The characteristic of an instance is specified by its instance profile located in the IFS.

The SAP system can be installed in a two-tier configuration (single machine for database and application server) or a three-tier configuration (network of machines where one machine is the database server and one or more machines are the application server). For more information about the architecture of an SAP system running on iSeries, see the IBM Redbook *Implementing SAP R/3 on OS/400*, SG24-4672.

You can install SAP in an IASP in the following manner:

- **SAP 2-tier without using a switchable IP address**: This can be done for all SAP releases being certified for OS/400 V5R2.
- **SAP 2-tier using a switchable IP-dress or SAP 3-tier**: This requires SAP kernel release 4.6D or higher and OS/400 V5R2.

This chapter explains how to set up SAP 2-tier system in a cluster-based IASP environment using a switchable IP address. It also explains how to perform a manual switchover using a typical customer scenario. In this scenario, the SAP system IAS is already installed on an iSeries server named CL4. It should be made switchable to a secondary system in the cluster named CL3.

For more information how to set up general SAP systems in an IASP environment, refer to informational note 568820 available from SAP Online Service & Support (OSS).

Note: You must have the required OS/400 and the iSeries skills to perform the actions that are documented in this chapter.

10.1 SAP in a non-clustered environment

Figure 10-1 shows an SAP system running in a non-clustered environment. This means that the data cannot be switched to another environment. All definitions are based on a typical SAP 2-tier implementation in which the application and database environment runs on one iSeries server (CL4). All data is stored in the system ASP (ASP1), except the journal receivers being placed in the user ASP (ASP2).

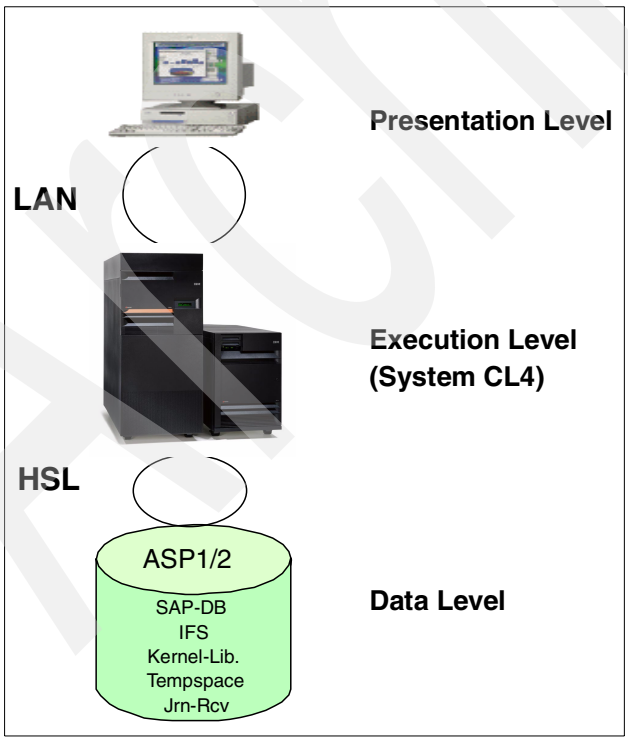


Figure 10-1 SAP environment on a single non-clustered system

Figure 10-2 shows a more detailed example of this implementation.

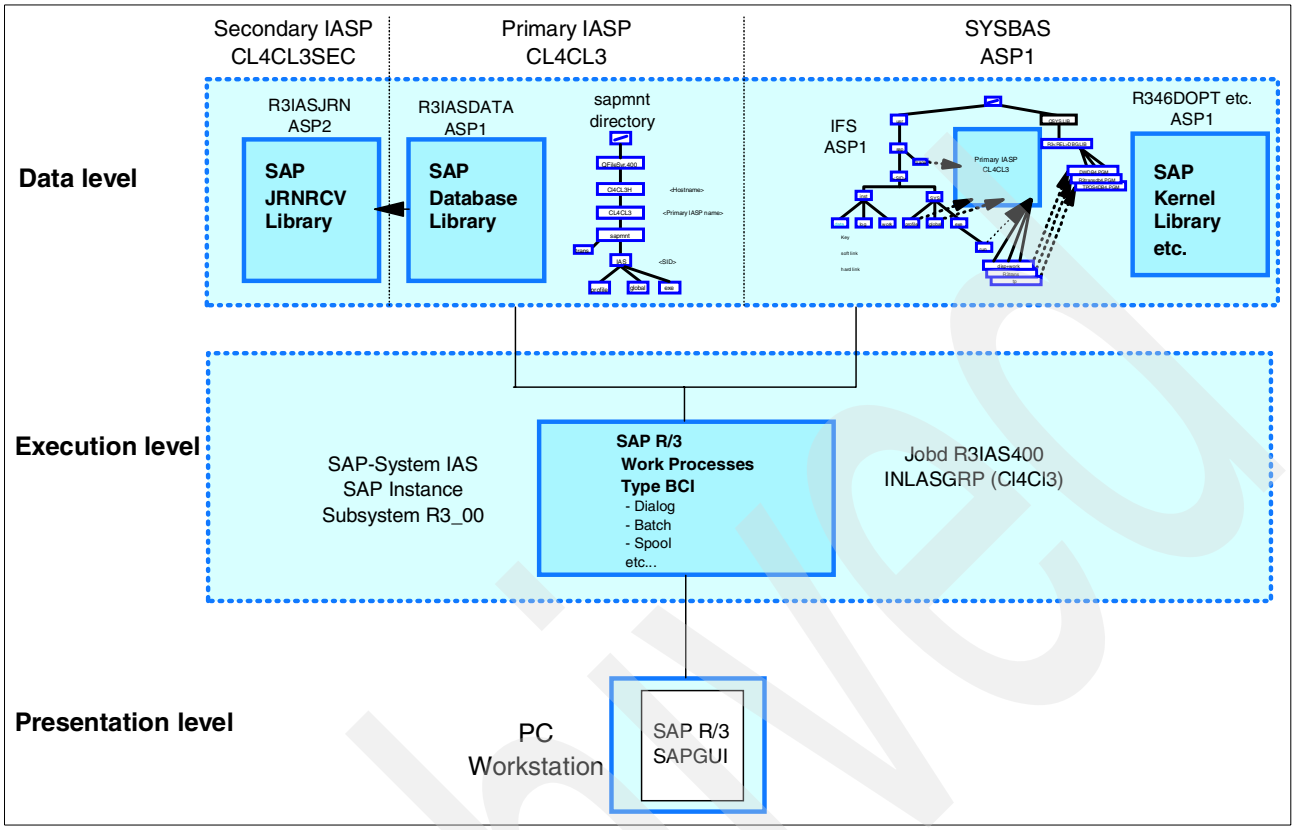


*Figure 10-2 SAP implementation on iSeries (*SYSBAS only)*

In this example, the entire SAP system IAS (application and database) runs on one non-clustered environment. All data, for example the SAP database R3IASDATA, the journal-receiver library R3IASJRN, the kernel library R346DOPT and other libraries, the IFS of the SAP application, and the temporary disk space are placed in non-switchable ASPs.

Figure 10-3 shows the SAP directory structure for the SAP system IAS in the IFS, which is completely located in the system ASP (ASP1). This structure contains static data, such as links to the SAP kernel library, that normally doesn't change. It also contains data, such as the configuration files for the entire SAP system and the application instances, that can change sometimes. These types of changeable data are physically located in the sapmnt directory structure.

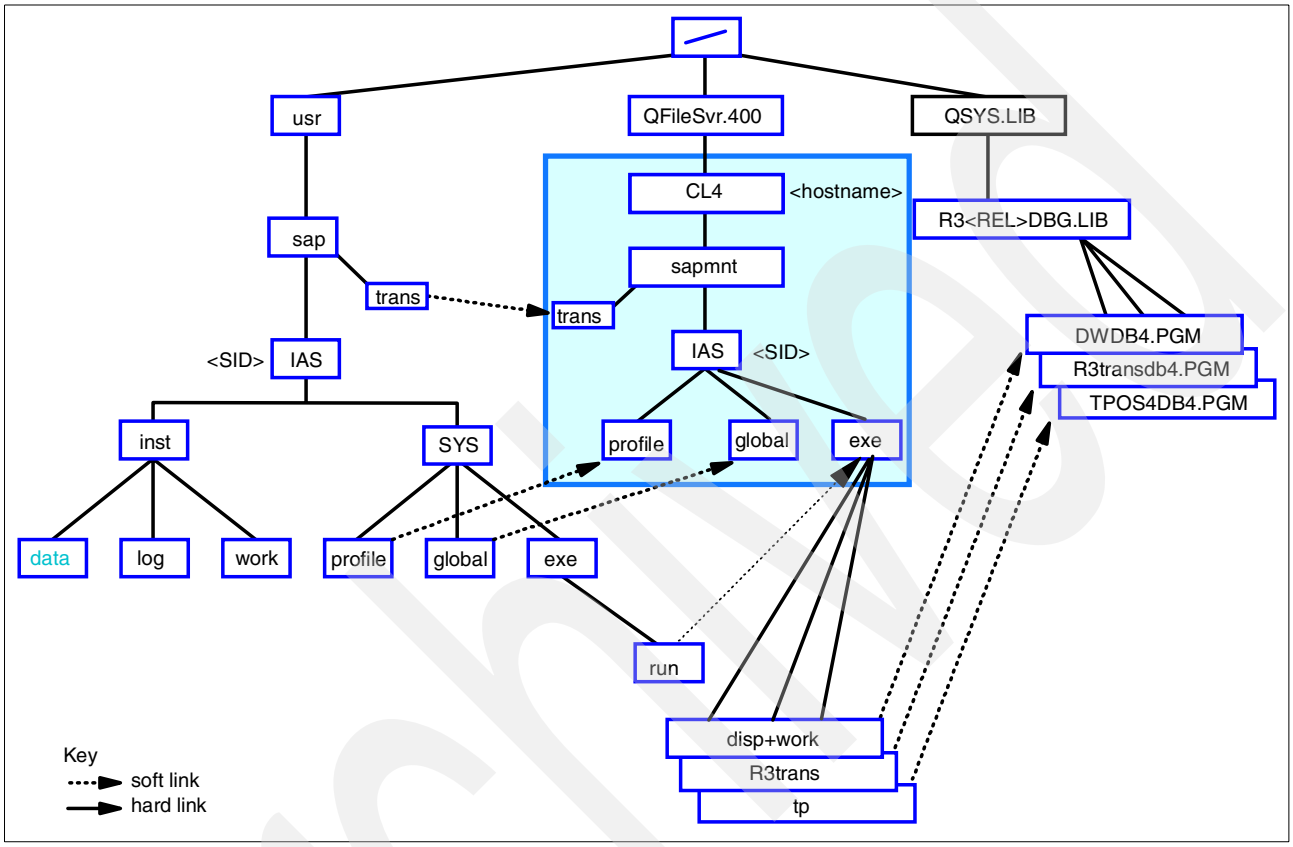


Figure 10-3 SAP directory structure in system ASP

10.2 Implementation overview

We want to change this environment to a clustered one. This means that two iSeries servers and a switchable tower containing IASPs are connected in high-speed link (HSL) loops as shown in Figure 10-4. This switchable tower containing frequently changed data (SAP database, journal/receiver, SAP system, and instance profile) can be switched between both system (CL4 for production and CL3 for backup).

These systems have a non-switchable system ASP that contains static data (SAP kernel, static part of the SAP directory structure, etc.) or temporary storage. This static data must be installed on both systems. The frequently changed data in the switchable tower needs to be available only once in an IASP.

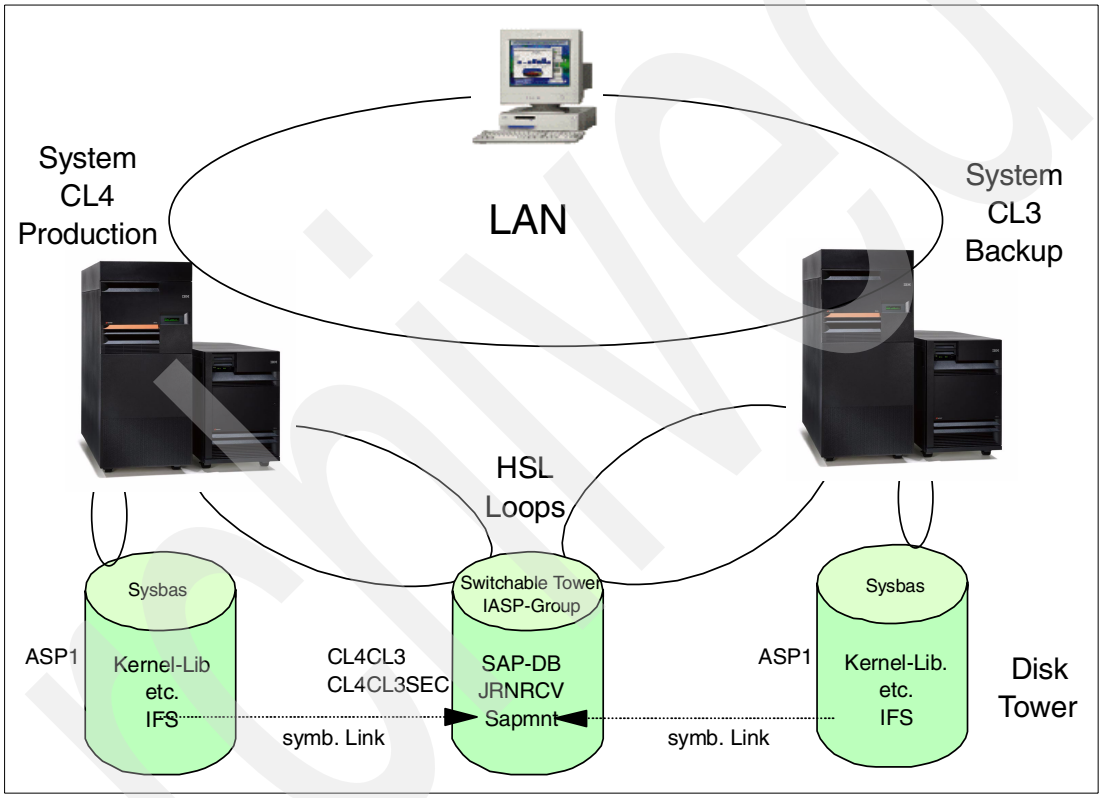


Figure 10-4 SAP landscape for IASP

Figure 10-5 shows a more detailed view of SAP running in an IASP environment. All data that can be changed frequently is stored in the IASP group CL4CL3. In the primary IASP (CL4CL3), we must install the SAP database (R3IASDATA) and the sapmnt directory. The secondary IASP (CL4CL3SEC) contains the journal receiver library R3IASJRN. All remaining libraries (kernel, SQL packages, R3IAS400, R3400, R3QTEMP, R3SYS, R3WRK00) and temporary storage remain in the system ASP (ASP1).

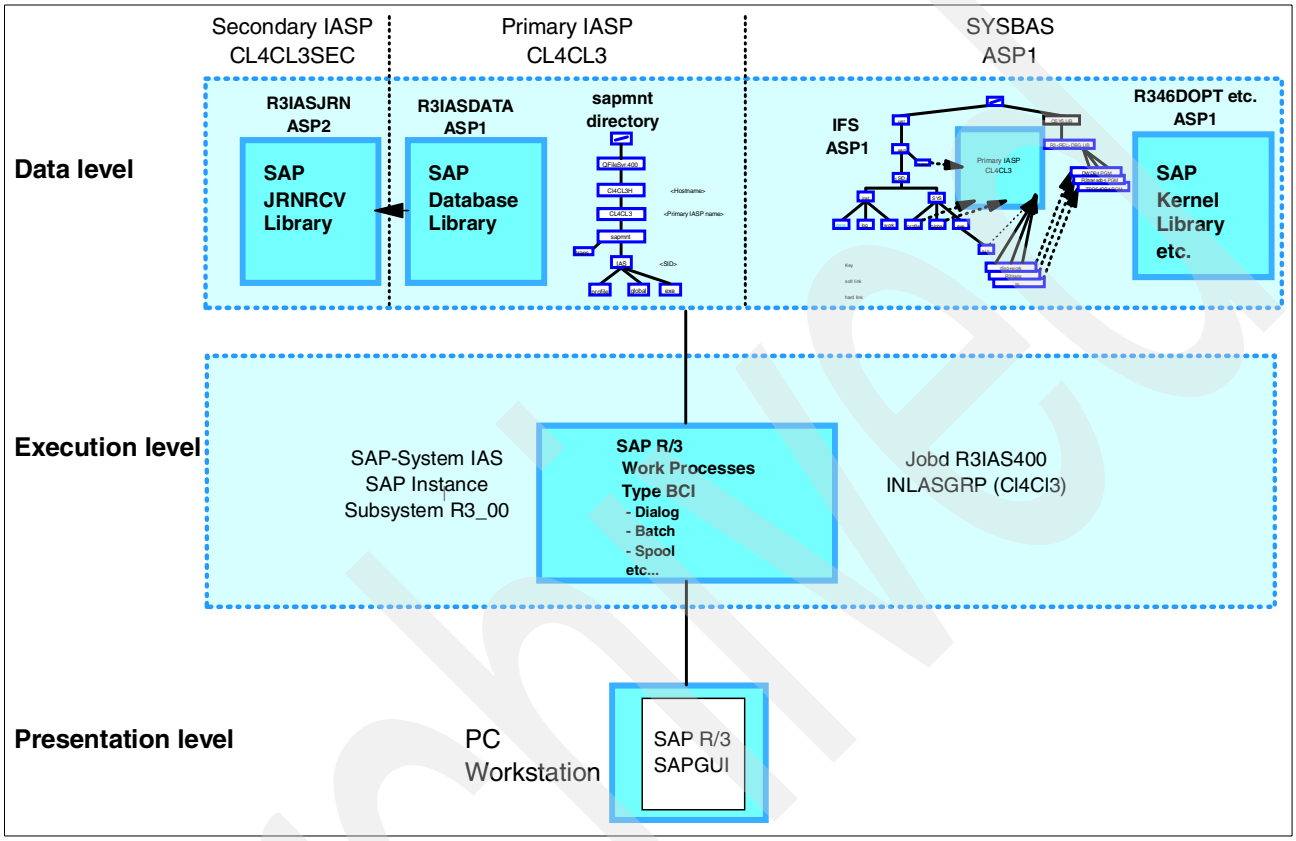


Figure 10-5 SAP implementation in an IASP

Figure 10-6 shows the part of the SAP file structure in the IFS that contains static data and that remains in the system ASP.

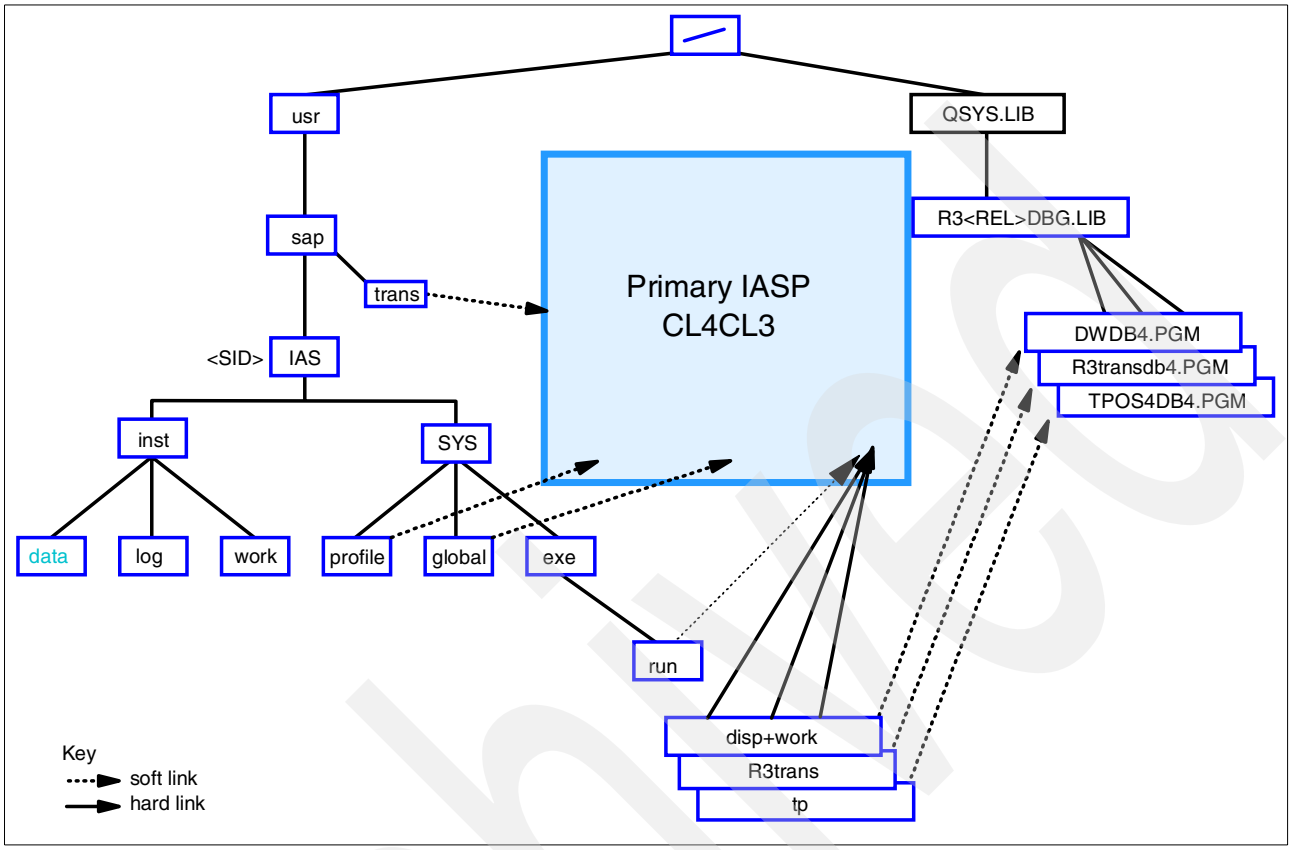


*Figure 10-6 SAP directory structure (*SYSBAS)*

The sapmnt directory that contains the SAP profile data can be changed sometimes. It is shared between all SAP systems (for example development, quality assurance, and production). This directory is located in the primary IASP. You can access it both from the production system CL4 and backup system CL3 (Figure 10-7).

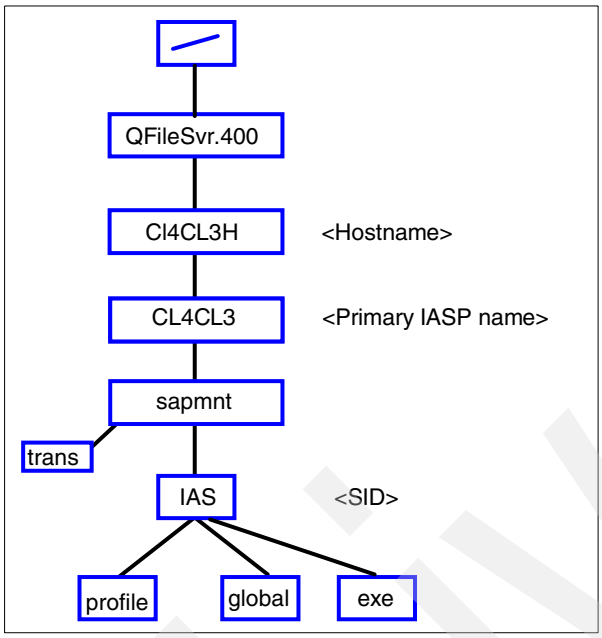


Figure 10-7 SAP directory structure (IASP)

10.3 The cluster environment

Figure 10-8 and Figure 10-9 show how the cluster SAP is set up. Two nodes are part of this cluster: CL4 for the SAP production system and CL3 as a backup. For more information about how to set up an iSeries cluster, see Chapter 7, “Switchable setup” on page 81.

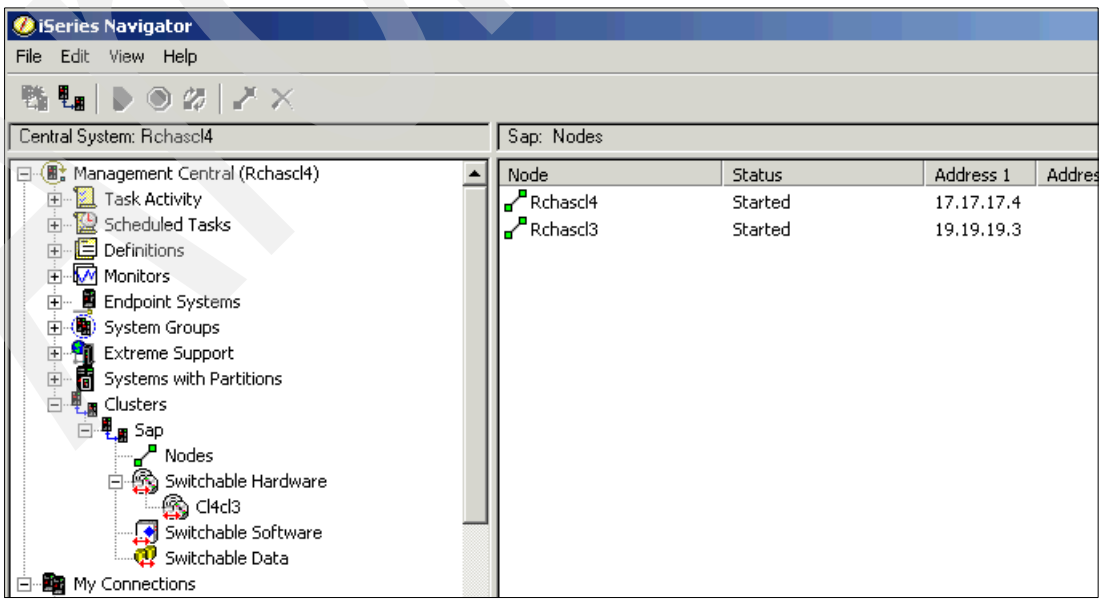


Figure 10-8 Cluster SAP: Nodes overview

Figure 10-9 shows that cluster resource group (CRG) CL4CL3 can be switched between both nodes. In our example, CL3 is assigned. You click the Switch button to switch over to backup system CL4.

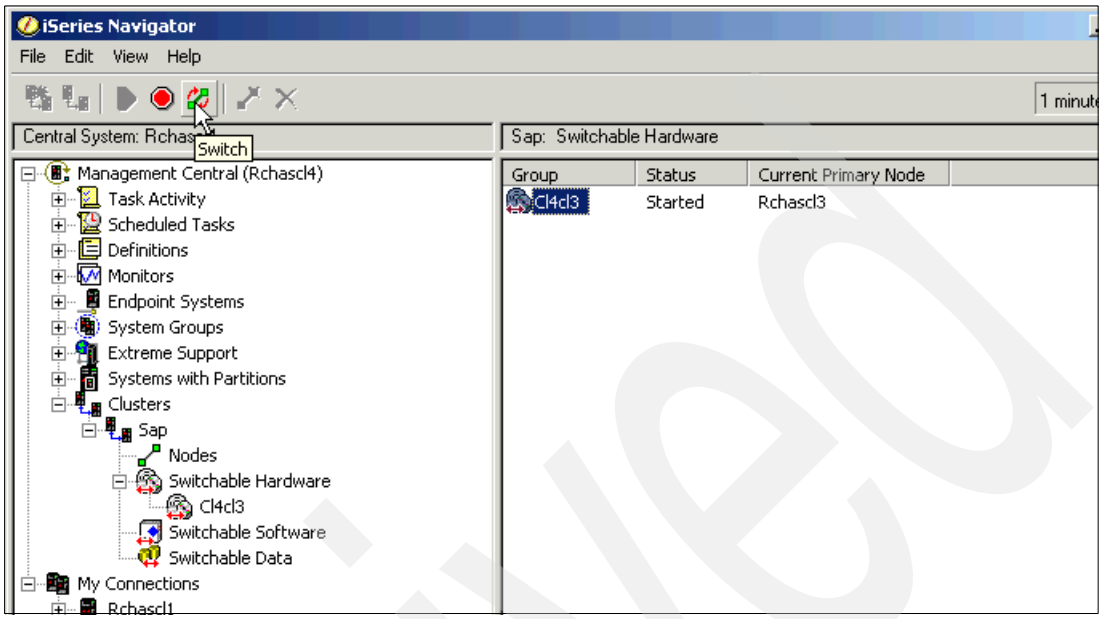


Figure 10-9 Cluster SAP: Switchable hardware

Both systems can assign cluster resource group CL4CL3, which consists of a primary IASP CL4CL3 and a secondary IASP CL4CL3SEC (Figure 10-10). To see which objects are stored in the primary and secondary IASP, refer to Figure 10-5 on page 132.

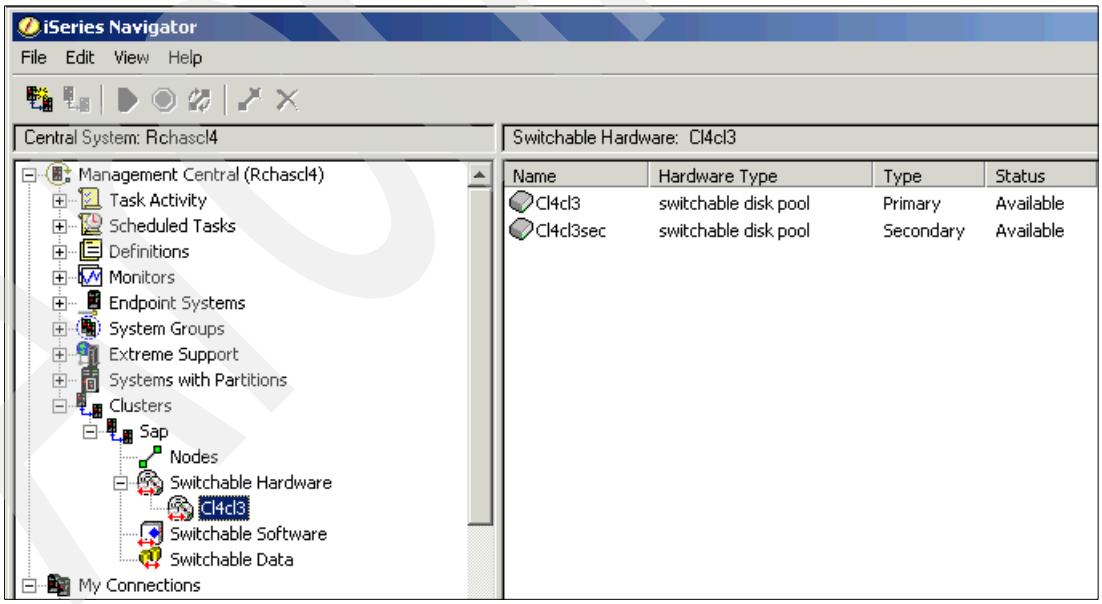


Figure 10-10 Cluster SAP: CRG definition

The primary IASP CL4CL3 is defined as a a switchable disk pool (Figure 10-11). That means that this primary pool, the related secondary pool (CL4CL3SEC), and the adjacent IP address 7.7.7.7 can be switched between the systems that belong to the cluster group SAP.

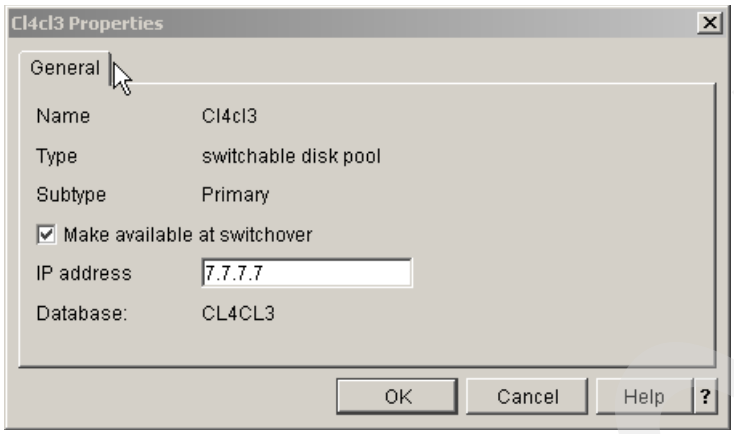


Figure 10-11 IASP properties

10.4 Setting up the SAP system

To take advantage of IASPs, you set up the SAP system manually. The result returns some duplication of libraries and their contents on all iSeries system ASPs in your multisystem environment. However, the R3IASDATA library, the journal, journal receivers, and some IFS directories are common between the iSeries servers.

You must perform this setup on the primary iSeries with the IASP varied on. Then repeat the setup on any secondary (backup) iSeries with the IASP varied off the primary and varied on to the secondary.

10.4.1 Manually setting up the SAP system

Before you continue, you must make the primary (CL4CL3) and the secondary (CL4CL3SEC) IASP available to your system. Enter the following command to vary on and vary off IASP devices:

```
WRKCFGSTS CFGTYPE(*DEV) CFGD (*ASP)
```

In our example (Figure 10-12), the IASPs CL4CL3 and CL4CL3SEC are available for system CL4.

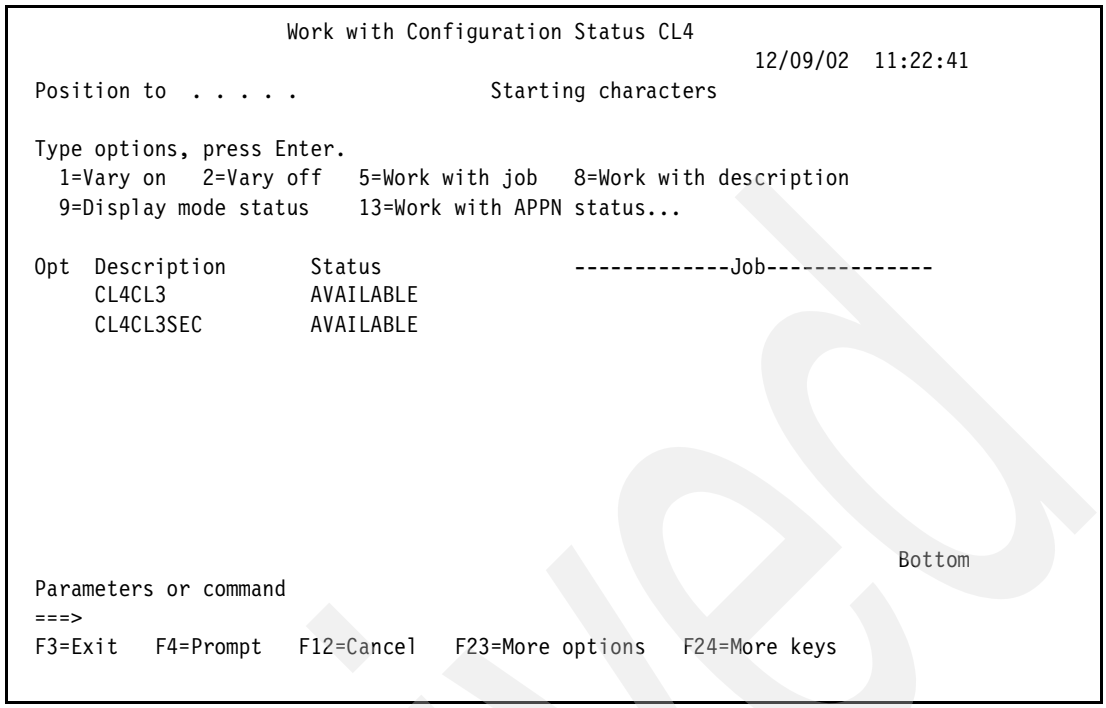


Figure 10-12 WRKCFGSTS to work with ASP devices

Then you must follow these steps:

1. Make sure your SAP systems are fully stopped on all iSeries servers in your environment.
2. Ensure that the primary IASP has the correct authorities. To do this, enter the command:

   ```
   CHGPGP OBJ('/CL4CL3') NEWPGP(R3GROUP) DTAAUT(*RWX) OBJAUT(*ALL) RVKOLDAUT(*NO)
   SYMLNK(*NO)
   ```

3. For all secondary IASPs linked to the primary IASP, enter the command:

   ```
   CHGPGP OBJ('/CL4CL3SEC') NEWPGP(R3GROUP) DTAAUT(*RWX) OBJAUT(*ALL) RVKOLDAUT(*NO)
   SYMLNK(*NO)
   ```

You may have an environment where:

- The SAP system is not installed on any iSeries servers, and therefore, you must perform new installations of the SAP system on all iSeries servers.
- The SAP system is already installed on all iSeries servers in your environment.
- The SAP system is already installed on one iSeries server in your multisystem environment (call this your primary iSeries) and new installations are needed on your secondary iSeries servers. You can learn more about this scenario in 10.5, “Setting up the IASP” on page 138.

SAP system partially installed: Your multisystem environment is partially installed. The SAP system is installed on a primary iSeries server in your environment but not installed on any secondary iSeries servers.

10.5 Setting up the IASP

In the following sections, you install an SAP system of the same release and patch level on all secondary iSeries servers in the environment. However, this installation must be modified. In the following instructions you are asked to delete the R3IASDATA library and the R3IASJRN library on the secondary systems only.

To set up the IASP for the SAP environment, you must first logon to an OS/400 session. We recommend that you log on as QSECOFR. Then change the job's thread to include IASP CL4CL3 by using the Set Auxiliary Storage Pool Group (SETASPGRP) command as shown here:

```
SETASPGRP ASPGRP(CL4CL3)
```

This make the IASP visible to the QSECOFR user job.

10.5.1 Setting up the libraries

Important: You must perform the Save Library (SAVLIB) and Restore Library (RSTLIB) process only using one iSeries server (your primary system) in the multisystem environment.

Only one copy of the data and journal libraries and the IFS directories exists in the environment. They reside on the IASP. You must delete any duplicate copies of these objects on the other iSeries server in the environment.

To vary on an IASP to an iSeries, duplicate library names must not exist between any system ASP and an IASP on that system. Otherwise, the IASP cannot be varied on. Duplicate library names can exist between different IASPs. However, this should not apply here.

SAVLIB/RSTLIB process

On your primary iSeries server, complete the following steps:

1. Perform a SAVLIB of the R3IASDATA library to either a *SAVF or to a system device. The following command uses the DEV(*SAVF) option but you can change this to use the system device you have available:

   ```
   SAVLIB LIB(R3IASDATA) DEV(*SAVF) SAVF(savflib/savf) SAVACT(*LIB)
   ```

2. After a successful SAVLIB of the R3IASDATA library, delete the library from the system ASP:

   ```
   DLTLIB LIB(R3IASDATA) LIB(R3IASDATA)
   ```

3. Restore the R3IASDATA library to the IASP. To do this, first create the R3IASDATA library on the IASP with the Create Library (CRTLIB) command:

   ```
   CRTLIB LIB(R3IASDATA) ASP(*ASPDEV) ASPDEV(CL4CL3) TEXT('R3 sid DATA library on IASP
   primary')
   ```

4. Make sure the owner of the R3IASDATA library is user profile IASOWNER. Enter the command:

   ```
   CHGOBJOWN OBJ(R3IASDATA) OBJTYPE(*LIB) NEWOWN(IASOWNER)
   ```

5. Restore the data library from the media that you specified on the SAVLIB command. Enter the RSTLIB command (the following example uses a *SAVF):

   ```
   RSTLIB SAVLIB(R3IASDATA) DEV(*SAVF) SAVF(savflib/savf) MBROPT(*ALL) ALWOBJDIF(*ALL)
   OPTION(*NEW) RSTASPDEV(CL4CL3)
   ```

10.5.2 User profile and job description changes

The job description (type *JOBD) R3_00 in library R3IAS400, where *00* is the instance number of your SAP system and IAS is the system ID of your SAP system, should already exist. Then follow these steps:

1. Change the JOBD parameter values to match your SAP system ID and instance number. Then change the value in the INLASPGRP() parameter to the device name of your IASP. Use the following command:

   ```
   CHGJOBD JOBD(R3IAS400/R3_00) INLASPGRP(CL4CL3)
   ```

   The change to the R3_00 job description makes the objects on the IASP visible to any user profile that uses this job description.

2. The user profile IAS00 should already reference the R3_00 job description. If not, use the command to make the objects on the IASP “visible” to the user profile:

   ```
   CHGUSRPRF USRPRF(IAS00) JOBD(R3IAS400/R3_00)
   ```

3. Change the IASOFR user profile so it references the R3_00 job description. Use the following command to make the objects on the IASP visible to the user profile:

   ```
   CHGUSRPRF USRPRF(IASOFR) JOBD(R3IAS400/R3_00)
   ```

You now have changed the user profiles on the primary. Since these changes are not replicated to other servers in the cluster, you must manually make the same changes to other servers in the cluster (the secondary CL4).

10.5.3 Journal and journal receiver

Complete the following steps:

1. If the R3IASJRN library exists, delete the library:

   ```
   DLTLIB LIB(R3IASJRN)
   ```

2. Create the journal library with the CRTLIB command and change the appropriate parameters as you did earlier for the CRTLIB R3IASDATA command:

   ```
   CRTLIB LIB(R3IASJRN) ASP(*ASPDEV) ASPDEV(CL4CL3SEC) TEXT('R3 sid Journal library on
   primary/secondary IASP')
   ```

   Note: If you created a secondary ASP in the same group as the primary IASP, these ASPs are linked in the same group. We recommend that you create the journal receiver library on the secondary ASP.

3. Make sure the owner of the R3IASJRN library is the IASOWNER profile. Use the command:

   ```
   CHGOBJOWN OBJ(R3IASJRN) OBJTYPE(*LIB) NEWOWN(IASOWNER)
   ```

4. Create the journal receiver in the R3IASJRN library and change the appropriate parameters with the command:

   ```
   CRTJRNRCV JRNRCV(R3IASJRN/QSQJRN0001) ASP(*LIBASP) THRESHOLD(200000)
   ```

   If the journal *already exists* in the R3IASDATA library, issue the change journal command to attach the journal to the receiver that you just created. Use the command:

   ```
   CHGJRN JRN(R3IASDATA/QSQJRN) JRNRCV(R3IASDATA/QSQJRN0001)
   ```

If the journal *does not exist* in the R3IASDATA library, create the journal in the R3IASDATA library and change the appropriate parameters with the command:

```
CRTJRN JRN(R3IASDATA/QSQJRN) JRNRCV(R3IASJRN/QSQJRN0001) ASP(*LIBASP) MNGRCV(*SYSTEM)
DLTRCV(*NO) RCVSIZOPT(*RMVINTENT *MAXOPT2) MINENTDTA(*FILE)
```

5. To start journaling and fix up the SAP system, enter the AS4FIXFILE command in the SAP kernel library R346DOPT:

```
AS4FIXFILE DBLIB(R3IASDATA) CHGOWN(*YES) CHGFILE(*YES) NEWOWN(*SIDOWNER)
```

10.5.4 IFS files

You must move IFS directories and contained files to the IASP. A sample CL program and command are available to help you do this. The IFS directories involved are:

- /usr/sap/*sid*/SYS/profile SYMLNK->DIR /sapmnt/*sid*/profile
- /usr/sap/*sid*/SYS/global SYMLNK->DIR /sapmnt/*sid*/global
- /usr/sap/*sid*/SYS/exe/run SYMLNK->DIR /sapmnt/*sid*/exe

You must change these directories and the symbolic links. In our example, the directories are located on /CL4CL3/sapmnt/IAS/…

The symbolic links point to the corresponding directories, for example:

- /usr/sap/IAS/SYS/profile -> QFileSvr.400/CL4CL3/sapmnt/IAS/profile
- /usr/sap/IAS/SYS/global -> QFileSvr.400/CL4CL3/sapmnt/IAS/global
- /usr/sap/IAS/SYS/exe/run -> QFileSvr.400 /CL4CL3/sapmnt/IAS/exe
- /usr/sap/trans SYMLNK->DIR QFileSvr400/CL4CL3/sapmnt/trans

This last directory “trans” is discussed in detail later in “CL production iSeries step 2 (trans directory)” on page 143 and in “CL backup iSeries step 2 (trans directory)” on page 144.

10.5.5 TCP/IP for switchable disk pool

As shown in Figure 10-13 and Figure 10-14, we use 7.7.7.7 as the switchable IP address in our cluster environment. This address is available regardless of whether you are working on your production environment CL4 or switch over to the backup system CL3.

All SAP work processes and the SAP graphical front end SAPGUI use this address via host name CL4CL3H. You must define this host name on all nodes in the cluster as shown in this chapter. You must also define it in the SAP configuration file as explained in 10.5.7, “Modifying SAP configuration files for IASP” on page 145. Figure 10-13 and Figure 10-14 show the relationship between the graphical overview of the environment and the actual 5250 display.

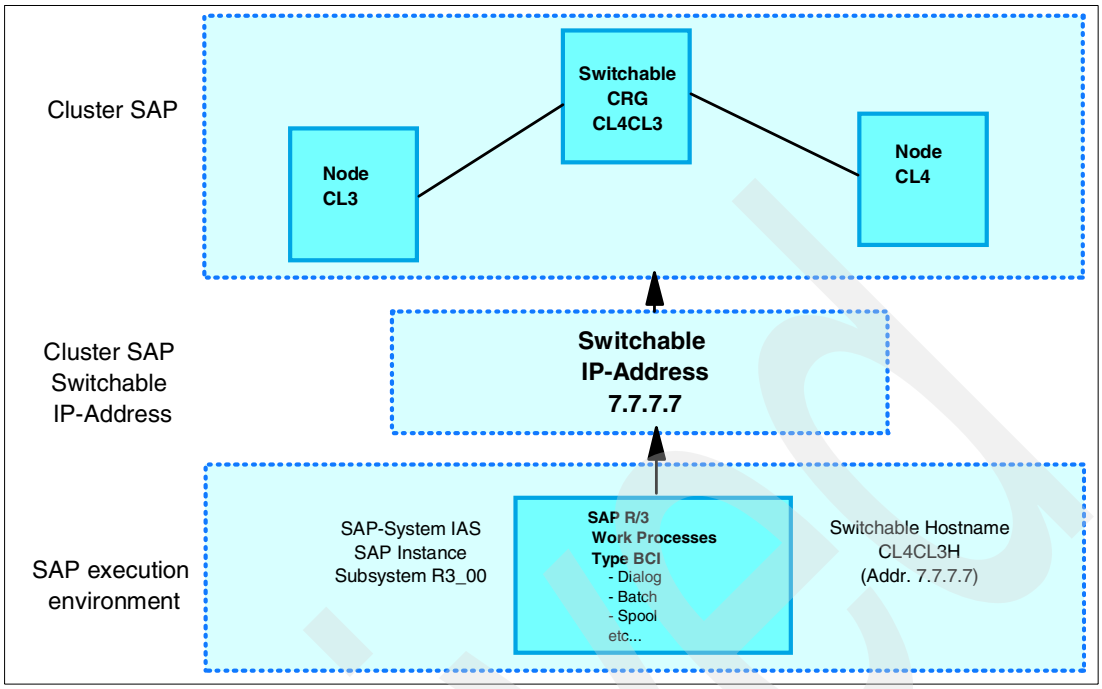


Figure 10-13 Switchable IP address for cluster SAP

```
                          Display CRG Information

Cluster  . . . . . . . . . . . . . :   SAP
Cluster Resource Group . . . . . :   CL4CL3
Reporting Node Identifier  . . . : CL3
Consistent Information in Cluster:   *YES

                       Configuration Object Information

Configuration   Object    Device    Device      Vary       Server
 Object Name    Type      Type      Subtype     Online     Ip Address
 CL4CL3         *DEVD     *ASP      Primary     *YES       7.7.7.7
 CL4CL3SEC      *DEVD     *ASP      Secondary   *PRIMARY   *NONE


                                                                    Bottom
Number of Device List Entries  . :   2


Enter=Continue   F12=Cancel   F3=Exit   F1=Help
```

Figure 10-14 DSPCRGINF panel 2

For both systems CL3 and CL4, the switchable disk pool (IP address 7.7.7.7) must be accessible under the same IP address and the same host name (in our example CL4CL3H).

Figure 10-15 shows the TCP/IP interfaces, which include the switchable IP address.

```
                        Work with TCP/IP Interfaces
                                                                System: CL4
Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   9=Start   10=End

     Internet         Subnet              Line      Line
Opt  Address          Mask             Description  Type

     7.7.7.7          255.255.255.255  *VIRTUALIP   *NONE
     9.5.123.74       255.255.255.0    SITETRN      *TRLAN
     17.17.17.4       255.255.255.0    ETHLINECL4   *ELAN
     51.51.51.4       255.255.255.0    *OPC         *NONE
     127.0.0.1        255.0.0.0        *LOOPBACK    *NONE




                                                                       Bottom
F3=Exit      F5=Refresh   F6=Print list   F11=Display interface status
F12=Cancel   F17=Top      F18=Bottom
```

Figure 10-15 TCP/IP interface for switchable IP address

Figure 10-16 shows the Add Host Table Entry display. For the switchable IP address, add entries for attributed to the switchable host name, CL4CL3H. You must add the host table entry on both the cluster nodes CL3 and CL4.

```
                     Work with TCP/IP Host Table Entries
                                                                System: CL3
Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   7=Rename

     Internet         Host
Opt  Address          Name

     7.7.7.7          CL4CL3H
                      CL4CL3H.RCHLAND.IBM.COM™
     17.17.17.4       RCHASCL4
                      RCHASCL4.RCHLAND.IBM.COM
     19.19.19.3       RCHASCL3
                      RCHASCL3.RCHLAND.IBM.COM
     127.0.0.1        LOOPBACK
                      LOCALHOST



                                                                       Bottom
F3=Exit   F5=Refresh   F6=Print list   F12=Cancel   F17=Position to
Internet address entry 7.7.7.7 added to host table.
```

Figure 10-16 CFGTCP to add a host name for the switchable disk pool

To have a switchable host name for the content of the IASP, add a directory entry in QFileSvr.400 for the host name CL4CL3H as shown in Figure 10-17.

```
                          Create Directory (MKDIR)

Type choices, press Enter.

Directory  . . . . . . . . . . .   /QFILESVR.400/CL4CL3H

Public authority for data  . . .   *INDIR        Name, *INDIR, *RWX, *RW...
Public authority for object  . .   *INDIR        *INDIR, *NONE, *ALL...
               + for more values
Auditing value for objects . . .   *SYSVAL       *SYSVAL, *NONE, *USRPRF...




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

Figure 10-17 Switchable host name for IASP

10.5.6 Control Language (CL) source and corresponding command

You perform the following steps, CL production iSeries step 1 and CL production iSeries step 2 (trans directory), only once on the production iSeries server (CL4). Then you perform “CL backup iSeries step1” on page 144 and “CL backup iSeries step 2 (trans directory)” on page 144 only on the backup iSeries server (CL3). The IASP must be varied off of the production system and varied on to the backup system.

Restore the library SAPIASP from the file attachment IASPSAVF to your iSeries to use the programs and commands to assist you with the following steps. To successfully compile and run any of these programs, make sure your kernel library is in the library list (the user portion is sufficient) of the job that you will compile or that will run the job that compiles or runs the restore programs and commands.

CL production iSeries step 1

A sample CL source file (lnk2iasp.qclsrc) handles the profile, global, and exe/run directories. Use the CL *PGM supplied in the SAPIASP library or create the CL program and run it with the proper authorities.

You can use the command source file (lnk2iasp.qcmdsrc) to call the program (LNK2IASP), or you can call the program directly with the correct parameters. The parameters shown in the following command example apply to the example described in this chapter. When you run this command, change these values to those in your specific environment.

```
LNK2IASP SID(IAS) IASP(CL4CL3) HOSTNAME(*SAME)
```

CL production iSeries step 2 (trans directory)

In this case, the transport directory should reside on the IASP. Therefore, you must use a sample CL source file (transmov.qclsrc) that can handle the trans directory. The transport directory is still visible to all other SAP systems in your environment because the symbolic links to the actual directories are still on the system ASP.

Use the CL *PGM supplied in the SAPIASP library or create the CL program and run it with the proper authorities.

You can use the command source file (transmov.qcmdsrc) to call the program (TRANSMOV), or you can call the program directly with the correct parameters. The parameters shown in the following command example apply to the example described in this chapter. When you run this command, change these values to those in your specific environment.

```
TRANSMOV IASP(CL4CL3) HOSTNAME(*SAME)
```

CL backup iSeries step1

A sample CL source file (lnksonly.qclsrc) handles the symbolic links to the profile, global, and exe/run directories only. Use the CL *PGM supplied in the SAPIASP library or create the CL program and run it with the proper authorities.

You can use the command source file (lnksonly.qcmdsrc) to call the program (LNKSONLY), or you can call the program directly with the correct parameters. The parameters shown in the following command example apply to the example described in this chapter. When you run this command, change these values to those in your specific environment.

```
LNKSONLY SID(IAS)) IASP(CL4CL3) HOSTNAME(CL4CL3H)
```

Figure 10-18 shows the link structure that is created by the LINKSONLY command.

```
                          Display Symbolic Link

Object link  . . . . . :   /usr/sap/IAS/SYS/profile



Content of Link  . . . :   /QFileSvr.400/CL4CL3H/CL4CL3/sapmnt/IAS/profile




Press enter to continue.

F3=Exit   F5=Refresh   F12=Cancel   F14=Work with link content
F22=Display entire field
```

Figure 10-18 Symbolic links for the backup system

CL backup iSeries step 2 (trans directory)

Note: Only complete this step if you completed “CL production iSeries step 2 (trans directory)” on page 143.

A sample CL source file (translnk.qclsrc) handles the symbolic link to the trans directory only. Use the CL *PGM supplied in the SAPIASP library or create the CL program and run it with the proper authorities.

You can use the command source file (translnk.qcmdsrc) to call the program (TRANSLNK), or you can call the program directly with the correct parameters. The parameters shown in the following command example apply to the example described in this chapter. When you run this command, change these values to those in your specific environment.

```
TRANSLNK IASP(CL4CL3) HOSTNAME(CL4CL3HOST)
```

Figure 10-19 shows the link structure that is created by the TRANSLNK command.

```
                          Display Symbolic Link

Object link  . . . . . :   /usr/sap/trans



Content of Link  . . . :   /QFileSvr.400/CL4CL3H/CL4CL3/sapmnt/trans









Press enter to continue.

F3=Exit   F5=Refresh   F12=Cancel   F14=Work with link content
F22=Display entire field
```

Figure 10-19 Link to the SAP transport directory

10.5.7 Modifying SAP configuration files for IASP

Update the configuration files for SAP to use the host name with the associated switchable IP address (CL4CL3H).

SAP instance data area

Change the SAP instance data area R3IAS400 for instance 00:

```
CHGDTAARA DTAARA(R3IAS400/R3_00 (8 8)) VALUE(CL4CL3H)
```

Trans directory

Update the system.cfg file in the trans directory, which is shown in Figure 10-20.

```
 Edit File: /usr/sap/trans/config/IAS/SYSTEM.CFG
 Record :       1   of       9 by   8           Column :    1     59 by  74
 Control :

CMD ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+
     ************Beginning of data**************
    # Created on 20021203165544
    #
    SAPSYSTEMNAME = IAS
    SAPDBHOST = CL4CL3H
    SAPMSHOST = CL4CL3H
    #
    db4/kernel = R346DOPT
    db4/dbasp = 1
    db4/jrnrcvasp = 1
     ************End of Data********************




 F2=Save   F3=Save/Exit   F12=Exit   F15=Services   F16=Repeat find
 F17=Repeat change  F19=Left  F20=Right
```

Figure 10-20 SAP system profile

Change the instance profile for the current system. In this example, the instance profile for the current system is CL4_00.CFG as shown in Figure 10-21.

```
 Edit File: /usr/sap/trans/config/IAS/CL4_00.CFG
 Record :       1   of       9 by   8           Column :    1     59 by  74
 Control :

CMD ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+
     ************Beginning of data**************
    # Created on 20021203165546
    #
    SAPSYSTEMNAME = IAS
    INSTANCE_HOST = CL4CL3H
    SAPSYSTEM = 00
    #
    INSTANCE_ROLE = DVEBMGS
    #
    db4/kernel = R346DOPT
     ************End of Data********************




 F2=Save   F3=Save/Exit   F12=Exit   F15=Services   F16=Repeat find
 F17=Repeat change  F19=Left  F20=Right
```

Figure 10-21 Instance profile in the trans directory

Create a new instance profile for system CL3_00.CFG (Figure 10-22) by copying the original one from system CL4.

```
                           Work with Object Links

 Directory  . . . . :   /usr/sap/trans/config/IAS

 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
       .                     DIR
       ..                    DIR
       RCHASCL3_00.CFG       STMF
       RCHASCL4_00.CFG       STMF
       SYSTEM.CFG            STMF




                                                                      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 10-22 Copying the instance profile for system CL3 in the trans directory

SAP system profile

Change the SAP default profile (DEFAULT.PFL), as shown in Figure 10-23, and the instance profile IAS_DVEBMGS00_CL4 (Figure 10-24) to the switchable host name.

```
 Edit File: /usr/sap/IAS/SYS/profile/DEFAULT.PFL
 Record :       1   of       7 by   8           Column :    1     59 by  74
 Control :

CMD ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+
     ************Beginning of data**************
    SAPSYSTEMNAME = IAS
    SAPDBHOST = CL4CL3H
    rdisp/mshost = CL4CL3H
    rdisp/vbname = $(rdisp/myname)
    rdisp/enqname = CL4CL3H_IAS_00
    rdisp/btcname = CL4CL3H_IAS_00
    rdisp/bufrefmode = sendoff,exeauto
    dbs/db4/rdbname = CL4CL3
     ************End of Data*********************




 F2=Save   F3=Save/Exit   F12=Exit   F15=Services   F16=Repeat find
 F17=Repeat change  F19=Left  F20=Right
```

Figure 10-23 Changing the default profile

```
 Edit File: /usr/sap/IAS/SYS/profile/IAS_DVEBMGS00_CL4
 Record :       1   of      57 by   8           Column :    1     70 by  74
 Control :

CMD ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+
     ************Beginning of data**************
    #---------------------------------------------------------------------
    # Instance profile
    #---------------------------------------------------------------------

    SAPSYSTEMNAME = IAS
    INSTANCE_NAME = DVEBMGS00
    SAPSYSTEM     = 00
    SAPLOCALHOST  = CL4CL3H
    SAPLOCALHOSTFULL = CL4CL3H
    dbs/db4/opticonnect = 0

    #---------------------------------------------------------------------
    # Number of work processes per service
    #---------------------------------------------------------------------

 F2=Save   F3=Save/Exit   F12=Exit   F15=Services   F16=Repeat find
 F17=Repeat change  F19=Left  F20=Right
```

Figure 10-24 Changing the instance profile

Make a copy of the CL4 instance profile IAS_DVEBMGS00_CL4 to system CL3 (IAS_DVEBMGS00_CL3).

```
                          Work with Object Links

Directory  . . . . :   /usr/sap/IAS/SYS/profile

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
      .                      DIR
      ..                     DIR
      DEFAULT.PFL            STMF
      IAS_DVEBMGS00_CL4 >    STMF
      IAS_DVEBMGS00_CL3 >    STMF
      START_DVEBMGS00_RC >   STMF



                                                                      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 10-25 Copying the instance profile

Make a copy of start profile START_DVEBMGS00_CL4 to system CL3. Adjust the reference in the start profiles for CL3 and CL4 as shown in CL3.

```
 Edit File: /usr/sap/IAS/SYS/profile/START_DVEBMGS00_CL3
 Record :     16   of      48 by   8           Column :    1     72 by  74
 Control :

CMD ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+

    #-------------------------------------------------------------------
    # delete the old sql packages
    #-------------------------------------------------------------------

    Start_Program_01 = CL3 $(DIR_EXECUTABLE)/DLTOLDPKGS IAS

    #-------------------------------------------------------------------
    # start message server
    #-------------------------------------------------------------------

    _MS = $(DIR_EXECUTABLE)/MSG_SERVER
    Start_Program_02 = local $(_MS) pf=$(DIR_PROFILE)/IAS_DVEBMGS00_CL3

    #-------------------------------------------------------------------

 F2=Save   F3=Save/Exit   F12=Exit   F15=Services   F16=Repeat find
 F17=Repeat change  F19=Left  F20=Right
```

Figure 10-26 SAP start profile

Post installation steps

There are some post installation steps that you must also perform. For more information about the about the post installation steps, visit SAP Online Service & Support and search for informational note 568820.

10.6 Operating the SAP environment

This section explains the operation of a switchable IASP-based SAP solution.

10.6.1 Starting the SAP system

Start the SAP system on the productive host CL4 as explained here:

1. Vary on the IASP with the following command:

   ```
   CHGCRGPRI CLUSTER(SAP) CRG(CL4CL3)
   ```

   This automatically varies on the devices cl4cl3 and cl4cl3sec system CL4 as shown in Figure 10-27.

```
                      Work with Configuration Status              RCHASCL4
                                                        12/12/02  15:19:24
Position to  . . . . .              Starting characters

Type options, press Enter.
  1=Vary on   2=Vary off   5=Work with job   8=Work with description
  9=Display mode status    13=Work with APPN status...

Opt  Description       Status               -------------Job--------------
     CL4CL3           AVAILABLE
     CL4CL3SEC        AVAILABLE









                                                                      Bottom
Parameters or command
===>
F3=Exit   F4=Prompt   F12=Cancel   F23=More options   F24=More keys
```

Figure 10-27 WRKCFGSTS to work with ASP devices

2. Enter the following command and add a kernel library to the library list. If you sign on as IASOFR, this is done automatically by JOBD R3IAS400/R3_00.

   ```
   SETASPGRP ASPGRP(CL4CL3)
   ```

3. Enter the following command to bring up your SAP system:

   ```
   STARTSAP SID(IAS)
   ```

10.6.2 Manual switchover to a secondary system

Now switch over the SAP system from production system CL4 to production system CL3. To switch the SAP application from the primary server, you must complete the following steps:

1. On the *production server CL4*, enter the following command:

   ```
   STOPSAP SID(IAS)
   ```

2. Vary off the IASP to ensure that no job on CL4 is still connected with the IASP group. Enter the following command:

   ```
   WRKCFGSTS CFGTYPE(*DEV) CFGD(*ASP)
   ```

   You see the Work with Configuration Status CL4 display as shown in Figure 10-28.

```
                  Work with Configuration Status CL4
                                                              12/10/02  11:38:32
Position to  . . . . .               Starting characters

Type options, press Enter.
  1=Vary on   2=Vary off   5=Work with job   8=Work with description
  9=Display mode status    13=Work with APPN status...

Opt  Description       Status                -------------Job--------------
     CL4CL3            VARIED OFF
     CL4CL3SEC         VARIED OFF









                                                                       Bottom
Parameters or command
===>
F3=Exit   F4=Prompt   F12=Cancel   F23=More options   F24=More keys

Vary off completed for device CL4CL3.                                    +
```

Figure 10-28 WRKCFGSTS: Varying off IASP

3. Log on to the *backup server CL3*.

4. Enter the Change CRG Primary (CHGCRGPRI) command. Then you see the display shown in Figure 10-29.

```
                      Change CRG Primary (CHGCRGPRI)

Type choices, press Enter.

Cluster  . . . . . . . . . . . .   sap           Name
Cluster resource group . . . . .   cl4cl3        Name
Exit program data  . . . . . . .   *SAME

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

Figure 10-29 CHGCRGPRI: Switchover CRG

This command automatically switches off all IASPs of the cluster resource group and brings down all jobs that are connected with these IASPs.

The switchover process takes a few minutes to run. During that time, no one can access the IASP because the cluster resource group CL4CL3 is pending (Figure 10-30).

```
                        Display CRG Information

Cluster  . . . . . . . . . . . . :   SAP
Cluster Resource Group . . . . . :   *LIST
Consistent Information in Cluster:   *YES
Number of Cluster Resource Groups:   1


                      Cluster Resource Group List

Cluster Resource Group   CRG Type      Status                  Primary Node
     CL4CL3              Device        Switchover Pending        RCHASCL4









                                                                      Bottom
Enter=Continue   F12=Cancel   F3=Exit   F1=Help
```

Figure 10-30 DSPCRGINF: Cluster resource group status during the switchover

All ASP devices are automatically available, and then you can access the IASP group again from system CL3 (Figure 10-31).

```
                    Work with Configuration Status CL3
                                                              12/09/02  15:36:07
Position to  . . . . .               Starting characters

Type options, press Enter.
  1=Vary on   2=Vary off   5=Work with job   8=Work with description
  9=Display mode status    13=Work with APPN status...

Opt  Description       Status                -------------Job--------------
     CL4CL3            AVAILABLE
     CL4CL3SEC         AVAILABLE









                                                                       Bottom
Parameters or command
===>
F3=Exit   F4=Prompt   F12=Cancel   F23=More options   F24=More keys
```

Figure 10-31 WRKCFGSTS: Configuration status of IASP devices on backup system CL4

5. Check the status of the CRG using the following OS/400 command:

   ```
   DSPCRGINF CLUSTER(SAP) CRG(CL4CL3)
   ```

   The status is shown in the Display CRG Information displays as shown in Figure 10-32, Figure 10-33, and Figure 10-34. You see that the CRG CL4CL3 of cluster SAP is active again and assigned to the backup server RCHCL3. You can access the entire environment under the switchable IP address 7.7.7.7.

```
                          Display CRG Information

Cluster  . . . . . . . . . . . . :   SAP
Cluster Resource Group . . . . . :   CL4CL3
Reporting Node Identifier  . . . :   RCHASCL3
Consistent Information in Cluster:   *YES

Cluster Resource Group Type  . . . :   Device
Cluster Resource Group Status  . . :   Active
Previous CRG Status  . . . . . . . :   Switchover Pending
Exit Program . . . . . . . . . . . :   *NONE
  Library  . . . . . . . . . . . . :     *NONE
Exit Program Format  . . . . . . . :   *NONE
Exit Program Data  . . . . . . . . :   *NONE


User Profile . . . . . . . . . . . :   *NONE
Text . . . . . . . . . . . . . . . :

                                                                       More...
Enter=Continue   F12=Cancel   F3=Exit   F1=Help
```

Figure 10-32 DSPCRGINF (Display 1 of 3)

```
                          Display CRG Information

Cluster  . . . . . . . . . . . . . :   SAP
Cluster Resource Group . . . . . :   CL4CL3
Reporting Node Identifier  . . . :   RCHASCL3
Consistent Information in Cluster:   *YES

                       Configuration Object Information

Configuration   Object    Device    Device      Vary       Server
 Object Name    Type      Type      Subtype     Online     Ip Address
 CL4CL3         *DEVD     *ASP      Primary     *YES       7.7.7.7
 CL4CL3SEC      *DEVD     *ASP      Secondary   *PRIMARY   *NONE


                                                                    Bottom
Number of Device List Entries  . :   2


Enter=Continue   F12=Cancel   F3=Exit   F1=Help
```

Figure 10-33 DSPCRGINF (Display 2 of 3)

```
                          Display CRG Information

Cluster  . . . . . . . . . . . . . :   SAP
Cluster Resource Group . . . . . :   CL4CL3
Reporting Node Identifier  . . . :   RCHASCL3
Consistent Information in Cluster:   *YES

                        Recovery Domain Information

                                                    Preferred
Node Identifier     Status          Current Node Role     Node Role
   RCHASCL3         Active            Primary             Backup   1
   RCHASCL4         Active            Backup   1          Primary


                                                                    Bottom
Number of Recovery Domain Nodes  :   2

Enter=Continue   F12=Cancel   F3=Exit   F1=Help
```

Figure 10-34 DSPCRGINF (Display 3 of 3)

6. Copy the SAP license file to your backup server as explained in 10.6.4, “SAP license considerations” on page 155.
7. Start SAP on the backup server as explained in 10.6.1, “Starting the SAP system” on page 150.

10.6.3 Logging on to SAP using switchable IP address

Using your SAPGUI (SAP's graphical user interface) configuration, refer to the switchable host name CL4CL3H (Figure 10-16 on page 142). This allows you to log on with the same IP address regardless of whether you are working on production system CL4 or backup system CL3 (Figure 10-35).

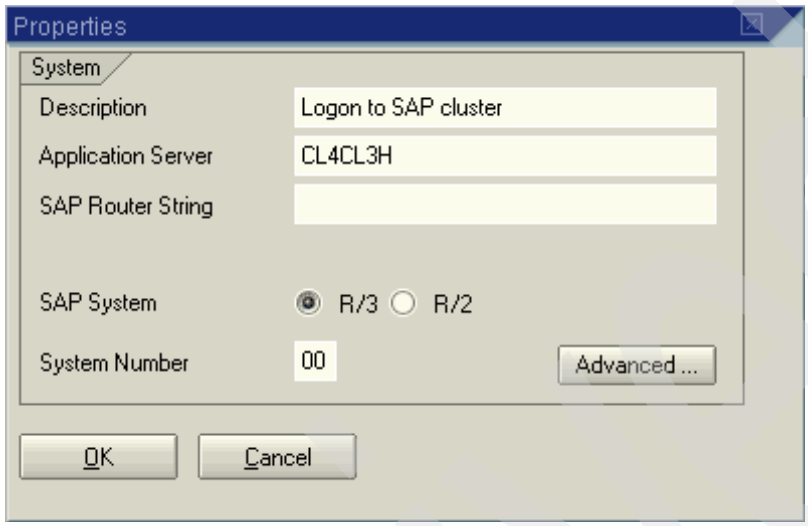


Figure 10-35 Logging on using a switchable host name

10.6.4 SAP license considerations

R3IASDATA/MLICHECK contains the licensing information. You can keep the license information for both the primary and backup servers in this file, which is placed in the switchable IASP CL4CL3. Therefore, the license information is automatically available regardless of whether SAP is started on the primary or backup server.

10.7 SAP IT landscape with IASP considerations

For an SAP IT landscape based on IASP implementation, consider the same criteria as you would for a non-IASP landscape. This chapter only discusses aspects that are specific for SAP running in a cluster-based IASP implementation.

10.7.1 Test system considerations

In all computer-based applications, we recommend that you isolate the production environment from the test and quality assurance environment as much as possible. This is also valid for an SAP solution.

In a cluster-based IASP landscape for SAP, consider testing the switchover functionality as well. Figure 10-36 shows how you can do this using our example with system CL4 for SAP production and system CL3 for SAP backup.

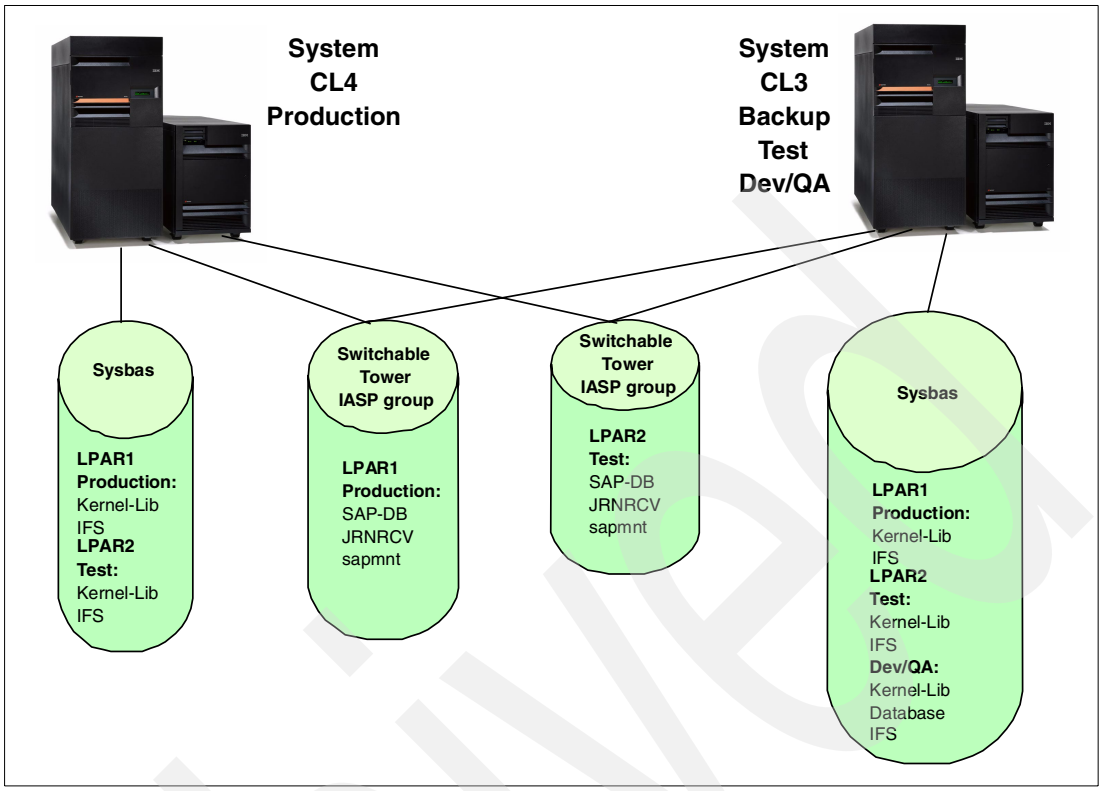


Figure 10-36 SAP IT landscape

Production system CL3 is attached to a *SYSBAS consisting of two logical partitions (LPARs) for the non-switchable part of the production and test system. The backup system is attached to a *SYSBAS. The *SYSBAS contains the entire SAP quality assurance system that cannot be switched and the non-switchable part of the production and test system being installed in separate LPARs. The switchable part both of the SAP production and test system is installed in IASP groups that are placed in separate switchable towers. This allows both the production system and test system to be switched over from one system to the other.

10.7.2 Disaster recovery considerations

The section explains how an SAP IASP implementation can help to recover an SAP production system off site from a disaster. There are some hardware restrictions you should consider in this situation.

Both systems (production and backup) must be connected in an HSL loop with a maximum distance of 750m in total. There are restrictions between the different CPUs and extension towers within the loop. This means that the maximum distance between a production system and a backup system can be 250m.

In Figure 10-37, system CL4 is the production system with a *SYSBAS that contains the non-switchable part of the SAP production system and an IASP group being installed with RAID-1 with bus-level protection on switchable towers. If Room 1 is completely unavailable, you can switch over to system CL3 located in Room 2. Here, a copy of the non-switchable part in the *SYSBAS of system CL3 and the mirrored part of the IASP group are still available.

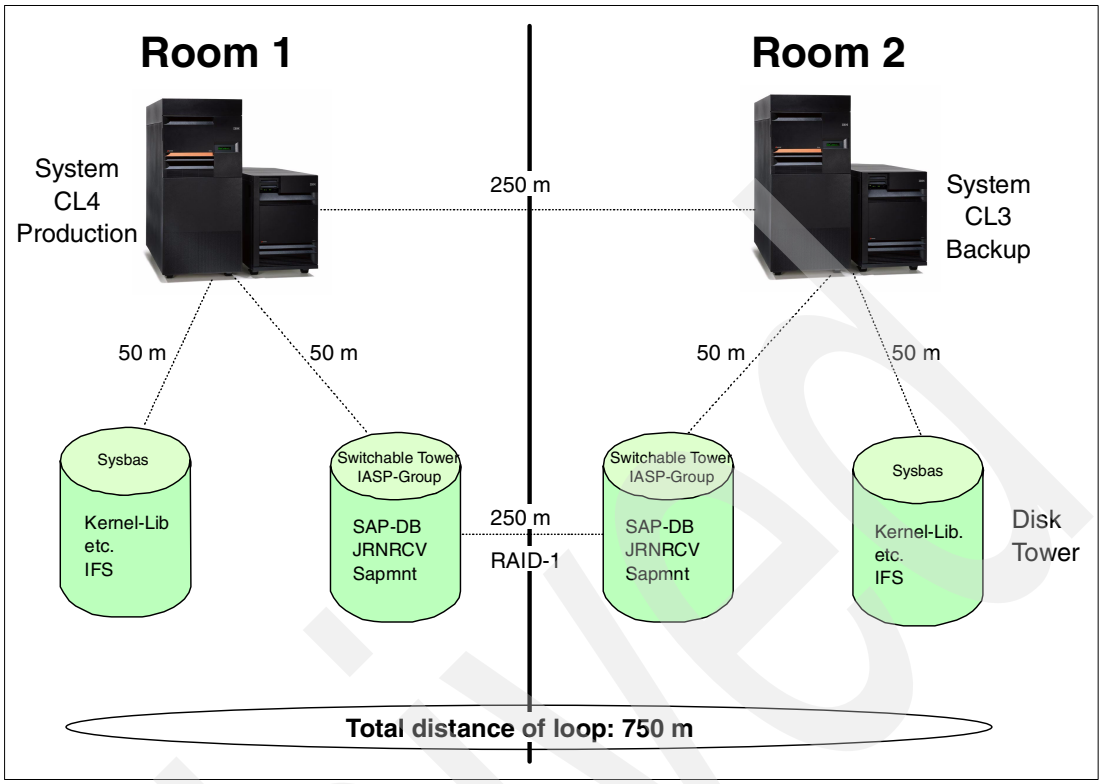


Figure 10-37 IASP landscape for disaster recovery

Save considerations

You need to save and restore your SAP environment regardless of whether you are running SAP in an IASP environment. In both cases, you may use the same techniques. For more information how to save an SAP system, see *Implementing SAP R/3 on OS/400*, SG24-4672.

In addition, you should provide a save copy of the non-switchable part of both systems, which is the *SYSBAS. To perform a save of *SYSBAS, you must complete the following steps:

1. End the subsystem by entering the following command:

   ```
   ENDSBS SBS(*ALL)
   ```

2. Work with the configuration status by entering the following command:

   ```
   WRKCFGSTS CFGTYPE(*DEV) CFGD(*ASP)
   ```

3. Select option 2 to vary off the IASPs.
4. Save only the *SYSBAS system by entering the following Save System command:

   ```
   SAVSYS
   ```

Recovery considerations

If your production system CL4 in Room 1 is unavailable, you can switch over to your backup system CL3 as explained in 10.6.2, “Manual switchover to a secondary system” on page 151. Assuming that all the data from your production system is destroyed, you must perform the following steps to recover your production system CL4:

1. Restore SAVSYS of *SYSBAS only.
2. Perform an initial program load (IPL) to reactivate cluster resource group CL4CL3.

3. Manually start reactivation of all disk units in the switchable tower primary being assigned to production system CL4.
4. Manually start resynchronizing all disk units.
5. Switch over from the backup system CL3 to the production system CL4 as explained in 10.6.2, “Manual switchover to a secondary system” on page 151.

10.7.3 Disk performance considerations

A lack of disk arms can bottleneck the performance of the processor. Two avoid this, a minimum number of disk devices is necessary for optimal performance depending on the iSeries processor level and the SAP workload running on it. The International SAP/IBM Competence Center (ISICC) in Walldorf published guidelines and recommendations depending on the SAP workload and the iSeries processor level. These recommendations refer to an SAP system running in a non-switchable environment, meaning that all components of the SAP system are running in *SYSBAS.

If SAP is running in an IASP environment, one part of the input/output (I/O) workload runs in *SYSBAS (I/Os for operating system, temporary storage, non-switchable part of the SAP directory system). The other part is required in the IASP group (I/Os for database and journal receiver). In an IASP configuration, you must optimize both environments for its own. The minimum requirements for doing this are:

- As a basis for your calculation, use the disk guidelines for a non-switchable environment as published by the ISICC.
- For *SYSBAS, provide 30% of the disk arms as recommended in the ISICC sizing guidelines.
- For the switchable IASP group, provide the amount of disk arms as recommended in the ISICC sizing guideline, minus 20%.

Note: Although these are the minimum requirements, you must optimize each environment individually.

11

Backup and recovery

This chapter describes techniques and strategies for backup and recovery of independent disk pools. When you add an independent disk pool to your system configuration, you need to plan for the backup and recovery of the user data on these devices, because these devices operate differently than the system or basic user auxiliary storage pools (ASPs). These differences mean that you have to carefully plan your backup strategy to assure you have a complete system backup.

This chapter explains what you need to know about saving and restoring objects that are stored in independent ASPs (IASPs). It covers:

- Private IASPs
- Switchable IASPs (planned)
- Switchable IASPs (unplanned)
- Full system recovery
- Backup Recovery and Media Services (BRMS)

11.1 Using native OS/400 save and restore functions

The native OS/400 SAVxxx and RSTxxx commands have been enhanced to provide support for IASPs. Using these commands in your own Control Language (CL) programs to backup the system is relatively straightforward, since you are in control of the environment when the programs are running. In general, these native commands must have access to the name space where the objects to be saved reside. You can achieve this by using the Set Auxiliary Storage Pool Group (SETASPGRP) command or by using the ASPDEV parameter on the SAVxxx and RSTxxx commands.

If you understand the concept of *using* an IASP, you should be able to save or restore specific libraries or objects in that IASP. However, if you are more familiar with using the Save and Restore menus to save or restore your system or its components (for example, *NONSYS, *ALLUSR, or *IBM saves), you need to understand the way in which these are affected by addressability to the IASPs. This is particularly important if you use Option 21 (Entire system).

11.2 Saving IASPs

This section looks at various methods for saving IASPs using native OS/400 SAVxxx commands. You must decide on your backup strategy. Do you want to save the entire system, or do you want to save it in a more granular manner? Why are you using IASPs? Are they switchable or non-switchable? If they are switchable, which node in the cluster will back them up? The answers to these questions can help you determine the best method for saving your IASPs.

11.2.1 Private IASPs

The simplest form of IASP is a private or stand-alone IASP. These allow you to segment or partition the database of a single OS/400 image. In doing this, you create multiple name spaces, one in the system ASP and one in each private IASP. Segmenting the database allows you to have the same library and object names in different IASPs. Clearly, this has implications for backup and recovery.

Depending on how you use private IASPs, you may find it easier to back them up separately, especially if each database has different backup characteristics (for example, frequency of save, retention, or media type). If an IASP is used for archival records, a backup may only be required when the archives are made.

11.3 Saving your entire system

Let's walk through an example to see what Option 21 (Save entire system) actually does and what you must do to ensure it works.

The following example has a system ASP with a library called WINONA, a directory called RAMSEY, and a user ASP with directory AUSTIN. There are three IASPs called MANKATO, SAINTPAUL, and STILLWATER. Each has a library called LOUNGE and a directory called BAR. The lounge only serves ALE, LAGER, and PILSNER in bottles, while the bar serves PALEALE, PORTER, and BOCK tap beers.

This example includes the following actions:

1. Save the entire system. Use Option 21 from the GO SAVE menu to complete the following steps:

   a. Vary on (make available) all IASPs.
   b. Put the system in a restricted state.
   c. Save the entire system.
   d. When save is complete, you must vary off (make unavailable) the IASPs.

   The actual commands that are run for you during these first four steps are:

   ```
   SAVSYS
   SAVLIB SAVLIB(*NONSYS) ACCPTH(*YES) ASPDEV(*SYSBAS)
   SAVDLO DLO(*ALL) FLR(*ANY)
   SAV ((‘/*’) (‘/QSYS.LIB’ *OMIT) (‘/QDLS’ *OMIT)) ASPDEV(*SYSBAS)
   ```

2. Build a list of available IASPs.
3. Perform the set ASP group function using one of the following commands:

   ```
   SAVLIB *NONSYS ASPDEV(*CURASPGRP)
   SAV ASPDEV(*CURASPGRP)
   ```

4. The save process is repeated for each IASP:

   ```
   CHKTAP ENDOPT(*UNLOAD)
   ```

11.3.1 Special considerations on save commands

The ASPDEV parameter allows you to save the IASP without changing your job thread, if you specify the name in this parameter. However, if you are saving to save files, this parameter does not affect the DEVICE parameter of the save commands. You must use the SETSASPGRP command if you are saving to a save file that exists in an IASP. This also allows you to save to a save file that exists in a different IASP than the one you are saving. Rather, the ASPDEV parameter acts as a filter on the SAV command.

For example, there is a file called MICROBREW and a save file called PREMIUM in a library called BEER in an IASP called ROCHESTER. There also is a save file called AMBER in library QGPL.

The following command does *not* work without the SETASPGRP command:

```
SAVOBJ OBJ(MICROBREW) LIB(BEER) DEV(*SAVF) SAVF(BEER/PREMIUM)
```

Even if we add ASPDEV as in the following command, it still does not find the save file. Therefore, the following command does *not* work:

```
SAVOBJ OBJ(MICROBREW) LIB(BEER) DEV(*SAVF) SAVF(BEER/PREMIUM) ASPDEV(ROCHESTER)
```

To save the file MICROBREW to the save file PREMIUM, you must use the SETASPGRP command. The correct sequence for using the save file is shown here:

```
SETASPGRP ASPGRP(ROCHESTER)
SAVOBJ OBJ(MICROBREW) LIB(BEER) DEV(*SAVF) SAVF(BEER/PREMIUM)
```

To save an individual object from the integrated file system (IFS) in an IASP, you must specify the exact path to that object in the OBJ parameter of the SAV command, starting with the IASP name. For example, if you have an IASP named ROCHESTER, with a directory called BONS, and a file exits in this directory called ALE.GIF, you use the following command to save this object:

```
SAV OBJ(‘/rochester/bons/ale.gif’)
```

When you use the SAV command and specify the direct path to an object as in the previous example, you do not have to specify the job thread with the SETASPGRP command or with the ASPDEV parameter.

To save to a save file in the IFS, you must use the SETASPDEV command before the SAV. Otherwise, the command will fail on the DEV parameter. If the save file PREMIUM is in the library BEER, which is in the IASP ROCHESTER, the following commands fail without issuing the SETASPGRP command before the SAV:

```
SAV DEV('/rochester/qsys.lib/beer.lib/premium.file') OBJ(('/rochester/bons/ale.gif'))
```

Specifying the ASPDEV parameter as shown in the following example still results in the "`Library BEER not found`" error message:

```
SAV DEV('/rochester/qsys.lib/beer.lib/premium.file') OBJ(('/rochester/bons/ale.gif'))
ASPDEV(ROCHESTER)
```

To save the entire IFS of an IASP, you must first unmount the file system and then run the following command:

```
SAV OBJ('/dev/*') ASPDEV(ROCHESTER)
```

11.4 Restoring IASPs

Restoring an entire system that uses independent disk pools becomes a more complicated matter. You must restore the system ASP first. Then you must manually create the independent disk pool or pools using iSeries Navigator. This requires knowledge of the original disk pool sizes and names. Also, if you are using the Restore menu to recover user data, you may want to exclude the Restore Authority (RSTAUT) command until you recover all of your IASPs.

The recovery of your system involves this procedure:

1. Install the Licensed Internal Code (LIC).
2. Configure and protect your disk drives
3. Install OS/400 as explained in *iSeries Backup and Recovery*, SC41-5304.
4. If you are recovering your system from an option 21 save, you can perform an option 21 restore at this point using the following commands (prompt for the commands):

   ```
   RSTUSRPRF
   RSTCFG OBJ(*ALL)
   RSTLIB SAVLIB(*NOSYS)
   RSTDLO DLO(*ALL) FLR(*ANY)
   RST (('/*') ('/QSYS.LIB' *OMIT) ('/QDLS' *OMIT))
   ```

   Note: If you are restoring one or more IASPs, you should not perform the RSTAUT command until all data in the IASP is restored.

5. Create your independent storage pools from iSeries Navigator.
6. Restore your independent ASPs with the following commands:

   ```
   RSTLIB SAVLIB(*NONSYS) ASPDEV(yourIASP)
   RST OBJ('/dev/*')
   ```

7. Repeat step 6 for all independent ASPs.
8. Perform the authority restore:

   ```
   RSTAUT USRPRF(*ALL)
   ```

Note: If the RSTAUT command was performed prior to recovering your IASPs, you must use the RSTUSRPRF command first and then the RSTAUT command.

11.4.1 Recovering an IASP after losing the system ASP

When you install LIC on a failed system ASP, the Install Licensed Internal Code display appears as shown in Figure 11-1.

```
                  Install Licensed Internal Code (LIC)

Disk selected to write the Licensed Internal Code to:
     Serial Number    Type      Model     I/O Bus    Controller    Device
       xx-xxxxxxx     xxxx      xxx         x           x           x

Select on of the following:
   1. Restore Licensed Internal Code
   2. Restore Licensed Internal Code and Initialize System
   3. Restore Licensed Internal Code and Recover Configuration
   4. Restore Licensed Internal Code and Restore Disk Unit Data
   5. Restore Licensed Internal Code and Upgrade Load Source
```

Figure 11-1 Install Licensed Internal Code (LIC) display

To install LIC and recover your IASP, select option 3 from the Install Licensed Internal Code menu. When you install the LIC using option 3 from the Install Licensed Internal Code menu, the system:

- Clears disk unit 1, which contains information about how all the other disk units on your system are configured.
- Prepares to delete all data in the system ASP. The system ASP is not actually cleared until you perform the initial program load (IPL) after you install the LIC.

Every disk unit on your system contains information about how it is configured. Dedicated Service Tools (DST) provides an option to recover the disk configuration on your system by using this information. The system reads every disk, assigns it to the correct ASP, and rebuilds the disk configuration information on unit 1. In many cases, you can recover your disk configuration and avoid reloading all your user ASPs. To recover your disk configuration, follow these steps:

1. After you install the LIC, the Disk Configuration Error Report display appears on the A or B mode IPL. If you type 5 in the option column (OPT), you see the Missing Disk Configuration display. From either display, press F3 (Exit) to use DST.
2. Sign on to DST. The system displays the Use Dedicated Service Tools menu. If you are using logical partitions, and you want to recover the primary partition, your LPAR recovery process should cater for the multiple restore operations.
3. From the Use Dedicated Service Tools (DST) menu, select option 4 (Work with disk units).
4. From the Work with Disk Units menu, select option 2 (Work with disk unit recovery).
5. From the Work with Disk Unit Recovery menu, select option 5 (Recover disk configuration).
6. Check the configuration of disk units on the display. The display shows the disk units that are assigned to each user ASP and to the system ASP (ASP 1). The warning message on the display means that the system clears all data on disk units in the system ASP.

If this configuration *is not correct*, contact a service representative or software support for assistance. Do not proceed further without getting help.

If the configuration that is shown *is correct*, press F10 to confirm the configuration. The system builds the configuration information and returns to the DST menu.

8. Press F12 to cancel the DST menu. You see the IPL or Install the System menu.

At this point, continue with the normal recovery of your system ASP, including OS/400, user profiles, configuration, library, documents, and IFS data. You may want to wait until you recover your IASP before you run the RSTAUT command.

After the system ASP is recovered, perform the following steps using iSeries Navigator:

1. In iSeries Navigator, expand **My Connections** (or your active environment).
2. Expand any iSeries server.
3. Expand **Configuration and Service**.
4. Expand **Hardware**.
5. Expand **Disk Units**.
6. If the Service Tools Signon window opens, sign on to service tools.
7. Select **Disk Pools**.
8. Right-click the disk pool. If iSeries Navigator detects a problem, then **Recover Configuration** or **Recover Unknown Configuration Source** appears in the list. If you see either of these options, select it to continue.
9. Follow the instructions on the window that opens.

Note: This is only an example recovery. Your own recovery may differ significantly. You should test your recovery before moving to production.

11.4.2 Saving and restoring Linux network storage space (NWSSTG) in an IASP

When a NWSSTG is created on an IASP, it still creates its pointers under the QFPNSSTG directory in the system ASP. This means to save and restore a network storage space that was created in an IASP, you must save the information under QFPNSSTG in the system ASP. You must also save the NWSSTG named under the /dev/IASPname directory. The following command is an example for creating a Linux storage space:

```
CRTNWSSTG NWSSTG(LINUXSTG) NWSSIZE(3000) FORMAT(*OPEN) ASP(40)
```

This command creates a NWSSTG called LINUXSTG in ASP40, which is called ROCHESTER. The actual storage space resides in /dev/rochester/linuxstg.udfs. It also creates an entry in the /qfpnsstg/linuxstg/qfpcontrol along with a /mount directory under /qfpnsstg/linuxstg.

To use a NWSSTG, you must also create a network storage description to link to the storage space. The following commands are examples for creating a Linux network storage description and add the link to the NWSSTG:

```
CRTNWSD NWSD(LINUXSVR) RSRCNAME(*NONE) TYPE(*GUEST) PARTITION(LINUX) +
ADDNWSSTGL NWSSTG(LINUXSTG) NWSD(LINUXSVR)
```

For more information about creating partitions on Linux on iSeries, see *Linux on the IBM @server iSeries Server: An Implementation Guide*, SG24-6232.

If the network storage space was created on a switchable IASP, you can save the pointers and the network server description and restore it to the other system in the cluster. By switching the IASP, Linux can be usable on the other system. The following commands save the objects needed for Linux from System A, place them in a save file in the switchable IASP, restore the objects to System B, and then allow Linux to be active in the partition on system B:

```
CRTLIB LIB(MYLIB) ASP(*ASPDEV) ASPDEV(ROCHESTER)
SETASPGRP ASPGRP(ROCHESTER)
CRTSAVF FILE(MYLIB/SAVEFILE1)
CRTSAVF FILE(MYLIB/SAVEFILE2)
SAV DEV('/ROCHESTER/QSYS.LIB/MYLIB.LIB/SAVEFILE1.FILE') OBJ(('/QFPNWSSTG/LINUXSTG'))
SAVCFG DEV(*SAVF) SAVF(MYLIB/SAVEFILE)
SETASPGRP ASPGRP(*NONE)
```

Note that System B must already have a Linux partition configured, but the network server space and the network server description are restored from System A.

After switching the IASP to System B, you can use the following commands to make Linux usable on System B (assuming that the partition already exits):

```
SETASPGRP ASPGRP(ROCHESTER)
RST DEV('/ROCHESTER/QSYS.LIB/MYLIB.LIB/SAVEFILE1.FILE') OBJ(('/QFPNWSSTG/LINUXSTG'))
RSTCFG OBJ(LINUXSVR) DEV(*SAVF) OBJTYPE(*NWSD) SAVF(MYLIB/SAVFILE2)
SETASPGRP ASPGRP(*NONE)
```

It is important to restore the network server description last, since this automatically links the network server description to the network server storage space. At this point, varying on the network server storage description should bring up the Linux partition.

11.5 Using BRMS with IASPs

Auxiliary disk pool devices are specified in OS/400 operations by the 10-character device description name rather than by specifying an identifying number. BRMS also uses the name when targeting saves or restores to ASP devices. Refer to the system documentation if you are not familiar with ASP device operations or terminology.

11.5.1 Backing up ASP devices

The Edit Backup Control Group Entries display has been updated as shown in Figure 11-2 to include a new Auxiliary storage pool device field. This field does not appear on some backup items entries. Typically this occurs for backup items that cannot reside on ASP devices. The Auxiliary storage pool device prompt is automatically filled in for entries of your existing backup control groups to reflect the scope of the save across ASP devices. These default values should not affect your current backup strategy and should be consistent with what is saved by the control group in V5R1.

The *SYSBAS value on the *ALLUSR backup item saves all user libraries on the system (1) and any basic user (2 to 32) ASPs.

The *ALLAVL value for the *LINK backup items saves the links on the system (1) and any basic user (2 to 32) ASPs, as well as the links on all available ASP devices.

Note: When saving the directory and files, you should unmount any mounted user-defined file systems (UDFSs) before the save to assure the objects in the mounted over directories are saved. UDFSs are automatically unmounted on ASP devices when the system is in restricted state. UDFSs on the system or basic user ASPs must be explicitly unmounted. You must remount any unmounted UDFSs after the save.

```
                  Edit Backup Control Group Entries              AAAAS400

Group . . . . ..... . : SYSTEM
Default activity  . . . . . FIIIIII
Text  . . . . . . . . . . . Backs up the entire system

Type information, press Enter.

                        Auxiliary   Weekly    Retain  Save      SWA
      Backup      List  Storage     Activity  Object  While     Message
Seq   Items       Type  Pool Device SMTWTFS   Detail  Active    Queue

____  __________  ____  _________
__10  *EXIT_____                  *******
__20  *SAVSYS___  ____            FIIIIII
__30  *IBM______  ____            FIIIIII   *NO_    *NO_____
__40  *ALLUSR___  ____  *SYSBAS   FIIIIII   *ERR    *NO_____
__50  *ALLDLO___                  FIIIIII   *NO_    *NO_____
__60  *LINK_____  ____  *ALLAVL   FIIIIII   *YES    *NO_____
__70  *EXIT_____                  *******

                                                                  Bottom
F3=Exit                  F5=Refresh            F10=Change item
F11=Display exits        F12=Cancel            F24=More keys
```

Figure 11-2 Edit Backup Control Group Entries (Display 1)

Restrictions of the operating system prevent you from attaching a primary ASP to the current job if the system is in restricted state. The only way you can save the libraries on ASP devices while in restricted state is to specify the device name in the Auxiliary storage pool device prompt for the backup item.

Assume that the system is configured with an ASP group consisting of a device named PAYROLL serving as the primary ASP device and ACCOUNTS serving as the secondary ASP device. Figure 11-3 shows that to perform a save of all user libraries, three separate *ALLUSR backup items are required. One *ALLUSR with *SYSBAS is specified to save the libraries on the system and basic user ASPs, and one *ALLUSR for each of the PAYROLL and ACCOUNTS ASP devices.

You may consider using this approach if the ASPs devices are configured as private (non-switchable) disk pools. Notice that the *ALLUSR backup items directed to the PAYROLL and ACCOUNTS ASP devices is after the *LINK backup item. This was done to optimize the recovery because objects saved from ASP devices are always recovered after the objects on the system and basic user ASPs are recovered.

```
                  Edit Backup Control Group Entries              AAAAS400

Group . . . . . . . . . . . : SYSTEM
Default activity  . . . . . . FIIIIII
Text  . . . . . . . . . . . . Backs up the entire system

Type information, press Enter.

                        Auxiliary   Weekly    Retain  Save      SWA
      Backup      List  Storage     Activity  Object  While     Message
Seq   Items       Type  Pool Device SMTWTFS   Detail  Active    Queue
____  __________  ____  __________
__10  *EXIT_____                    *******
__20  *SAVSYS___  ____              FIIIIII
__30  *IBM______  ____              FIIIIII   *NO_    *NO_____
__40  *ALLUSR___  ____  *SYSBAS___  FIIIIII   *YES    *NO_____
__50  *ALLDLO___                    FIIIIII   *YES    *NO_____
__60  *LINK_____  ____  *ALLAVL___  FIIIIII   *YES    *NO_____
__70  *ALLUSR___  ____  PAYROLL___  FIIIIII   *YES    *NO_____
__80  *ALLUSR___  ____  ACCOUNTS__  FIIIIII   *YES    *NO_____
__90  *EXIT_____                    *******

                                                                 Bottom
F3=Exit                    F5=Refresh              F10=Change item
F11=Display exits          F12=Cancel              F24=More keys
```

Figure 11-3 Edited Backup Control Group Entries (Display 2)

It may be that the ASP group is configured for use in a clustered environment where the group is automatically switched to an alternate system by the cluster management software when it detects the primary system is no longer operational. It may be useful in this environment to set up a backup strategy that consists of two backup control groups, one to backup the base system and one to backup the switchable ASP group.

Figure 11-4 shows the backup control group entries that would be used on primary and alternate systems to save only the objects on the system and basic user ASPs. The *SYSBAS special value used for the Auxiliary storage pool device parameter of the backup items restricts the scope of the saves to the system and basic user ASPs. The exception is the *SAVSYS backup item that saves the private authorities for all objects on all available ASP devices.

```
                  Edit Backup Control Group Entries              AAAAS400

Group . . . . . . . . . . . : SYSTEM
Default activity  . . . . . FIIIIII
Text  . . . . . . . . . . . Backs up entire system but not switched ASPs

Type information, press Enter.

                       Auxiliary   Weekly    Retain  Save      SWA
      Backup      List  Storage     Activity  Object  While     Message
Seq   Items       Type  Pool Device SMTWTFS   Detail  Active    Queue
____  __________  ____  __________
__10  *EXIT                       *******
__20  *SAVSYS     ____            FIIIIII
__30  *IBM        ____            FIIIIII   *NO_    *NO_____
__40  *ALLUSR     ____  *SYSBAS___  FIIIIII   *YES    *NO_____
__50  *ALLDLO                     FIIIIII   *NO_    *NO_____
__60  *LINK       ____  *SYSBAS___  FIIIIII   *YES    *NO_____
__70  *EXIT                       *******

                                                                  Bottom
F3=Exit                  F5=Refresh             F10=Change item
F11=Display exits        F12=Cancel             F24=More keys
```

Figure 11-4 Edit Backup Control Group Entries (Display 3)

The entries shown in Figure 11-5 are specified in a second backup control group that is scheduled to run on the primary system after the SYSTEM backup control group is run. It is not scheduled to run on the alternate system unless switchover of these ASP devices occurs. As you can see from the entries, only the user libraries and links on the ASP devices are saved by this backup control group.

Note: When using a switched ASP device, you need to consider any private authorities that may be defined for objects of these types of devices. All user profiles and all private authorities for objects on ASP devices are saved with the *SAVSYS or *SAVSECDTA backup items. You need to consider how to restore these private authorities if this authority changes while the device is switched to an alternate system. You may mean that as part of the recovery, where you restore the associated user profiles from the alternate system to the primary system before you restore any objects to the switched ASP device.

When running multiple backup control groups, you need to consider the order in which these control groups run and which control group manages the restart of subsystems to return from restricted state.

Another not so obvious implication of switched ASP devices is when an incremental save is specified for the Weekly Activity. When the ASP device switches to the alternate system, the alternate system has no history of the switched objects. Therefore, the first save of the objects on the switched ASP devices is a full save rather than an incremental save.

```
                    Edit Backup Control Group Entries              AAAAS400

Group . . . . . . . . . . . : ASPS
Default activity  . . . . . FIIIIII
Text  . . . . . . . . . . . Backs up the switched ASPs

Type information, press Enter.

                         Auxiliary   Weekly    Retain  Save      SWA
      Backup      List   Storage     Activity  Object  While     Message
Seq   Items       Type   Pool Device SMTWTFS   Detail  Active    Queue

____  __________  ____  __________
__10  *EXIT_____                    *******
__20  *ALLUSR___  ____  PAYROLL___  FIIIIII   *YES    *NO_____
__30  *ALLUSR___  ____  ACCOUNTS__  FIIIIII   *YES    *NO_____
__40  *LINK_____  ____  PAYROLL___  FIIIIII   *YES    *NO_____
__50  *LINK_____  ____  ACCOUNTS__  FIIIIII   *YES    *NO_____
__60  *EXIT_____                    *******
                                                              Bottom
F3=Exit                   F5=Refresh              F10=Change item
F11=Display exits         F12=Cancel              F24=More keys
```

Figure 11-5 Edit Backup Control Group Entries (Display 4)

Specifying the individual ASP devices in the previous example can be cumbersome especially if you have a large number of secondary ASP devices in the group. If the objects on these ASP devices can be saved using save while active, there is no need to perform the save of these objects in restricted state. And, you can use the *SETASPGRP backup item to attach the device to the job. To do so, add a *SETASPGRP backup item and specify the primary ASP device name in the Auxiliary storage pool device prompt for this backup item. Then use the *CURASPGRP special value for the Auxiliary storage pool device prompt on all subsequent backup items to be backed up from that primary ASP device and any of its associated secondary ASP devices. This is shown in the Edit Backup Control Group Entries display in Figure 11-6.

Notes:

- The *SETASPGRP backup item cannot run while in a restricted state.
- When more than one *SETASPGRP is used as backup items, any previously attached primary ASP is detached from the job when the subsequent *SETASPGRP backup item is processed.
- To explicitly remove an attached primary ASP from the job, use *SETASPGRP with *SYSBAS for the Auxiliary storage pool device prompt.

```
                    Edit Backup Control Group Entries              AAAAS400

Group . . . . . . . . . . . : ASPS
Default activity  . . . . . FIIIIII
Text  . . . . . . . . . . . Backs up the switched ASPs using save-while-active

Type information, press Enter.

                         Auxiliary   Weekly    Retain  Save      SWA
      Backup      List   Storage     Activity  Object  While     Message
Seq   Items       Type   Pool Device SMTWTFS   Detail  Active    Queue
____  __________  ____   __________  *******
__10  *EXIT_____                     *******
__20  *SETASPGRP  ____   PAYROLL___  FIIIIII
__30  *ALLUSR___  ____   *CURASPGRP  FIIIIII   *YES    *YES____  *LIB_____
__40  *LINK_____  ____   *CURASPGRP  FIIIIII   *YES    *YES____  *LIB_____
__50  *SETASPGRP  ____   *SYSBAS___  FIIIIII
__60  *EXIT_____                     *******

                                                                    Bottom
F3=Exit                    F5=Refresh              F10=Change item
F11=Display exits          F12=Cancel              F24=More keys
```

Figure 11-6 Edit Backup Control Group Entries (Display 5)

11.5.2 Recovery of ASP devices

When your system recovery includes objects saved from ASPs devices, additional steps are added to the System Recovery Report to aid you in recovering your system. These steps appear after the words *Step: Perform IPL* because iSeries Navigator is used to reconfigure the ASP devices. These steps do not appear if you are recovering by ASP. The assumption is that the ASP you are recovering is already configured. The step shown in Figure 11-7 provides the guidance to configure the ASP devices during system recovery.

```
********************************************************************************
STEP: Configure auxiliary storage pool devices

   Start date/time _________________  Stop date/time _________________  Duration

         The recovery includes objects saved from auxiliary storage pool
         devices. The auxiliary storage pool devices must be deleted and
         re-configured before recovery can continue.

         Use the following command to view the restored auxiliary storage
         pool device descriptions:

            WRKDEVD DEVD(*ASP)

         Use option 4=Delete to remove the device descriptions from the system.

         Use the "Display ASP Information" report (QP1AASP) to review the
         names of the auxiliary storage pool devices of the saved system.

         --- Attention ------------------------------------------------------
         If the auxiliary storage pool devices are intended to be used as
         switched auxiliary storage pools in a clustered environment, review
         the documentation provided with your cluster management product to
         identify any pre-configuration steps that may be required prior to
         configuring these auxiliary storage pool devices.
         --------------------------------------------------------------------

         Re-configure the auxiliary storage pool devices using iSeries
         Operations Navigator referring to the online disk unit help.
********************************************************************************
```

Figure 11-7 Configuring ASP devices verification

The step shown in Figure 11-8 provides the guidance to verify the names of the configured ASP devices to the names of the saved ASP devices. When you configure the ASP devices using iSeries Navigator, these devices may be assigned the same ASP number depending on the number of disk resources and the order in which they are configured. Since the ASP number may change, it is important to keep the name the same to allow for automated recovery.

If you are restoring objects to a different ASP device than the one from which it was saved, you need to change the default recovery options to specify the ASP where the objects are to be restored.

```
********************************************************************************
STEP: Verify auxiliary storage pool device names

   Start date/time ___________________  Stop date/time ___________________  Durati

         Use the following command to view the configured auxiliary storage
         pool device descriptions:

            WRKDEVD DEVD(*ASP)

         Verify the configured auxiliary storage pool device names match the
         names of the auxiliary storage pool devices of the saved system.

         Use option 8=Work with status to verify each auxiliary storage pool
         device status is AVAILABLE.
********************************************************************************
```

Figure 11-8 Verifying ASP device names

BRMS saves the history information for ASP devices by the ASP device name. This device name is displayed adjacent to the saved item name as shown in the sample report in Figure 11-9. The step shown is included in the report if libraries are saved from ASP devices. You may also see a similar section for recovery of any directories and files that may be saved from ASP devices.

When recovering libraries on ASP devices and recovering these to the same system and the same ASP device, you need to consider these points:

- When you vary on an ASP device, library QSYS200*nnn* (where *nnn* is the ASP number) is created on the ASP device as well as several objects.
- Recover the saved QSYS200nnn library to the ASP device before you recover the other libraries on that ASP device.
- When you recover the saved QSYS200nnn library, use F9 (Recovery defaults) from the Select Recovery Items display. You must also set the Allow object differences prompt on the Restore Command Defaults panel to *ALL.
- When you recover the remaining libraries, use F9 (Recovery defaults) from the Select Recovery Items display. You must also set the Allow object differences prompt on the Restore Command Defaults panel to *FILELVL.

```
********************************************************************************
STEP: Recover additional user libraries

   Start date/time _________________  Stop date/time _________________  Duration

You should restore the current version of your user libraries saved
         from auxiliary storage pool devices or saved to TSM servers.

         If you are performing a complete system restore, run the following
         command to continue:

            STRRCYBRM OPTION(*RESUME)

         Otherwise, run the following command(s):

            STRRCYBRM OPTION(*ALLUSR) ACTION(*RESTORE) USEADSM(*YES)
            STRRCYBRM OPTION(*ASP) ACTION(*RESTORE) ASP(auxiliary-storage-pool-n

         Select and recover the following saved item(s) on the "Select Recovery
Items" display using the specified volume(s).

         --- Attention --------------------------------------------------------
         If you have logical files whose based-on physical files are in a
         different library, you must restore all based-on physical files
before you can restore the logical file.
         ----------------------------------------------------------------------

         --- Attention --------------------------------------------------------
         If you use journaling, the libraries containing the journals must be
restored before restoring the journaled files.
         ----------------------------------------------------------------------
   Saved            ----- ASP ------  Save     Save                  Sequence  C
   Item       Type  Name      Number  Date     Time    Objects Omit  Number    G
   ---------- ----- ---------- ----- -------- -------- ------- ----- --------- -
__ EMPLOYEES  *FULL PAYROLL    00033  8/22/02 13:09:52       6               1 S
__ AUG2002    *FULL ACCOUNTS   00034  8/22/02 13:21:57       6               2 S
********************************************************************************
```

Figure 11-9 Recovering additional user libraries

11.6 Recovering an independent disk pool

If you experience problems accessing an independent disk pool or making it available, there may be a problem with the disk pool. This section explains a couple of possible problems.

First the configuration source may be corrupted. When corruption occurs, the independent disk pool appears to have no disk units in it. If this occurs, you can select to recover the configuration information on the configuration source. Recovering the configuration attempts to determine the original configuration and recover it. During this process, the dependent disk pool may need to be cleared, destroying all data on the disk units in the pool. If the disk pool must be cleared, a message appears that warns you of this and allows you to cancel the recovery.

Also, the mirrored disk unit of the configuration source may be damaged. When this happens, the mirrored configuration source becomes unknown. The disk pool is unavailable, and you must recover the configuration information of an unknown configuration source before you make it available. You should only attempt to recover the state of the unknown configuration source when you know its mirrored disk unit was active before the failures that caused the state to become unknown.

To attempt to recover an independent disk pool, follow these steps:

1. In iSeries Navigator, expand **My Connections** (or your active environment).
2. Expand any iSeries server.
3. Expand **Configuration and Service**.
4. Expand **Hardware**.
5. Expand **Disk Units**.
6. Expand **Disk Pools**.
7. Right-click the problematic disk pool. If iSeries Navigator detects one of the problems that was previously mentioned, then **Recover Configuration** or **Recover Unknown Configuration Source** appears in the list. If you see either option, select it to continue.
8. Follow the instructions on the window that opens.

A

Prerequisite steps

Before you can create any independent auxiliary storage pools (IASPs), you must perform some prerequisite steps within Operations Navigator. 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.

Configuring the Service Tools adapter on the iSeries server

A service table entry is required to use the disk functions of Operations Navigator. This section presents the steps to add a service table entry and submit a job to reclaim Transmission Control Protocol (TCP) on the iSeries server that is to be used with the IASP.

1. Enter the Add Service Table Entry (ADDSRVTBE) command.
2. Fill in the parameters as shown in Figure A-1 to add a service table entry named *'as-sts'* Press Enter to continue.

> **Important:** Enter the value for the Service and Protocol parameters in *lowercase letters* and *enclosed in single quotation marks*. If you do not follow these rules, and Additional Parameters is selected, the Protocol value becomes uppercase letters, which causes an error to occur.

```
                    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 A-1 Adding a service table entry on the iSeries server

3. On an OS/400 command line, enter the Submit Job (SBMJOB) command.
4. Complete the information on the Submit Job display as shown in Figure A-2. Press Enter to continue. This step submits a job to reclaim TCP.

```
                              Submit Job (SBMJOB)

Type choices, press Enter.

Command to run . . . . . .   call rcltcp_____________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________...
Job name . . . . . .         *JOBD        Name, *JOBD
Job description  . . . . .   *USRPRF      Name, *USRPRF
  Library  . . . . . .       _______      Name, *LIBL, *CURLIB
Job queue  . . . . . .       *JOBD        Name, *JOBD
  Library  . . . . . .       _______      Name, *LIBL, *CURLIB
Job priority (on JOBQ) . . . .  *JOBD     1-9, *JOBD
Output priority (on OUTQ)  . . .  *JOBD   1-9, *JOBD
Print device . . . . . .     *CURRENT     Name, *CURRENT, *USRPRF...

                                                                    More...
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

Figure A-2 Submitting a job to reclaim TCP on the iSeries server

Setting up Management Central

Configure the Management Central function in Operations Navigator. This section takes you through the necessary steps:

1. Start Operations Navigator.
2. Select the system name, right-click, and select **Application Administration** (Figure A-3).

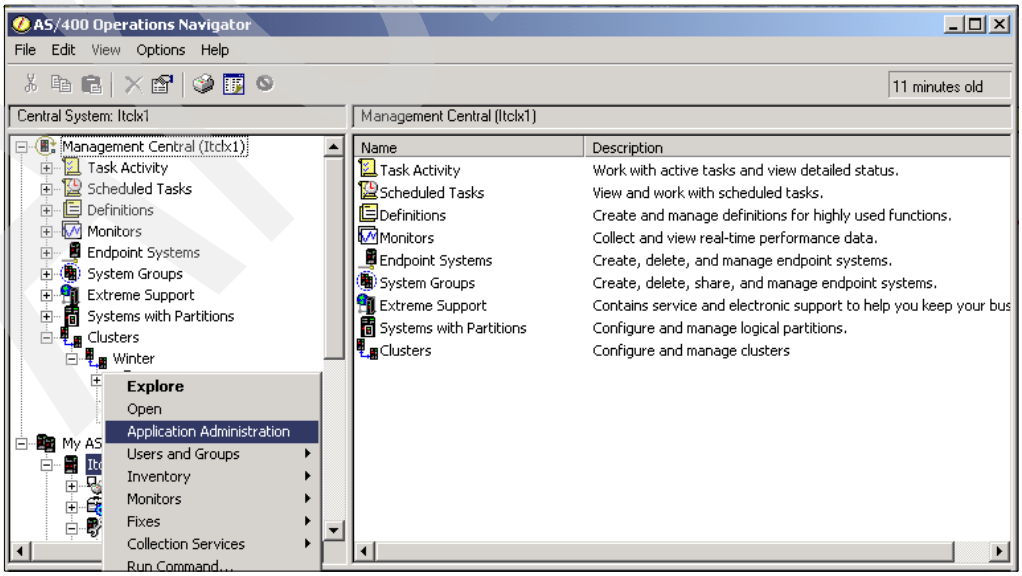


Figure A-3 Access Application Administration in Operations Navigator

3. If the Applications window (Figure A-4) opens, click **OK** to continue.

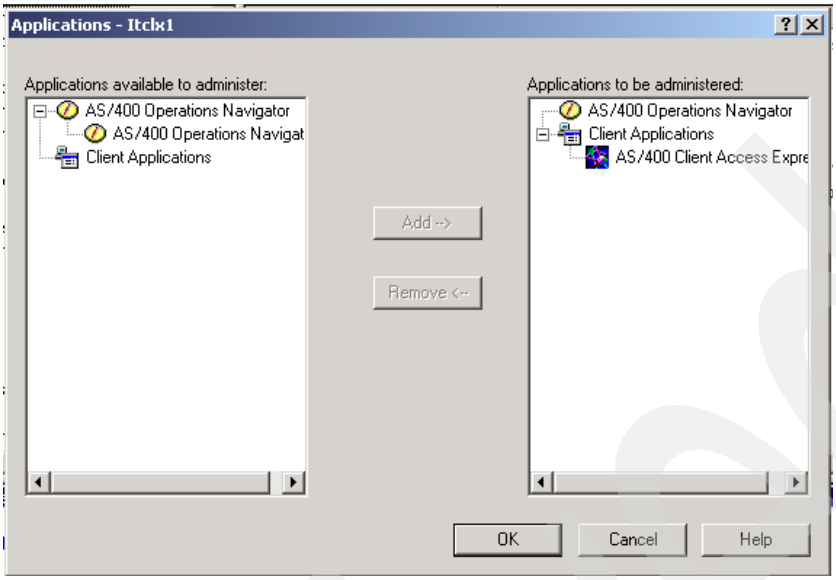


Figure A-4 Application window

4. On the Application Administration window (Figure A-5), click the **Host Applications** tab.

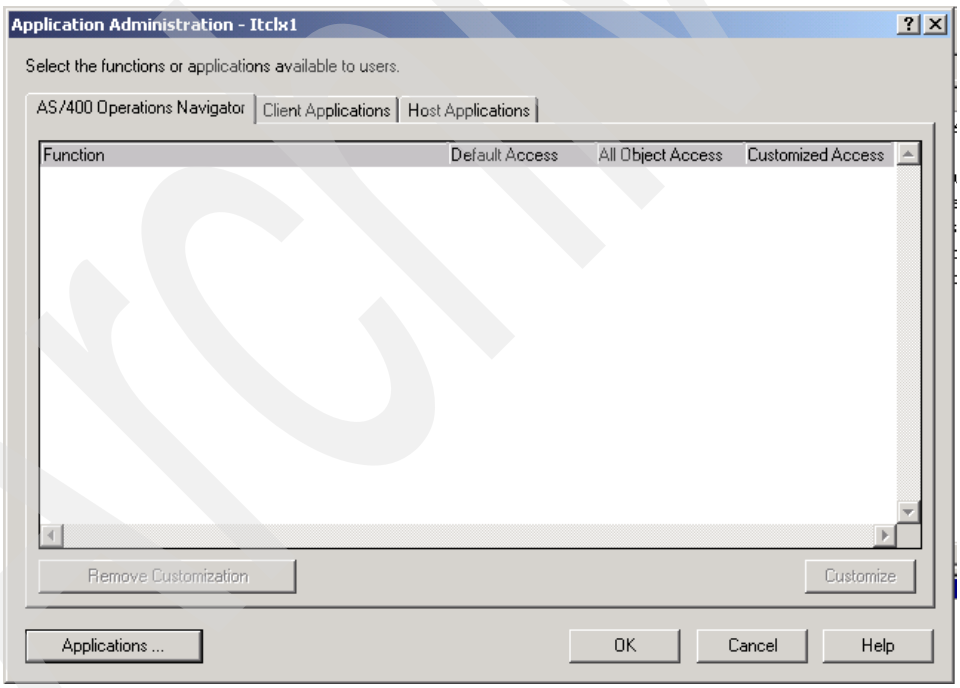


Figure A-5 Application Administration window

5. On the Host Applications page (Figure A-6), expand the **Operating System/400** and **Service** folders until you see Disk units. For Disk units, select the **Default Access** and **All Object Access** check boxes. Click **OK**.

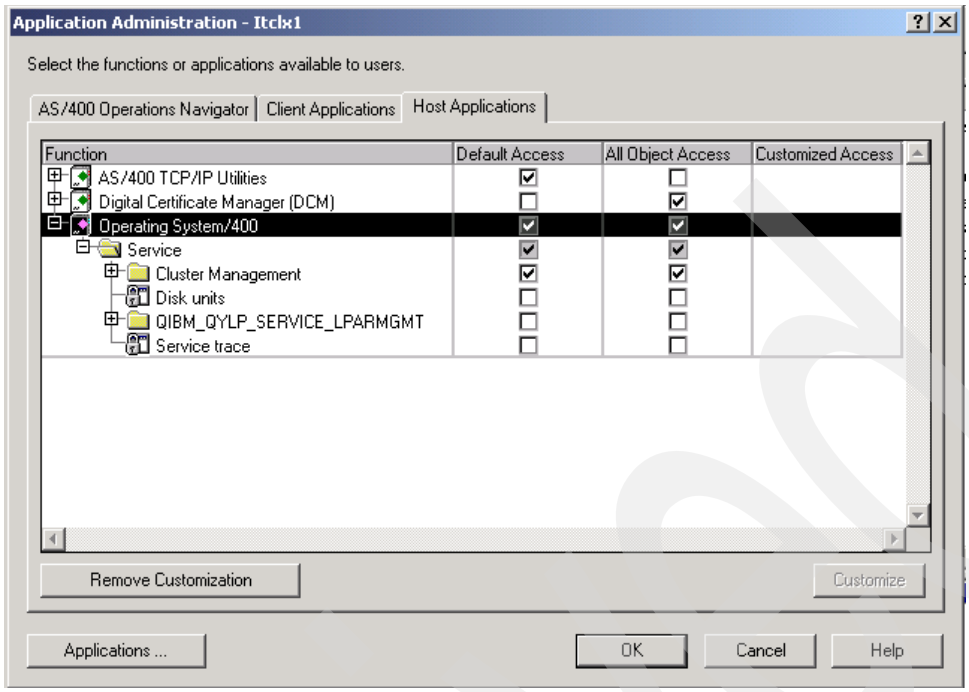


Figure A-6 Enabling disk unit access on the Host Applications tab

Now you can use the Disk Units function from within Operations Navigator.

B

Planning checklists

Use the checklists in this appendix for planning the various pieces of the independent disk pool puzzle. Three lists are presented here:

- Independent disk pool checklist
- Disk planning worksheet
- PCI card placement planning worksheet

Independent disk pool checklist

Table B-1 Independent disk pool checklist

Item #	Check	Description	Where to find this information
1.0	____	**Business needs**	
	____	Quantification	
	____	Accessibility	
2.0	____	**Performance requirements**	
	____	Response time	
	____	Throughput	
	____	Uptime	
	____	Throughput	
	____	Outage allowances	
3.0	____	**Software licensing requirements**	
	____	V5R1	
	____	V5R2	
	____	5722-TC1	
	____	Option 41	
	____	Option 22	
	____	Option 23	
4.0	____	**Application integration**	
	____	Unsupported objects in system ASP	
	____	Supported objects in IASP	
	____	Application objects common to all IASPs	
	____	Other application specific considerations	
5.0	____	**Security considerations**	
	____	User profiles	
	____	Authorization lists	
6.0	____	**Capacity planning**	
	____	Clustering requirements	
	____	Disk drive(s) required	
	____	IOAs required	
	____	IOPs required	
	____	HSL cable(s) required	
	____	SPCN cable(s) required	
	____	Additional racks required	

Item #	Check	Description	Where to find this information
7.0	____	**Hardware configuration**	
	____	Additional rack placement	
	____	Disk placement	
	____	Card placement	
	____	HSL cable placement	
	____	SPCN cable placement	
8.0	____	**Aesthetics**	
	____	Primary rack placement	
	____	Secondary rack placement	
	____	Additional rack placement	
	____	Tape drive placement	
	____	Console placement	
9.0	____	**Physical planning requirements**	
	____	Floor space requirements	
	____	Air conditioning requirements	
	____	Additional power requirements	
	____	Cable lengths	
	____	Power - 110 Volts	
	____	Plug type	
	____	Power - 220 Volts	
	____	Plug type	
	____	HSL	
	____	Communication	
	____	Twinax	
	____	SPCN	

Disk planning worksheet

Table B-2 Disk planning worksheet

Frame ID	IOA RSC name	Slot	Disk type or feature	Density	Resource name	Protection	ASP #	ASP name
		D31						
		D36						
		D41						
		D32						
		D37						
		D42						
		D33						
		D38						
		D43						
		D34						
		D39						
		D44						
		D35						
		D40						
		D45						
		D01						
		D11						
		D21						
		D02						
		D12						
		D22						
		D03						
		D13						
		D23						
		D04						
		D14						
		D24						
		D05						
		D15						
		D25						
		D06						

Frame ID	IOA RSC name	Slot	Disk type or feature	Density	Resource name	Protection	ASP #	ASP name
		D16						
		D26						
		D07						
		D17						
		D27						
		D08						
		D18						
		D28						
		D09						
		D19						
		D29						
		D10						
		D20						
		D30						

PCI card placement planning worksheet

Table B-3 PCI card placement planning worksheet

Frame ID	Slot or card position	Type or feature	Resource name	Serial #	Description
	C01				
	C02				
	C03				
	C04				
	C05				
	C06				
	C07				
	C08				Reserved for HSL connection
	C09				
	C10				
	C11				
	C12				
	C13				
	C14				
	C15				

C

Disk unit selection criteria

When you add disk units to a disk pool, you can select these units from a list. The list includes a column for the disk unit rank. The number assigned for disk unit rank indicates which disk units are best for a particular disk pool. Use the categories and ranks in Table C-1 to determine which disk units to add or how to improve a disk unit ranking. In the following descriptions, *switchable entity* is the term used to describe the switchable tower or switchable input/output (I/O) processor in the case of switching between logical partitions.

Table C-1 Disk unit rank and description

Rank	Description
Best disk ranks	
2	Best disk unit for non-switchable disk pool
53	Best disk unit for switchable disk pool
Valid disk ranks	
102	Valid disk unit for non-switchable disk pool, but disk unit could be made switchable
103	Valid disk unit for non-switchable disk pool because this disk unit's switchable entity already has a disk unit configured in the specified disk pool
104	Valid disk unit for non-switchable disk pool because this disk unit's switchable entity already has a disk unit configured in a non-switchable disk pool
106	Valid disk unit for non-switchable disk pool because the disk unit's switchable entity contains disk units that are all nonconfigured
121	Valid disk unit for unknown independent disk pool
151	Valid disk unit for switchable disk pool because the disk unit's switchable entity already has a disk unit configured in the same cluster resource group
153	Valid disk unit for switchable disk pool because this disk unit's switchable entity contains disk units that are all non configured
157	Valid disk unit for switchable disk pool, but this disk unit's tower could be blocked by another tower

Rank	Description
Warning disk ranks	
202	Allowable disk unit for non-switchable disk pool, but switchable entity might block switchable disk pool
204	Allowable disk unit for non-switchable disk pool, but switchable entity has unknown independent disk pool
221	Allowable disk unit for switchable disk pool, but switchable entity has unknown independent disk pool
223	Allowable disk unit for switchable disk pool, but disk unit may cross power domains
225	Allowable disk unit for switchable disk pool, but cannot determine if all nodes in the cluster resource group can access the disk unit
231	Cannot determine if independent disk pool is switchable because clustering is not active
Invalid disk ranks	
302	Invalid disk unit for non-switchable disk pool because switchable entity already contains disk units that are configured in a switchable disk pool
304	Invalid disk unit for non-switchable disk pool because switchable entity blocks access to switchable disk pool
351	Invalid disk unit for switchable disk pool because switchable entity has disk units configured in non-switchable disk pool
353	Invalid disk unit for switchable disk pool because some node(s) in the cluster resource group recovery domain cannot access switchable entity
355	Invalid disk unit for switchable disk pool because switchable entity has disk units configured in another switchable disk pool in a different cluster resource group
356	Invalid disk unit for switchable disk pool because switchable entity is in different power domain than disk units already configured in the specified disk pool
358	Invalid disk unit for switchable disk pool because switchable entity is in different power domain than disk units already configured in the specified disk pool
360	Invalid disk unit for switchable disk pool because Virtual OptiConnect is not configured
366	Invalid disk unit for switchable disk pool because disk unit is non-switchable, but could be made switchable
367	Invalid disk unit for switchable disk pool because disk unit is non-switchable

D

The SETASPGRP command

This appendix describes the Set Auxiliary Storage Pool (SETASPGRP) command. It is essentially the help text from the command on V5R2 and the syntax for the command based on the Information Center. For more information, go to the Information Center and type `SETASPGRP command` in the Search field. You can locate the Information Center on the Web at:

`http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html`

Details of the SETASPGRP command

The SETASPGRP command sets the auxiliary storage pool (ASP) group for the current thread. In addition, this command allows you to change the libraries in the library list for the current thread. If an ASP group was already set, this command removes the old ASP group from the current thread and sets the specified ASP group for the current thread. After the specified ASP group is set for the current thread, all libraries in the independent ASPs in the ASP group are accessible. Objects in those libraries can be referenced using regular library-qualified object name syntax.

The libraries in the independent ASPs in the specified ASP group, plus the libraries in the system ASP (ASP number 1) and basic user ASPs (ASP numbers 2 to 32), form the library name space for the thread. All libraries in the library list need to be in the new library name space or the library list is not changed and the new ASP group is not set.

Figure D-1 shows an example of the SETASPGRP display.

```
                          Set ASP Group (SETASPGRP)

Type choices, press Enter.

ASP group  . . . . . .  _________    Name, *CURUSR, *NONE
System library list  . . . . . .  *CUR*SYSBAS    *CUR*SYSBAS, *SYSVAL
Current library  . . . . . .  *CUR*SYSBAS    Name, *CUR*SYSBAS, *CURUSR...
Libraries for current job  . . .  *CUR*SYSBAS    Name, *CUR*SYSBAS, *CURUSR...
              + for more values
```

Figure D-1 Set ASP Group (SETASPGRP) example display

Here are some restrictions that you must keep in mind:

- You must have *USE authority to all ASP device descriptions in the ASP group and to all the specified libraries in the library list before the library name space and the library list are changed. If you are not authorized to an ASP device description or to one of the libraries, the ASP group is not set and the library list is not changed.
- When *CURUSR is specified for the ASPGRP or USRLIBL parameter, you must have *READ authority to the job description (JOBD) listed in your user profile and *EXECUTE authority to the library containing that JOBD.

The required parameters include:

- **ASP group (ASPGRP)**: This parameter specifies the name of the ASP group to set for the current thread. The ASP group name is the name of the primary ASP device within the ASP group.

  The possible values are:

  – *auxiliary-storage-pool-group-name*

    This value specifies the name of the primary ASP in the ASP group to be set for the current thread. All libraries from all ASPs in this ASP group are included in the library name space.

  – **CURUSR*

    The ASP group is set to the value defined for the Initial ASP group in the default job description of the user profile under which the thread is currently running.

– **NONE*

This value specifies for the current thread to have no ASP group. The library name space does not include libraries from any ASP group. Only the libraries in the system ASP and any basic user ASPs are in the library name space.

- **System library list (SYSLIBL)**: This parameter specifies the system part of the library list for the thread in which the command is entered.

  The possible values are:

  – **CUR*SYSBAS*

  The libraries in the system part of the library list of the current thread that are found in the system ASP (ASP number 1) or any configured basic user ASP (ASP numbers 2 to 32) are used as the new system part of the library list.

  – **SYSVAL*

  The system part of the library list is set from the current value of system value QSYSLIBL.

- **Current library (CURLIB)**: This parameter specifies the library to be used in the current library entry of the library list for the thread. If *CURUSR or a library name is specified and the library cannot be found in the new library name space, an error message is sent and the library list and ASP group are not changed.

  The possible values are:

  – **CUR*SYSBAS*

  The library name in the current library entry of the library list is used as the new current library if the library is found in the system ASP (ASP number 1) or any configured basic user ASP (ASP numbers 2 to 32). If the library name in the current entry is not found in the system ASP or any basic user ASP, the current library entry is removed from the library list.

  – **CURUSR*

  The current library is set to the value defined for Current library in the user profile under which the thread is currently running.

  – **CRTDFT*

  This value changes the library list to remove any name from the current library entry. If objects are created specifying *CURLIB for the library name, library QGPL is used.

  – *library-name*

  This value specifies the name of the library that replaces the current library entry in the library list.

- **Libraries for current job (USRLIBL)**: This parameter specifies the libraries that are placed in the user part of the library list. If *CURUSR or a list of library names is specified and any of these libraries cannot be found in the new library name space, an error message is sent and the library list and ASP group are not changed.

  The possible values are:

  – **CUR*SYSBAS*

  The libraries in the user part of the library list of the current thread that are found in the system ASP (ASP number 1) or any configured basic user ASP (ASP numbers 2 to 32) are used as the new user part of the library list.

– **CURUSR*

  The user part of the library list for the thread is set to the value defined for the Initial library list in the default job description of the user profile under which the thread is currently running.

– **SYSVAL*

  The user part of the library list is set from the current value of system value QUSRLIBL.

– **NONE*

  This value changes the user part of the library list to remove all library names.

– *library-name*

  This value specifies the names of the libraries to be used as the user part of the library list, in the order in which they are to be searched. The number of libraries that can be specified ranges from 1 through 250.

Examples of using the SETASPGRP command

The following sections provide some examples of using the SETASPGRP command.

Example 1: Setting as new ASP group

The command as shown in Figure D-2 sets the ASP group for the thread in which the command runs to be WAREHOUSE1. This changes the library name space for the thread to include all libraries in any of the independent ASPs in the ASP group identified by the independent ASP device named WAREHOUSE1.

The system part of the library list is set from the system value QSYSLIBL. The current library entry of the library list is set from the Current library value defined in the user profile under which the thread is currently running. The user part of the library list is set using the current user part of the library list and by removing any libraries that are not found in the system ASP or configured basic user ASPs.

```
SETASPGRP ASPGRP(WAREHOUSE1)
          SYSLIBL(*SYSVAL)
          CURLIB(*CURUSR)
```

Figure D-2 Setting a new ASP group

Example 2: Specifying no ASP group

The command as shown in Figure D-3 removes any ASP group for the thread in which the command runs. This changes the library name space for the thread to include only those libraries in the system ASP (ASP number 1) and basic user ASPs (ASP numbers 2 to 32).

The system part of the library list is set using the current system part of the library list and by removing any libraries that are not found in the system ASP or configured basic user ASPs. The current library entry of the library list is changed to be empty. This causes library QGPL to be used as the current library. The user part of the library list is changed to be empty.

```
SETASPGRP ASPGRP(*NONE)
          CURLIB(*CRTDFT)
          USRLIBL(*NONE)
```

Figure D-3 Set to no ASP group

Escape message for the SETASPGRP command

The escape message for the SETASPGRP command appears as shown here:

```
*ESCAPE Messages
   CPFB8E9
      ASP group &1 not set for thread &2
```

Figure D-4 shows the command syntax for the SETASPGRP command. All parameters that precede this point can be specified in positional form. Also, there is a maximum of 250 repetitions.

Figure D-4 SETASPGRP (Set ASP Group) syntax diagram

E

Command-to-Navigator cross reference

The primary reason for this cross-reference was the widespread impact to commands as a result of adding general ASP functionality to OS/400. Some functions are easier to perform through commands, where other functions are easier to do through iSeries Navigator. In addition, you can perform some functions only by using commands, while you can perform other functions only using iSeries Navigator.

Table E-1 shows a function to command and iSeries Navigator cross-reference. It includes a summary of the keywords and parameters.

Table E-1 Function to command and iSeries Navigator cross-reference

Description of function	Command	iSeries Navigator
ASP functions	**ASP commands**	
The Change Auxiliary Storage Pool Attributes command allows you to change attributes that control the behavior of an auxiliary storage pool (ASP).	CHGASPA	
The Change Device Description command changes the device description for an ASP device.	CHGDEVASP	
The Create Device Description command creates a device description for an ASP device.	CRTDEVASP	
The Set Auxiliary Storage Pool Group command sets the ASP group for the current thread. This command also allows you to change the libraries in the library list for the current thread.	SETASPGRP	
The End ASP Balance command allows you to end the ASP balance function that was started using the Start ASP Balance CL command.	ENDASPBAL	

Description of function	Command	iSeries Navigator
The Start ASP Balance command allows you to start the ASP balancing function for one or more ASPs.	STRASPBAL	
The Trace ASP Balance command controls the function that gathers the ASP usage statistics.	TRCASPBAL	
The Check ASP Balance command allows you to check which ASP balance function is currently active and which units have been marked to not allow new allocations (*ENDALC).	CHKASPBAL	
The Work with ASP Descriptions command takes you to the Work with ASP Descriptions display or produces the ASP Descriptions report.	WRKASPBRM	
AUT functions	**AUT commands**	
The Restore Authority command restores the private authorities to user profiles.	RSTAUT	
DSK functions	**DSK commands**	
The Print Disk Information command is used to print disk space information that was stored in database file QAEZDISK or QAEZD*nnnnn* by the Retrieve Disk Information command, where *nnnnn* is the ASP number of the independent ASP (IASP) for which disk space information was retrieved.	PRTDSKINF	
The Retrieve Disk Information command is used to collect disk space information.	RTVDSKINF	
The Start Disk Reorganization command allows you to start the disk reorganization function for one or more ASPs.	STRDSKRGZ	
JOB functions	**JOB commands**	
The Retrieve Job Attributes command is used in a CL program to retrieve the values of one or more job attributes and place the values into the specified CL variable.	RTVJOBA	
The Submit Job command allows a job that is running to submit another job to a job queue to be run later as a batch job.	SBMJOB	
JOBD functions	**JOBD commands**	
The Change Job Description command changes the job-related attributes specified for a job description object through the Create Job Description command.	CHGJOBD	
The Create Job Description command creates a job description object that contains a specific set of job-related attributes that can be used by one or more jobs.	CRTJOBD	

Description of function	Command	iSeries Navigator
JRN functions	**JRN commands**	
The Create Journal command creates a journal as a local journal with the specified attributes, and attaches the specified journal receiver to the journal.	CRTJRN	
The Create Journal Receiver command creates a journal receiver. After a journal receiver is attached to a journal, journal entries can be placed in it.	CRTJRNRCV	
Library functions	**LIB commands**	
The Clear Library command deletes all of the objects from the specified library that you have the authority to delete.	CLRLIB	Go to Connection, Database, IASP Name, libraries, click **Library**. In the opposite pane, click the first object. Press and hold down the Shift key. Scroll down to the last object and click it. Right-click the highlighted objects, and select **Delete**.
The Create Library command adds a new library to the system.	CRTLIB	Go to Connection, Database, IASP Name, Right-click **libraries**, New Library, complete Library and Description, click **OK**.
The Delete Library command deletes a specified library from the system after all objects in the library are deleted. If a library that is deleted contains objects, this command first deletes all of the objects and then deletes the library.	DLTLIB	Go to Connection, Database, IASP Name, Right-click the library name and select **Delete**.
The Display Library command displays the contents of one or more specified libraries. That is, it displays a list of the names and types of all objects contained in each library, regardless of the authorization on each object.	DSPLIB	Go to Connection, Database, IASP Name, right-click libraries, Select libraries to display (Repeat for each IASP)
The Restore Library command restores to the system one library or a group of libraries that was saved by the Save Library command.	RSTLIB	Available through Backup Recovery and Media Services (BRMS) plug-in only
The Retrieve Library Description command is used to retrieve the description of a library.	RTVLIBD	
The Save Library command allows you to save a copy of one or more libraries.	SAVLIB	
The Save/Restore Library command allows you to save and restore a copy of one or more libraries to another system.	SAVRSTLIB	
The Work with Libraries command shows a list of libraries and allows you to copy, delete, display, print, save, restore, change, and clear specified libraries.	WRKLIB	
NWS function	**NWS command**	
The Create Network Server Description command creates a description for a network server.	CRTNWSD	

Description of function	Command	iSeries Navigator
The Create Network Server Storage Space command creates a storage space used by a network server.	CRTNWSSTG	
Object functions	**OBJ commands**	
The Analyze User Objects command collects or reports information for user-created objects on the system.	ANZUSROBJ	
The Display Object Description command shows the names and attributes of specified objects in the specified library or in the libraries of the thread's library list.	DSPOBJD	
The Change Object Auditing command allows users with *AUDIT special authority to set up auditing on an object.	CHGOBJAUD	
The Change Object Owner command transfers object ownership from one user to another.	CHGOBJOWN	
The Change Object Primary Group command changes the object's primary group from one user to another.	CHGOBJPGP	
The Create Duplicate Object command copies a single object or a group of objects.	CRTDUPOBJ	
The Display Object Description command shows the names and attributes of specified objects in the specified library or in the libraries of the thread's library list.	DSPOBJD	
The Display Object Authority command displays the list of authorized users of an object and their assigned authority.	DSPOBJAUT	
The Edit Object Authority command displays the list of authorized users of an object and their associated user authorities.	EDTOBJAUT	
The Grant Object Authority command grants specific authority for the objects named in the command.	GRTOBJAUT	
The Move Object command removes an object from its currently assigned library and places it in a different library.	MOVOBJ	
The Rename Object command changes the name of an object in a library.	RNMOBJ	
The Retrieve Object Description command returns the description of a specific object to a CL program or REXX procedure.	RTVOBJD	
The Revoke Object Authority command is used to take away specific (or all) authority for the named object or objects from one or more users named in the command, or to remove the authority of an authorization list for the named object or objects.	RVKOBJAUT	

Description of function	Command	iSeries Navigator
The Save Changed Object command saves a copy of each changed object or group of objects located in the same library.	SAVCHGOBJ	
The Save Object command saves a copy of a single object or a group of objects located in the same library.	SAVOBJ	
The Save/Restore Object command saves and restores a single object, or a group of objects located in the same library, to another system.	SAVRSTOBJ	
The Save/Restore Changed Object command saves and concurrently restores a copy of each changed object, or group of objects located in the same library, to another system.	SAVRSTCHG	
The Restore Object command restores to the system a single object or a group of objects in a single library that were saved on diskette, tape, optical volume, or in a save file using a single command.	RSTOBJ	
The Work with Object Locks command allows you to work with the object lock requests in the system for a specified object.	WRKOBJLCK	
The Work with Objects by Owner command is used to manage objects for any user profile.	WRKOBJOWN	
RCY functions	**RCY commands**	
The Change Recovery for Access Paths command is used to change the target access path recovery time for the system or for one or more ASPs.	CHGRCYAP	
The Display Recovery for Access Paths display shows a list of access path recovery times for the system and for the ASPs that are currently on the system.	DSPRCYAP	
The Edit Recovery for Access Paths display shows a list of access path recovery times for the system and for ASPs that are currently active on the system.	EDTRCYAP	
Miscellaneous functions	**Miscellaneous commands**	
The Create Save File command creates a save file.	CRTSAVF	
The Install Windows Server command installs the Windows server base operating system on an Integrated xSeries Server.	INSWNTSVR	
The Save Security Data command saves all security information without requiring a system in a restricted state.	SAVSECDTA	
The Save command saves a copy of one or more objects that can be used in the integrated file system.	SAV	

Description of function	Command	iSeries Navigator
The Work with Device Descriptions command is used to display and to work with device description functions through the Work with Device Descriptions display.	WRKDEVD	Parameter *ASP

F

Supported and unsupported objects, APIs

Table F-1 lists all object types on V5R2 and whether they are supported in an IASP. This table is in Object type sequence.

As a general rule, no object can be referenced unless the independent disk pool is made available, and, the SETASPGRP command is run to assure the ASP device is in the job's thread.

Table F-1 Object types in IASPs

Supported yes/no	Object type	Description	Qualifications
Yes	*ALRTBL	Alert Table	If pointed to from the network attributes, this object needs to be in *SYSBAS.
No	*AUTL	Authorization List	
Yes	*BLKSF	Block Special File	
Yes	*BNDDIR	Binding Directory	
No	*CFGL	Configuration List	
Yes	*CHRSF	Character Special File	
Yes	*CHTFMT	Chart Format	
Yes	*CLD	C Locale Description	
Yes	*CLS	Class	If referenced from an Active SBSD, this object needs to be in *SYSBAS.
Yes	*CMD	Command	
No	*CNNL	Connection List	
No	*COSD	Class-of-service Description	

Supported yes/no	Object type	Description	Qualifications
No	*CRG	Cluster resource group	
Yes	*CRQD	Change Request Description	
Yes	*CSI	Communication Side Information Object	
No	*CSPMAP	Cross System Product Map	
No	*CSPTBL	Cross System Product Table	
No	*CTLD	Controller Description	
No	*DDIR	Distributed Directory	
No	*DEVD	Device Description	
Yes	*DIR	Directory (QDLS)	
No	*DOC	Document	
Yes	*DSTMF	Distributed Stream File	
Yes	*DTAARA	Data Area	
Yes	*DTADCT	Data Dictionary	
Yes	*DTAQ	Data Queue	
No	*EDTD	Edit Description	
No	*EXITRG	Exit Registration	
Yes	*FCT	Forms Control Table	
Yes	*FIFO	First-in-first-out Special File	
Yes	*FILE	File	If referenced from an Active SBSD, this object needs to be in *SYSBAS (for example, the sign-on display file).
No	*FLR	Folder	
Yes	*FNTRSC	Font Resource	
Yes	*FNTTBL	Font Mapping Table	
Yes	*FORMDF	Forms Definition	
Yes	*FTR	Filter	
Yes	*GSS	Graphics Symbol Set	
Yes	*IGCDCT	DBCS Conversion Dictionary	
No	*IGCSRT	DBCS Sort Table	
No	*IGCTBL	DBCS Font Table	
No	IMGCLG	Image Catalog	
No	*IPXD	Internet Packet Exchange Description	

Supported yes/no	Object type	Description	Qualifications
Yes	*JOBD	Job Description	If referenced from an Active SBSD, this object needs to be in *SYSBAS (for example, autostart job entry, communication entry, remote location name entry, workstation entry, etc.).
No	*JOBQ	Job Queue	
No	*JOBSCD	Job Schedule	
Yes	*JRN	Journal	
Yes	*JRNRCV	Journal Receiver	
Yes	*LIB	Library	
No	*LIND	Line Description	
Yes	*LOCALE	Locale	
No	*M36	System 36 Machine	
No	*M36CFG	System 36 Machine Configuration	
Yes	*MEDDFN	Media Definition	
Yes	*MENU	Menu	Give special thought to where initial menus in user profiles are located, especially for QSECOFR.
Yes	*MGTCOL	Management Collection	
No	*MODD	Mode Description	
Yes	*MODULE	Module	
Yes	*MSGF	Message File	
Yes	*MSGQ	Message Queue	If referenced by a Network Attribute, it must be in *SYSBAS.
Yes	*NODL	Node List	
Yes	*NODGRP	Node Group	
No	*NTBD	NetBIOS Description	
No	*NWID	Network Identifier	
No	*NWSD	Network server Description	
No	*OUTQ	Output Queue	
Yes	*OVL	Overlay	
Yes	*PAGDFN	Page Definition	
Yes	*PAGSEG	Page Segment	
Yes	*PDG	Printer Description Group	

Supported yes/no	Object type	Description	Qualifications
Yes	*PGM	Program	If referenced from an Active SBSD, this object needs to be in *SYSBAS (for example, routing entries, prestart job entries).
Yes	*PNLGRP	Panel Group	
No	*PRDAVL	Product availability	
Yes	*PRDDFN	Product Definition	
Yes	*PRDLOD	Product Load	
Yes	*PSFCFG	Printer services Facility Configuration	
Yes	*QMFORM	Query Form	
Yes	*QMQRY	Query Manager Query	
Yes	*QRYDFN	Query Definition	
Yes	*RCT	Reference code translation Table	
No	*S36	System 36 Machine Description	
Yes	*SBSD	Subsystem Description	To go active, this object must reside in *SYSBAS
Yes	*SCHIDX	Search Index	
No	*SOCKET	Socket	
Yes	*SPADCT	Spelling Aid Dictionary	
Yes	*SQLPKG	SQL Package	
Yes	*SQLUDT	User Defined SQL Type	
Yes	*SRVPGM	Service Program	
No	*SSND	Session Description	
Yes	*STMF	Stream File	
Yes	*SVRSTG	Server Storage	
Yes	*SYMLNK	Symbolic Link	
Yes	*TBL	Table	
Yes	*USRIDX	User-defined Index	
No	*USRPRF	User Profile	
Yes	*USRQ	User Queue	
Yes	*USRSPC	User-defined Space	
Yes	*VLDL	Validation List	
Yes	*WSCST	Workstation Customization Table	

APIs relating to IASPs

This section lists the APIs that you can use to query or manipulate IASP information.

QYASPOL

The Open List of ASPs (QYASPOL) API generates a list of ASPs or information about an ASP. The information may include:

- Identification of all ASPs configured to a system
- Attributes of an ASP
- Unassigned disk units or disk units assigned to an ASP
- Hardware problems during vary on of an independent ASP
- Current vary on activity
- Jobs using an independent ASP

Upon successful completion of this API, a handle is returned in the list information parameter. You may use this handle on subsequent calls to the following APIs:

- Get List Entries (QGYGTLE)
- Close List (QGYCLST)

This API resides in OS/400, 5722-SS1, BOSS Option 12 (Host Servers) in library QGY.

QGYCLST

The Close List (QGYCLST) API closes a previously opened list. Any internal storage associated with that list is freed. The handle specified on the call to this API is no longer valid after the call completes.

The handle is generated by one of the following list APIs:

- Open List of Jog Log Messages (QGYOLJBL)
- Open List of Messages (QGYOLMSG)
- Open List of Objects (QGYOLOBJ)
- Open List of Printers (QGYRPRTL)
- Open List of Spooled Files (QGYOLSPL)
- Open List of ASPs (QYASPOL)

QGYGTLE

The Get List Entries (QGYGTLE) API allows requests to get entries from previously opened lists on the iSeries server. A list exists if an initial request is already made and the list is not closed using the Close List (QGYCLST) API.

Initial requests are made by calling the following APIs:

- Open List of Job Log Messages (QGYOLJBL)
- Open List of Messages (QGYOLMSG)
- Open List of Objects (QGYOLOBJ)
- Open List of Printers (QGYRPRTL)
- Open List of Spooled Files (QGYOLSPL)
- Open List of User Certificates (QSYOLUC)
- Open List of Validation List Entries (QSYOLVLE)
- Open List of ASPs (QYASPOL)
- Retrieve Objects Secured by Authorization List (QGYRATLO)

QHSMMOVL

The Move Library to ASP (QHSMMOVL) API moves a library and its contents from its existing ASP to the specified target ASP through a save and restore process. The API, however, preserves private authorities to the objects that are normally lost with a save and restore operation.

The Move Library to ASP (QHSMMOVL) API has the following restrictions:

- When you move a library from the system ASP to a user ASP, libraries and their objects are checked for eligibility. Libraries that begin with the letter Q are considered to be system libraries and are not moved. See *iSeries Backup and Recovery*, SC41-5304, for detailed information about the list of objects that cannot be moved.
- If the library cannot be renamed, the library is not allowed to be moved. See the Rename Object (RNMOBJ) Control Language (CL) information in the iSeries Information Center for the restrictions on renaming a library. The Information Center is located on the Web at:

  `http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html`
- Libraries that contain journal objects or journaled files are not allowed to be moved.
- Libraries that contain files with database dependencies outside the library are not allowed to be moved.
- Job queue and output queue entries are moved if the queue is not allocated.
- Data queue entries are the responsibility of the user and are lost.
- A library cannot be moved if it is in the library list of the current thread.
- A library cannot be moved if it is in the library list of any primary thread that is active on the system when the QLIBLCKLVL system value is set to lock libraries in the library list.
- The QSYSWRK subsystem must be active.
- Program (*PGM) objects in a library are placed in library QRPLOBJ (or library QRPL*xxxxx* if the library is in a primary ASP with an ASP number corresponding to *xxxxx* (where *xxxxx* is the ASP number right adjusted and padded on the left with zeros) or in a secondary ASP in the same group as that primary ASP)) and another copy of each *PGM object are moved with the library to the target ASP.
- After a library is moved, the following attributes are changed:
  - The date last used is set to blank.
  - The change date and time are set to the current date and time.
  - The days used count is set to zero.
  - The date use count reset is set to blank.
  - The restore date and time are set to the current date and time.
- The target ASP must have enough space for the library and its objects for the API to perform the move operation.
- The target ASP must be either the system ASP (ASP 1), a library-type or empty basic user ASP (ASPs 2 to 32), or a primary or secondary ASP (ASPs 33 to 255).

QHSMMOVF

The Move Folder to ASP (QHSMMOVF) API moves a root folder and its contents from its existing ASP to the specified target ASP through a save and restore process. The API, however, retains private authorities to the objects that are normally lost with a save and restore operation.

The Move Folder to ASP (QHSMMOVF) API has the following restrictions:

- The folder must be a root folder and is moved as such.
- The folder and its contents must not be in use by other jobs.
- Folders that were restored using Restore Licensed Program (RSTLICPGM) should not be moved.
- After the root folder is moved, the following parameters are changed:
  - The date last used is set to blank.
  - The change date and time are set to the current date and time.
  - The days used count is set to zero.
  - The days used count reset is set to blank.
  - The save date/time and restore date/time are updated.
- The target ASP must have enough space for the folder and its objects for the API to perform the move action.
- The target ASP must be either the system ASP, a library-type ASP, or an empty ASP.
- The user must be enrolled in the system distribution directory.
- Access codes are the responsibility of the user and are lost.
- A root folder that contains documents checked out or saved with STG(*FREE) is not moved.
- A root folder that contains more than 99 subfolders is not moved.

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 209.

- *IBM @server iSeries and AS/400e System Builder*, SG24-2155
- *Implementing SAP R/3 on OS/400*, SG24-4672
- *Clustering and IASPs for Higher Availability on the IBM @server iSeries Server*, SG24-5194
- *Linux on the IBM @server iSeries Server: An Implementation Guide*, SG24-6232
- *LPAR Configuration and Management: Working with IBM @server iSeries Logical Partitions,* SG24-6251
- *IBM @server iSeries Server Migration: System Migration and Upgrades at V5R1 and V5R2*, SG24-6055

Other publications

iSeries Backup and Recovery, SC41-5304, is also relevant as a further source of information.

Referenced Web sites

These Web sites are also relevant as further information sources:

- iSeries High Availability and Clusters Web site

  http://www-1.ibm.com/servers/eserver/iseries/ha/

- iSeries Logical Partition Web information

  http://www-1.ibm.com/servers/eserver/iseries/lpar/

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.

Index

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

X

IBM @server iSeries Independent ASPs: A Guide to Moving Applications to IASPs

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

IBM @server iSeries Independent ASPs

A Guide to Moving Applications to IASPs

Redbooks

Take advantage of the new V5R2 support for independent ASPs (IASPs)

Improve your uptime by using switchable IASPs

Install and configure IASPs for your environment

This IBM Redbook explains how to install and configure the new independent auxiliary storage pool (IASP) functionality of OS/400 V5R2. It is designed to help IBM technical professionals, Business Partners, and Customers understand and implement IASP in the @server iSeries™ server and under OS/400® V5R2.

In addition this redbook provides the background information that is necessary to plan, implement, and customize this functionality to your particular environment. It provides advice on running native OS/400 applications with either application data or most application objects residing in an IASP. Considering you can also use IASPs in a cluster environment, this redbook shows you the basic steps to make your IASP switchable between two iSeries servers in a high-speed link (HSL) loop.

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

SG24-6802-00 ISBN 0738425745