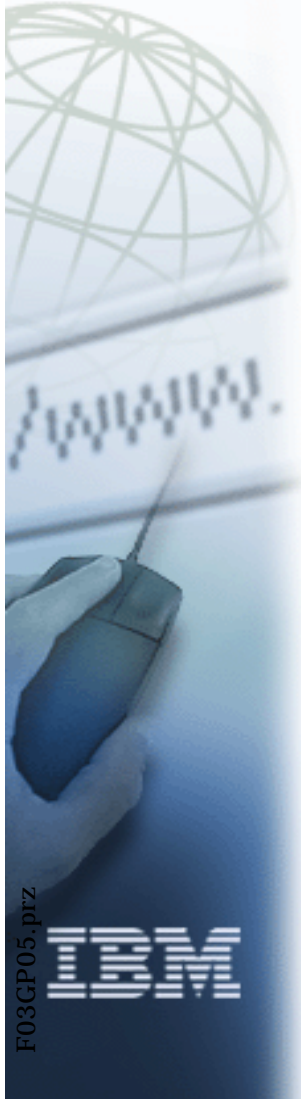


ibm.com



e-business



Moving Applications to iASPs

GP05

ITSO iSeries Technical Forum

Nick Harris



Doug Bidwell
Darryl Conner
Alan Kennedy
Stu Preacher

Redbooks

International Technical Support Organization

© 2002 IBM Corporation

Trademarks and Disclaimers



© IBM Corporation 1994-2002. All rights reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

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

AS/400	IBM (logo)
AS/400e	iSeries
e (logo) business	OS/400
IBM	

Lotus, Freelance Graphics, and Word Pro are registered trademarks of Lotus Development Corporation and/or IBM Corporation.

Domino is a trademark of Lotus Development Corporation and/or IBM Corporation.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. 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.

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

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

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information in this presentation concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information in this presentation addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

Objectives



The objectives of this session are to run through the V5R2 iASP support. Includes iASP support as it relates to the January 2003 announcement.

With this support most OS/400 objects can now be located in an iASP. This offers new opportunities for server consolidation.

iASPs are also available able as a switchable resource in a cluster environment. We will discuss this support.

This presentation under construction until GA.



Agenda



- Independant storage pools overview
- Planning for iASPs
- Operating iASPs
- iASP Scenarios
- iASP Benefits

Notes Agenda



- Introduction to iASPs
 - We will cover the basic iASP concept. Disk pools, storage layout and the iASP environment.
- Planning for iASPs
 - In planning for iASPs we will cover objects supported and not supported, disk arrangements for iASP, iASP structure, and migration.
- Operating iASPs
 - To manage and operate one's iASPs there are some new functions and commands
- iASP Scenarios
 - Standalone iASP setup
 - Switchable iASP support
 - Linux and Windows in NWSSTG spaces
- iASP Benefits



Independent Disk Storage Pools (iASP) Overview

iASP and January 2003 announcement



- No change for basic iASP support
- New models all support iASPs
- New towers and expansion units support iASPs
- New faster IOAs and disk will improve iASP performance

iASP is a.....



An Independent ASP is a method of configuring multiple disk storage units into one group. The group can include

Non-switchable

- Any or all units in the system tower, except the load source
- Any one or more disks anywhere on the system

Switchable, all disk units in 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

Notes iASP is a.....



An Independent ASP is a method of configuring multiple disk storage units into one group. The group can include

Non-switchable

- Any or all units in the system tower, except the load source
- Any one or more disks anywhere on the system

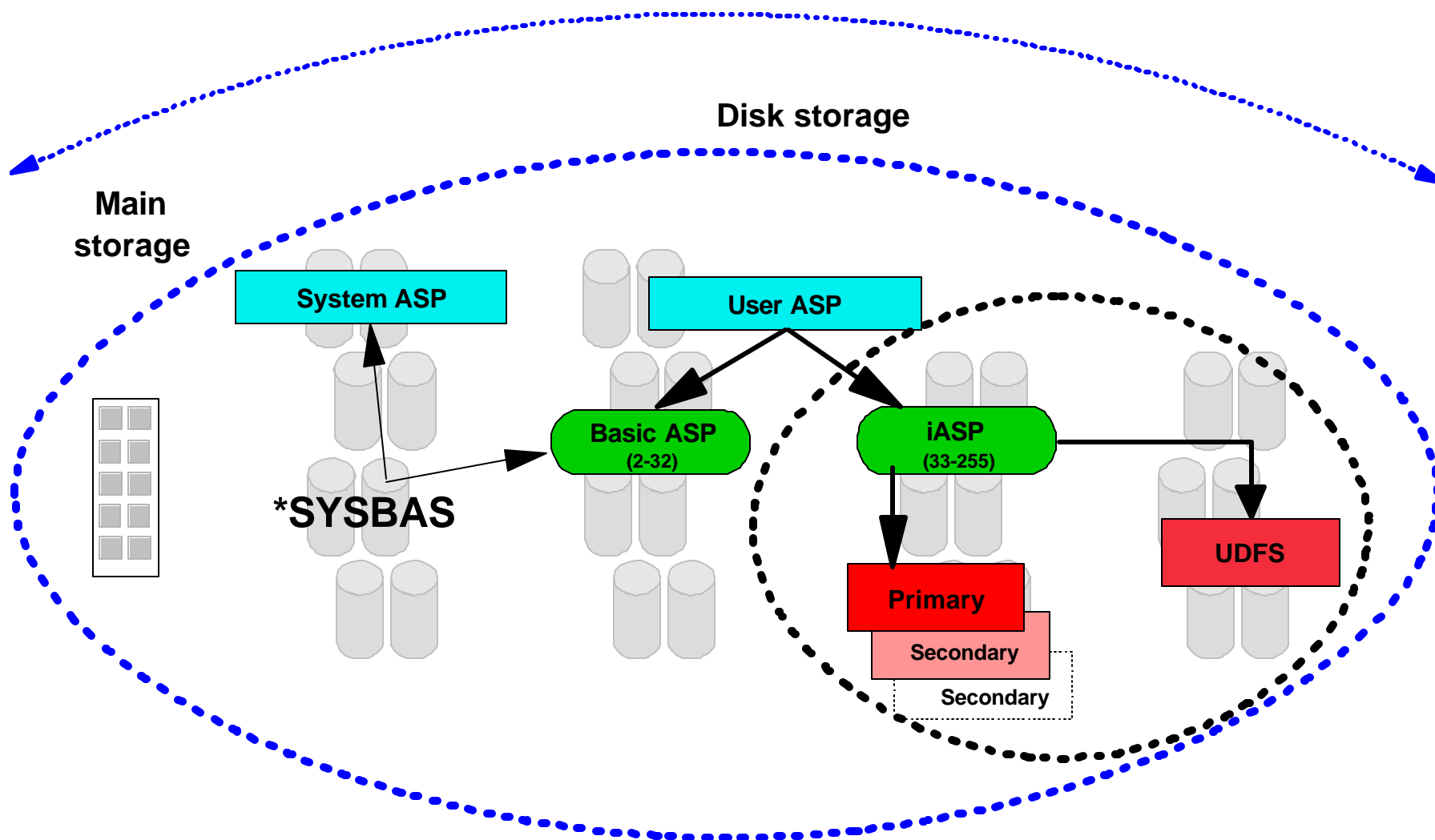
Switchable, all disk units in tower other than the system tower.

- 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

iSeries Storage Layout



Single Level Storage



Notes iSeries Storage Layout



In the chart you can see the full breadth of single level storage (SLS) spanning main storage and disk storage. Within the disk storage the disk are allocated to either the system ASP or a user ASP. Both these spaces are part of SLS. The user ASP portion can be further sub-divided into Basic ASPs or Independent ASPs

*SYSBAS is the system ASP and all Basic User ASPs

Basic ASP = Auxiliary Storage Pool (Disk Pool)

- An ASP is a software definition of a group of disk units on your system. An ASP provides a way of organizing data to limit the impact of storage-device failures and to reduce recovery time.
- A basic disk pool is used to isolate some objects from the other objects that are stored in the system disk pool. Data in a basic disk pool is always accessible whenever the server is up and running.

IASP = Independent ASP (Independent Disk Pool)

- An independent ASP can be made available and made unavailable to the server without restarting the system. When an independent ASP is associated with a switchable hardware group, it becomes a switchable ASP and can be switched between one server and another server in a clustered environment.
- Differences from Basic User ASP
 - Introduced in R510
 - Identified by device name
 - Overflow not allowed
 - VRYCFG used to independently bring online or take off-line

Type of ASP	OS/400 release supported	ASP numbers	Maximum quantity supported on the system
System ASP	All	ASP 1	1
Basic User ASP1	V4R5 and earlier	ASP 02 - ASP 16	15
Basic User ASP1	V5R1	ASP 02 - ASP 32	31
Independent ASP	V5R1	ASP 33 - ASP 99	67
Independent ASP	V5R2	ASP 33 - ASP 255	223

Disk Pools



System ASP

- ASP# 1
- OS/400

Basic

- ASP# 2-32
- Also known as User or Dependent ASPs

Independent (IASP)

- ASP# 33-255
- User Defined File System (UDFS) (V5R1)
- Primary (QSYS.LIB objects) (V5R2)
 - Secondary

The screenshot shows the 'As25: Disk Pools' window with a table of disk pools and a 'New Disk Pool' dialog box open.

Disk Pool	Capacity	% Used	Free Space
Disk Pool 1	171.8 GB	16%	143.9 GB
Disk Pool 33 (Dbitsc)	34.3 GB	0%	34.0 GB
Disk Pool 34 (Europe)	17.2 GB	Unknown	Unknown

The 'New Disk Pool' dialog box shows the following configuration:

- Type of disk pool: Secondary
- Disk pool: Journals
- Primary: (Libobjects)
- Note: If you want to create a switchable disk pool, be sure to use the appropriate clustering function before using this wizard.
- Protect the data in this disk pool
- Buttons: OK, Cancel, Help, ?

Notes Disk Pools



A set of one or more storage units selected from all available disk units. Disk pools (also known as auxiliary storage pools) provide a means of placing certain objects on specific disk units to prevent the loss of data due to failures of disk units in other disk pools.

When independent disk pools were introduced in V5R1, they supported user-defined file systems (UDFS) only. Support for library-based (QSYS.LIB) objects has been added in V5R2. You can now create as many as 223 independent disk pools. Previous releases only supported 67 independent disk pools. In V5R1 independent disk pools were numbered from 33-99. That range has been expanded to 33-255 at V5R2. See "Supported and unsupported OS/400 object types" in the iseries Information Center for details. There are three types of disk pools, also known as auxiliary storage pools (ASPs):

- **System**

The system automatically creates the system disk pool (Disk Pool 1) which contains disk unit 1 and all other configured disks that are not assigned to a basic or independent disk pool. The system disk pool contains all system objects for the OS/400 licensed program and all user objects that are not assigned to a basic or independent disk pool. You can see the system ASP as Disk Pool 1 in foil.

- **Basic**

Prior to V5R2, Auxiliary Storage Pools 2 to 32 were known as User ASPs. The function of these has not changed but they are now often referred to Basic User ASPs or Basic Disk Pools in V5R2. Prior to V5R2 they were referred to as Dependent Pools.

- **Independent**

A disk pool that contains objects, the 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 or non-switchable IASP. Independent disk pools can also function in conjunction with other independent disk pools in a disk *pool group*.

In the upper right window pane you can see "statistics" are known for the system ASP and IASP 33 (named Dbitsc). IASP Dbitsc has been varied on to the local system so the system "knows the disk configuration is active. IASP disk pool 34 (Europe) % Used and amount of Free Space. statistics are not known because this

Notes Disk Pools 2



Three type of IASPs can be defined in V5R2:

– **User-defined file system (UDFS)**

An independent disk pool that contains only user-defined file systems. A UDFS disk pool cannot be a member of a disk pool group unless it is converted to a primary or secondary disk pool. An UDFS IASP is generally the equivalent to the only IASP support in V5R1.

– **Primary**

An independent disk pool that defines a collection of directories and libraries and may have other secondary disk pools associated with it. A primary disk pool also defines a database for itself and other disk pools that may be added in its disk pool group. Primary disk pools can only be implemented on V5R2 or later of OS/400.

– **Secondary**

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. A secondary pool is optional. Though not practical, you can, in theory, have as many as 200 or more secondary pools. Additional details are provided in the following foils.

Disk Pool Groups



Are made up of:

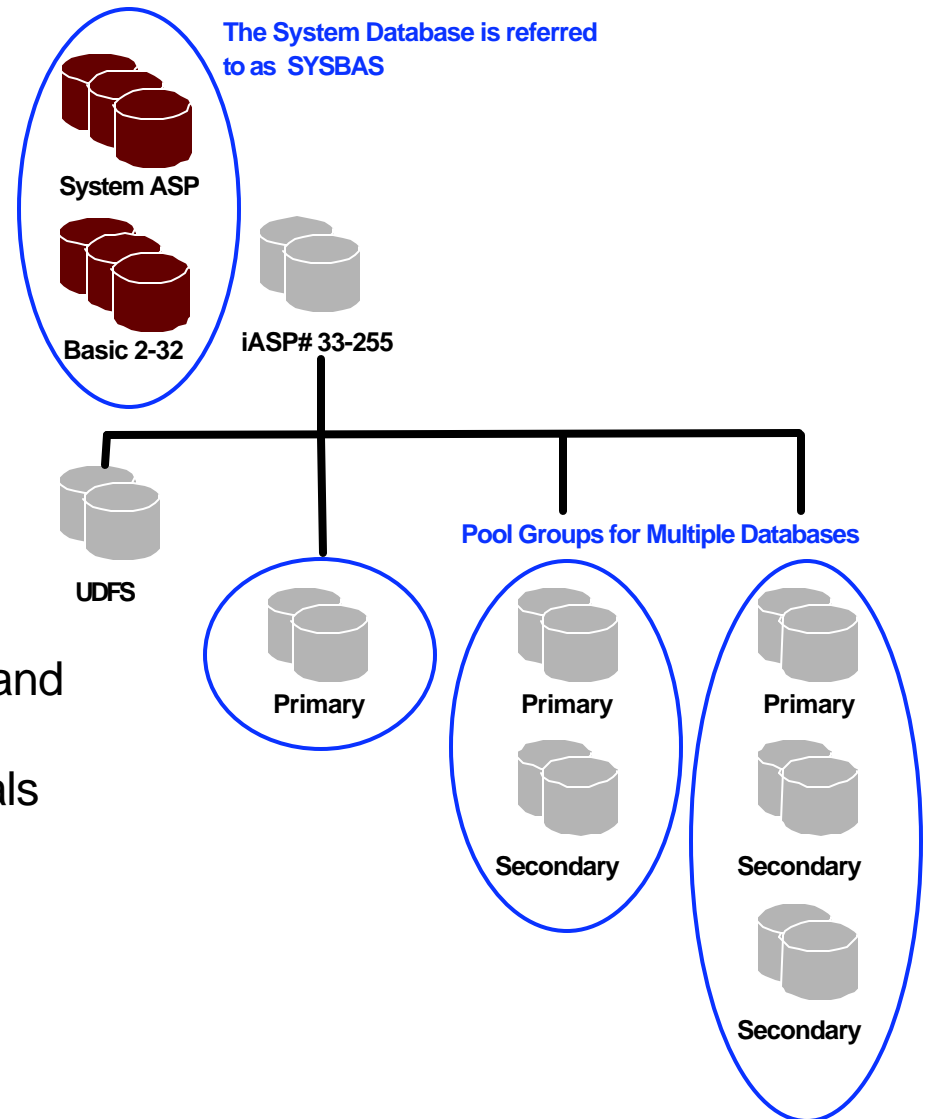
- a primary disk pool
- zero or more secondary disk pools

Groups logically connect disk pools

- Vary them On/Off together
- Switch them together

Share the same database

- Similar as System ASP and basic ASPs
- For example:
 - Primary independent ASP for libraries and database files
 - Secondary independent ASP for journals and journal receivers



Notes Disk Pool Groups



A disk pool group is made up of a primary disk pool and zero or more secondary disk pools. Each disk pool is independent in regard to data storage, but in the disk pool group they combine to act as one entity. If you make one disk pool available or unavailable, the rest of the disk pools in the group are also made available or unavailable at the same time. Also, in a clustered environment, all of the disk pools in a group switch to another node at the same time. The primary and secondary disk pools also share the same database.

An example of a practical use for a secondary IASP disk pool group would be for journals and journal receivers. The primary disk pool could contain the libraries and objects to be journaled, while the secondary disk pools could contain the associated journal receivers. The journals and journal receivers would remain separate for maximum availability and performance, but they would function together in the disk pool group. Disk pool groups can only be implemented on V5R2 or later of OS/400.

This is similar to the system ASP (disk pool 1) and any base pool being treated as an entity sharing the local System Database and any associated journal objects. In V5R2 SYSBAS is the term assigned to refer to the system ASP and any Basic user ASP.

IFS Journalling

With V5R1 you could journal IFS objects but not the ones in an UDFS independent pool. This is because journal is a library based object. Now with V5R2, you can place IFS objects in a library capable pool (or convert an UDFS pool to a library based pool), journal them and place the journal receiver in an associated secondary independent pool. **Only library capable pools can be grouped together.**

Contrast basic and independent disk pools

When the server IPLs, all of the disk units configured in SYSBAS must be accounted for in order for the server to continue the IPL. Independent disk pools are not included in the IPL. When you vary on the independent disk pool the node then verifies that all disk units are present.

Structuring disk pool groups

The iSeries server supports up to 223 independent disk pools, any number of which can be primary, secondary, or UDFS disk pools. Therefore, you have significant flexibility in how you place your data into independent disk pools and how you structure disk pool groups. For example, all application data could be placed in a single disk pool group which consists of one primary disk pool and one secondary disk pool. Alternatively, you could create several disk pool groups, some with only a primary disk pool and some with one or more secondary disk pools.

There are other factors to consider when planning the placement of your data in disk pools. These are dependent on the way your iASPs are used.. Please refer to the iSeries information center for more details.

Multiple Databases



Independent disk pools

- Distinct Databases

Requires V5R2

- A single system can even be a 6xx or 7xx

SETASPGRP command

- Sets the namespace

Reclaim Storage

- In parallel
- No restricted state required
 - except for SYSBAS

The image displays two screenshots of the iSeries Navigator interface. The top screenshot shows the 'Disk Pools' tree structure under 'Environment: My Connections'. It lists 'Disk Pool 1', 'Disk Pool 33 (Libobjects)', and 'Disk Pool 34 (Journals)'. A table on the right, titled 'As20: Group Libobjects', shows the following data:

Disk Pool	Type	Status
Disk Pool 33 (Libobjects)	Primary	Available
Disk Pool 34 (Journals)	Secondary	Available

The bottom screenshot shows the 'Databases' tree structure under 'Environment: My Connections'. It lists 'As20', 'Libobjects', and 'File Systems'. A table on the right, titled 'As20: Databases', shows the following data:

Name	Type	Status	Description
As20	Local	Available	Entry added by system
Libobjects	Local	Available	Multiple Databases for SCON
Benjamin	Remote	Unknown	Entry added by system
Newiasp	Remote	Unknown	Entry added by system
Ryan	Remote	Unknown	Entry added by system

Blue circles and arrows highlight the 'Disk Pool 33 (Libobjects)' and 'Disk Pool 34 (Journals)' entries in the top screenshot, and the 'Databases' and 'Libobjects' entries in the bottom screenshot.

Notes Multiple Databases



An independent disk pool can contain many kinds of objects in addition to database objects. By definition an independent disk pool is created, it will appear as a distinct user database on the server. This is separate from the system database, which was the only database available per system in previous releases.

Independent disk pools

Independent disk pools are used to set up user databases on the iSeries server. There are three types of independent disk pools: primary, secondary, and user-defined file system (UDFS). Databases are set up using primary independent disk pools. When a primary independent disk pool is configured, a new user database is defined that is separate from the system database. Once the independent disk pool is created, it appears as another database under the Database function of iSeries Navigator. The user database also includes any secondary disk pools that are associated with the primary disk pool. By default, the database and the independent disk pool will have the same name. You administer the user database with the same functions that you use for the system database. Multiple databases can be created on 6xx, 7xx and 8xx servers, as long as they are on V5R2.

SETASPGRP cmd

The Set Auxiliary Storage Pool Group (SETASPGRP) command sets the auxiliary storage pool (ASP) group for the current thread. Additionally, this command allows you to change the libraries in the library list for the current thread. If an ASP group had already been set, this command will remove the old ASP group from the current thread and set the specified ASP group for the current thread. Once the specified ASP group has been set for the current thread, all libraries in the independent ASPs in the ASP group are accessible and 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 2-32) 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.

Reclaim storage

With the introduction of IASPs comes the capability to run Reclaim Storage (RCLSTG) on an IASP while the rest of the system keeps running. This implies that multiple IASP RCLSTG processes can execute simultaneously. V5R1 functional changes to the RCLSTG command added to support IASPs are:

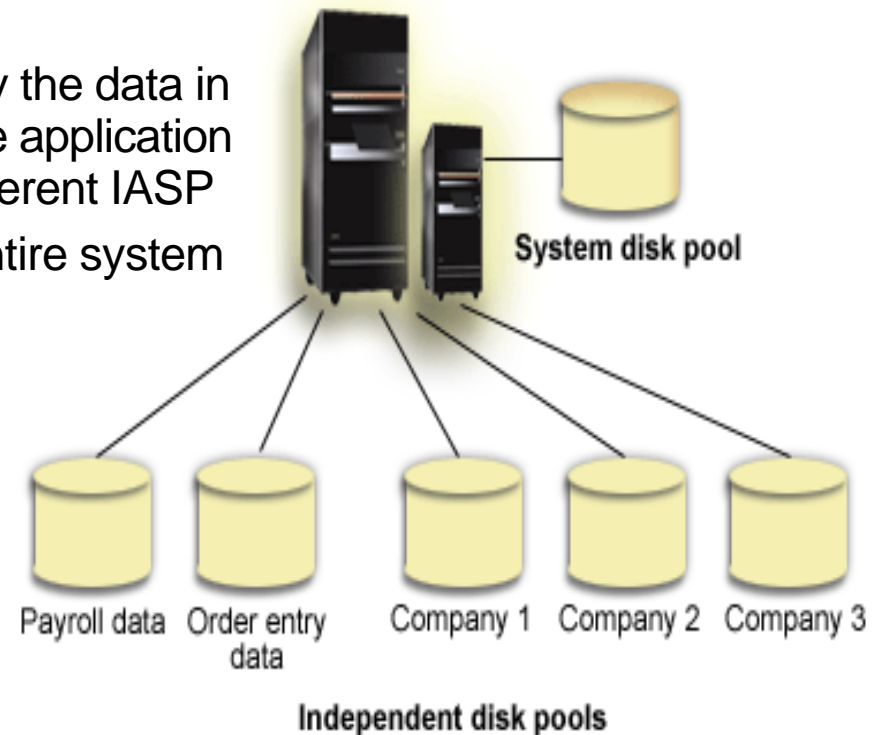
- *SYSBAS values If the *SYSBAS value is specified for the ASP device, the Reclaim Storage operation runs as it does on systems prior to V5R1. The reclaim operation is performed on the system and on basic user-defined ASPs. If the value specified is an ASP device name, then that ASP is reclaimed.
- Reclaim Storage for an ASP device (that is, an IASP) can be run without the system being in restricted state. Multiple ASP devices can be reclaimed in parallel.

Note: Reclaiming an auxiliary storage pool device requires that there are no active users of the ASP device that is the subject of the reclaim. The system will place the IASP in active state which is an intermediate state between unavailable and available.

iASPs in Single System Environment



- Can be any server that supports V5R2
- Isolate low-use data - bring online only when needed
- Reduce system start time
- Manage save/restore by iASP
- Reclaim storage by iASP
- Divide data between multiple databases
- Isolate data
 - if 2 RAID disk failures occur in IASP only the data in that failing IASP becomes unavailable; the application can continue to run against data in a different IASP
- Application maintenance does not affect entire system



Notes iASPs in Single System Environment



The chart shows an example of multiple databases which reside in independent disk pools. Here, we have independent disk pools for the Payroll data, Order entry data and data for Companies 1, 2 and 3. In this example, the actual application code could reside in the System ASP or another disk pool (either a User ASP or another IASP). A typical use of independent disk pools such as shown in this example would be for Server Consolidation of multiple branch office or store systems. Corporate data could reside in the other independent disk pools. Segmenting your databases in this manner can allow greater control and flexibility.

The use of iASPs in a single system environment aids the isolation of low-use data with ability to bring online only when needed. It also reduces system start time within an individual application area as the whole system does not need to be brought down. In current server profiles the whole system must be brought to a restricted state to perform backups. With iASPs you can manage save/restore by independent disk pool. The same applies to Reclaim Storage which can now be run by independent disk pool. Divide data between multiple databases. Isolate data associated with specific applications or associated with specific groups of users. Maintenance also become more granular, as you can perform application maintenance that does not affect entire system.

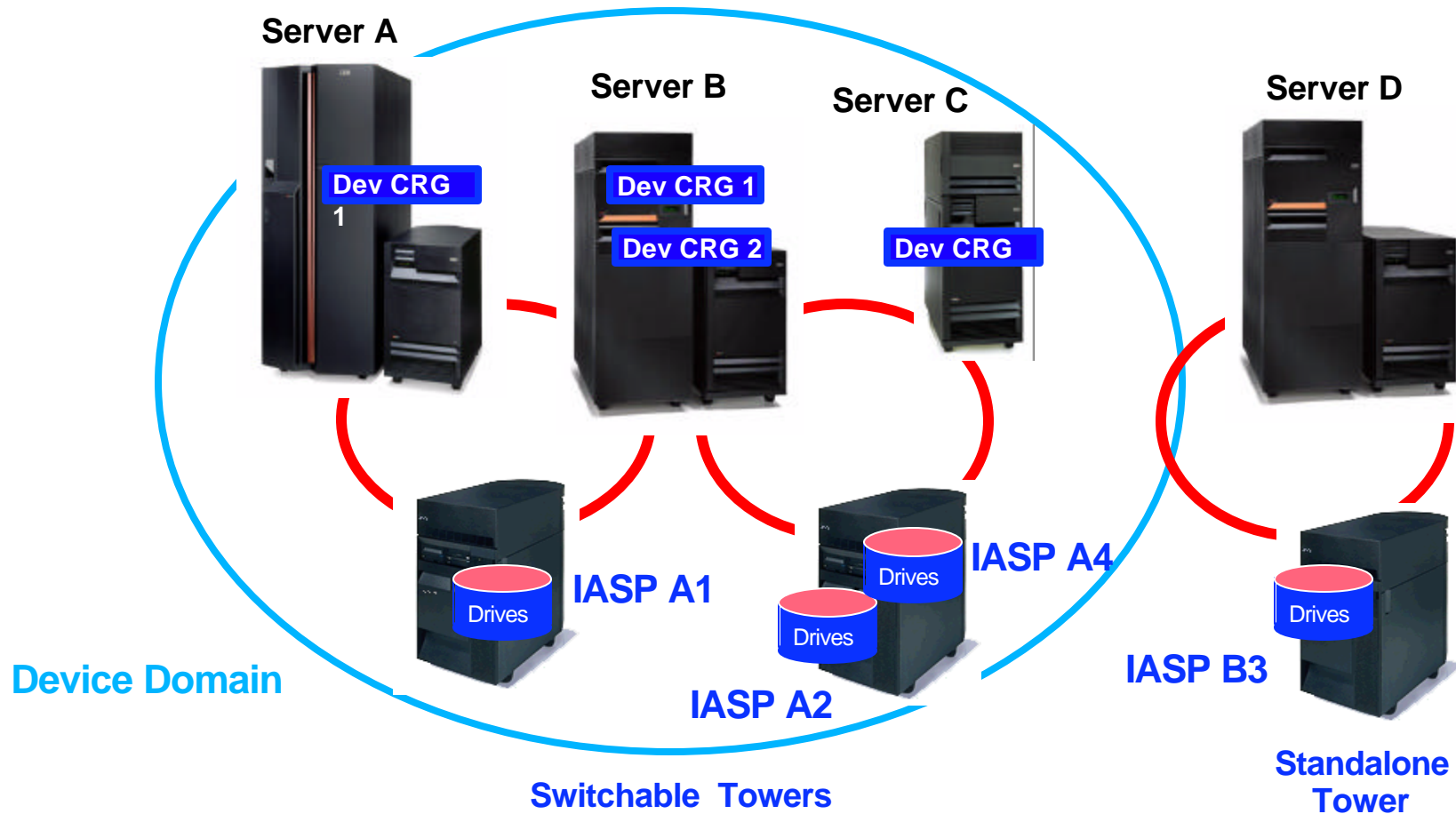
Terminology

ASP = Disk Pool

iASP = Independent Disk Pool

iASP Group = Disk Pool Group

iASPs in Multisystem Environment



Notes iASPs in Multisystem Environment



In a multi-system 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, 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 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 as it allows for continuous availability of data, one of the primary benefits of independent switchable disk pools.

In the chart you can see we have four servers A, B, C and D. There are two switchable towers in the device domain normally owned by servers A and C. These contain iASP A1, A2 and A4. The switchable resources are device CRGs 1 and 2. Device CRG 1 controls iASP A1 and it is switchable between servers A and B.

Servers B and C have two iASPs A2 and A4 in the switchable tower. These are controlled by CRG 2.

Server D is out side the device domain and the iASP is not defined as a switchable resource.

Multisystem iASPs keep data available to an application even in the event of a single system outage, either:

- Scheduled
- Unscheduled

The cluster environment seeks to eliminate the process of replicating data from one system to another. However, this is not as high available as replication based cluster as the switchable disk is a single point of failure..

In some situations the systems will survive failures as iASPdisk units are isolated.

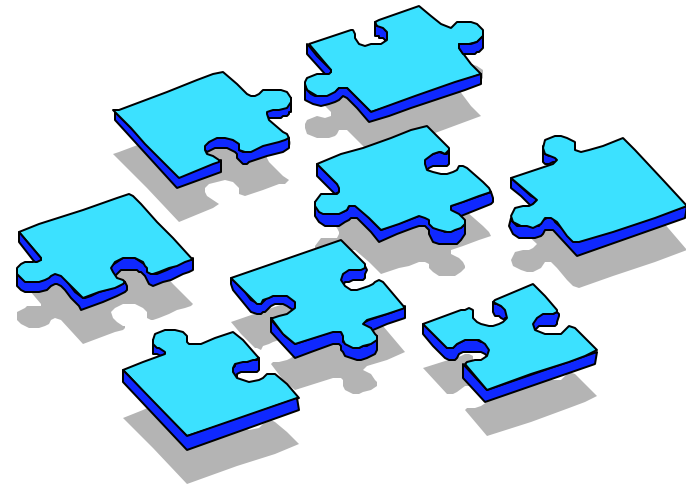


Planning for iASPs

Planning for iASPs



- The primary areas for consideration are:
- Business Needs
- Software Licensing Requirements
- Restrictions
- Application Integration
- Authority Considerations
- Capacity Planning
- Hardware Configuration
- Physical Planning Requirements



For further assistance please see <http://www-1.ibm.com/servers/eserver/series/ha/haplanning.htm>

Notes Planning 1



To implement an independent ASP solution is NOT something that simple. To be more explicit, it can be very complex. Planning is critical !!! The planning considerations below are not intended to be complete since every customer situation vary, but they should give you some starting points.

For further assistance please see: <http://www-1.ibm.com/servers/eserver/iserier/ha/haplanning.htm>

Business Needs

Business needs are quantified, in terms of volumes of data, throughput, response time, etc. When doing the planning for IASP's, keep the business needs close by. In each step of the planning process, the business need should be satisfied before approval and adoption of the plan.

Performance Considerations

Performance is usually one of the metrics of the business needs. Even if it is not, maximizing throughput of the configuration generally requires a small effort with a potentially high return. The following areas should be considered:

- Disk Drives - number of disk arms versus total disk storage capacity. For example a few 35 GB disks may have sufficient capacity but be two few disk arms in an environment that has thousands of disk write operations.
- Placement for data and journals for better performance: The general recommendation to place journal receivers in a separate ASP applies.
- Current name space: iSeries Database support is more efficient if the current name space is the one containing the library and database object referenced by the currently active application program. The use of a job description to specify the initial IASP or the SETASPGRP command can be used to establish the current name space as shown earlier in this presentation. Otherwise, for example, an SQL CONNECT statement will use the Distributed Relational Database support within the same system. This is faster than actually accessing a remote system database, but is a longer code path through the system. This is also discussed in other foils on Multiple Database Support - Setup Details

Software Requirements

System ASP's and User ASP's are supported under all releases. UDFS ASP's are supported under V5R1. Primary and Secondary ASP's are supported under V5R2.

Notes Planning 2



Application Integration

On V5R1, only IFS supported objects are supported in an IASP. On V5R2, the list of supported and unsupported objects can be found on the iSeries Information Center. 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, and other objects common to the application area are stored in system libraries, or libraries designated as common to that application. User profiles, authorization lists, job queues, and spool output queues are some of the object types that cannot exist in an IASP. These are in system libraries at application installation time in most cases. However, consideration should be given to how these are replicated, saved, restored, and/or used in light of Independent ASP's.

Authority Considerations

User profile information is stored in the system ASP. Each user profile object is an object type of *USRPRF. Copies of *USRPRF objects are not in any independent pool. However, some user profile information must be maintained on the IASP itself. Additional storage (above that consumed by objects) is required for these system security structures. This is necessary to make the independent ASP self-contained. When creating independent disk pools in a clustered system environment, it is assumed the content of any user profiles is synchronized across the cluster in which the user profile exists.

Capacity Planning

Consider the amount of disk required, the performance requirements, then the amount of rack space required, then the switchability of the config, then the amount of floor space required. Sizing the disk pool All IASP's or Disk Pools within the rack will be 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) will be switched as well.

Hardware Configuration

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 into the planning the following items:

Physical vs. Logical Switching

Remember that when you create an IASP with disks that are part of the same parity protection set they might have parity stripes on them, so plan to select the correct drives to meet your capacity requirements.

Notes Planning 3



Physical Planning Requirements

There is a lot to be considered in respect to physical planning. Except for the "normal" physical requirements like floor layouts and power requirements, things like HSL cable placement and SPCN cable placement are can be complex.

For placement and cabling assistance, see the following URL: <http://www.as400.ibm.com/tstudio/planning/hslguide.htm>

Building a drawing of the desired configuration, especially the towers and racks involved, can often save costly errors when assembling the configuration upon delivery, especially in the area of cables. Design it one way, assemble it another way, and, the end result may be the need for additional cables.

The next set of foils address disk and other I/O device considerations within an IASP with examples.

Object Management

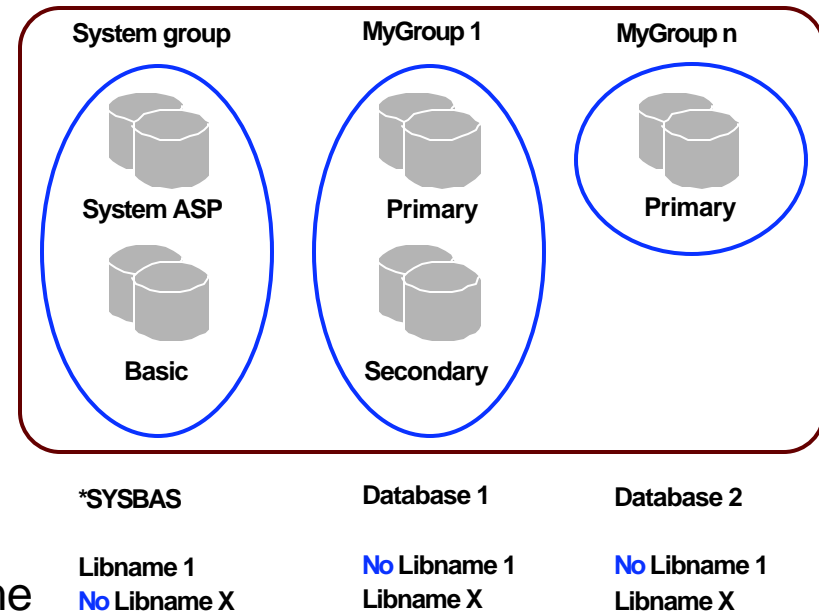


Library naming rules

- Multiple occurrences in multiple iASPs
- Libraries in an IASP cannot be in SYSBAS

Object management

- Most commands that move objects are enabled (save, restore, ...)
- New Job Description parameter - set initial ASP Group
- Most commands default to the current name space (no ASPGRP parameters)
- SETASPGRP needs to be done
 - Or use the JOBID linked to the user profile



IASPs are not available after an IPL

- Must be explicitly varied on/made available by an operator

Notes Object Management



Library names

With the independent ASPs it is now possible to have multiple databases. In a pool group there can reside only one primary independent ASP and zero or more secondary independent ASPs. The primary and secondary disk pools share the same database since they share the same database name space. So you could say that a pool group is equivalent to a database.

The same library name (or SQL collection name) can exist in two different independent disk pool groups though the same name cannot exist in the system disk pool. (ASP1).

Object management

Because the existence of an independent disk pool on a server implies that multiple databases will exist on a single server, identifying an object requires more complex planning than on a system with only a single system database. When multiple databases exist, it is possible to duplicate the names of libraries and objects in separate databases. The library name and object name don't necessarily uniquely identify an object. There will be times when you'll also need to know the name of the independent disk pool. The name of the independent disk pool and its database are, by default, the same. However, they don't necessarily have to match. A database name in DB2 UDB for iSeries can be up to 18 characters long, while an independent disk pool name can be up to 10 characters long. The SETASPGRP command points the current job/thread to any pool group.

iASPs are NOT automatically available after an IPL

Independent disk pools are available only after you have varied them on / made them available (iSeries Navigator term). They are not made available during a normal restart of your server, unless you include code in your Start Up Program to make them available. When you select to make a disk pool available, the disk pool goes through a process similar to that of a server restart (IPL). While this processing takes place, the disk pool cannot be used with the application.

V5R2 Supported Objects



*ALRTBL	*FIFO	*MODULE	*QRYDFN
*BLKSF	* FILE	*MSGF	* SBSD
*BNDDIR	*FNTRSC	* MSGQ	*SCHIDX
*CHTFMT	*FNTTBL	*NODGRP	*SPADCT
*CHRSF	*FORMDF	*NODL	*SQLPKG
*CLD	*FTR	*OVL	*SQLUDT
*CLS	*GSS	*PAGDFN	*SRVPGM
*CMD	*IGCDCT	*PAGSEG	*STMF
*CRQD	*JOB	*PDG	*SVRSTG
*CSI	* JRN	*PGM	*SYMLNK
*DIR	* JRNRCV	*PNLGRP	*TBL
*DSTMF	*LIB	*PSFCFG	*USRIDX
*DTAARA	*LOCALE	*QMFORM	*USRQ
*DTADCT	*MEDDFN	*QMQRV	*USRSPC
*DTAQ	*MENU		*VLDL
*FCT	*MGTCOL		*WSCST

Notes Supported Objects



This chart is pretty self explanatory.

A special case is Subsystem descriptions (SBSD). These are supported in an iASP, but the SBSB cannot be active. This means the SBSB is located in the iASP as an archive.

V5R2 Objects Not Supported



- *AUTHLR
- *AUTL
- *CFGL
- *CNL
- *COSD
- *CRG
- *CSPMAP
- *CSPTBL
- *CTLD
- *DDIR
- *DEVD
- *DOC
- *EDTD
- *EXITRG
- *FLR
- *IGCSRT

- *IGCTBL
- *IPXD
- *JOBQ
- *JOBSCD
- *LIND
- *MODD
- *M36
- *M36CFG
- *NTBD
- *NWID
- *NWSD**
- *OUTQ**
- *PRDAVL
- *USRPRF**
- *SOCKET
- *SSND
- *S36

Notes Unsupported objects



In this chart there are a number of very significant objects that are not supported.

JOBQ - job queues are not supported in a iASP. This can be a significant point for applications moving to iASPs. An application will need to be modified to overcome this problem.

NWSD - Network storage descriptions are less of a problem as they are a slowly changing object. They are only really changed if their resource changes or there is an additional drive linked. A way of keeping the nwsd current in a switch environment is to save the nwsd into a savefile and store this in the iASP. Then on switch over restore the nwsd before the windows or linux server is varied on.

OUTQ - This is probably the most significant object that is not supported. Effectively printed output will not be stored in the iASP. At a future release of OS/400 OUTQs will be supported. There are options to overcome this lack of support. One alternative is to use Infoprint Server. The OUTQ remains in SYSBAS, but the spool file is immediately converted to a PDF and then stored in the IFS. This enables the printed output to be switched with the iASP. Another option is to convert the output to a database file and then spool it off. This database file can then be located in an iASP.

USRPRF - A later chart discusses user profile support.

How to structure iASPs



For best possible single system protection and performance

- Most application **database objects** into IASPs
- Minimal number of **database objects** in SYSBAS
 - Primarily operating system objects, LPP libraries, few user libraries
 - Depending on customer requirement, place all application programs in SYSBAS or in IASP along with database objects
- Recommended: Use disks in a separate RAID group for each IASP

Notes How to Structure iASPs



The recommended usage structure for independent disk pools is to place the majority of your application database objects into independent disk pools and a minimal number of database objects in SYSBAS, which is the system disk pool and all configured basic disk pools. The system disk pool and basic user disk pools (SYSBAS) would contain primarily operating system objects, licensed program product libraries, and very few user libraries. This structure yields the best possible protection and performance. Application data is isolated from unrelated faults and can also be processed independently of other system activity. Vary on and switchover times are optimized with this structure. Other advantages of this structure are:

- Since a database network cannot span an independent disk pool boundary, entire database networks (SYSBAS and an IASP) are contained within disk pool groups.
- Library names can be duplicated across disk pool groups, but not between a disk pool group and the libraries in SYSBAS. Applications can access different databases with little or no changes. As discussed later in Multiple Database Support Setup, you can see the ways impact to applications programs can be zero or minimal.

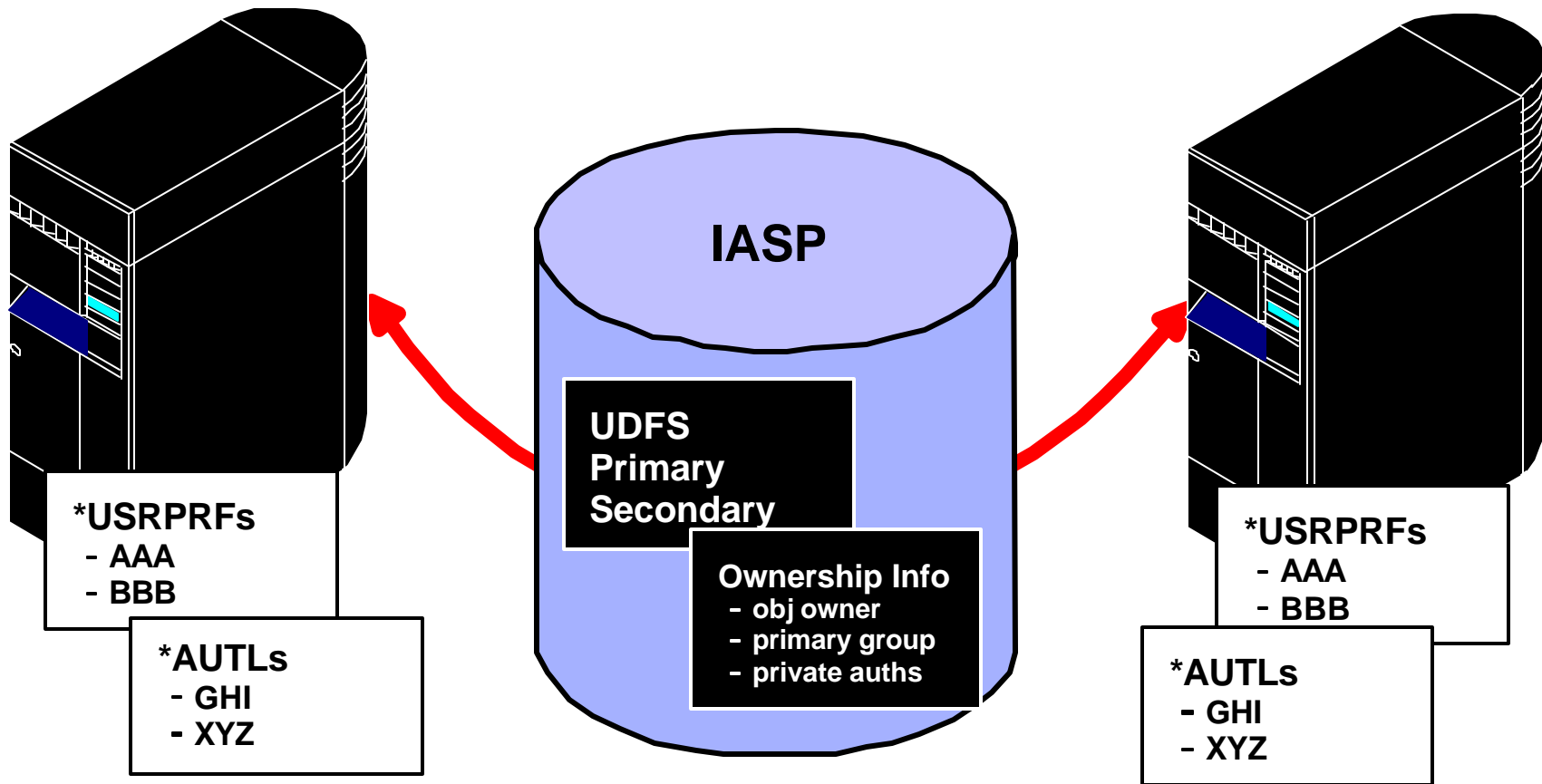
Although the above is the recommended structure, this does not exclude other configurations.

For example, you may start by migrating only a small portion of your data to a disk pool group and keeping the bulk of your data in SYSBAS. However, you should expect longer vary on and switchover times with this configuration since additional processing is required to merge database cross reference information from SYSBAS with the database reference information in the independent disk pool group.

Merging database cross reference information

At the time an independent pool is made available, the information of the SYSBAS database cross reference information is merged into the database cross reference information of the independent pool. The information in the database cross reference information of the independent pool is not merged into the database cross reference information of the system ASP. Keeping the system ASP cross reference information small enables faster vary-on, faster switchover of the independent pools and shorter IPL times.

User profiles



Notes User profiles



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

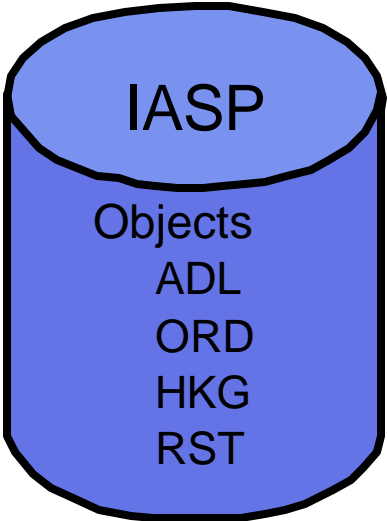
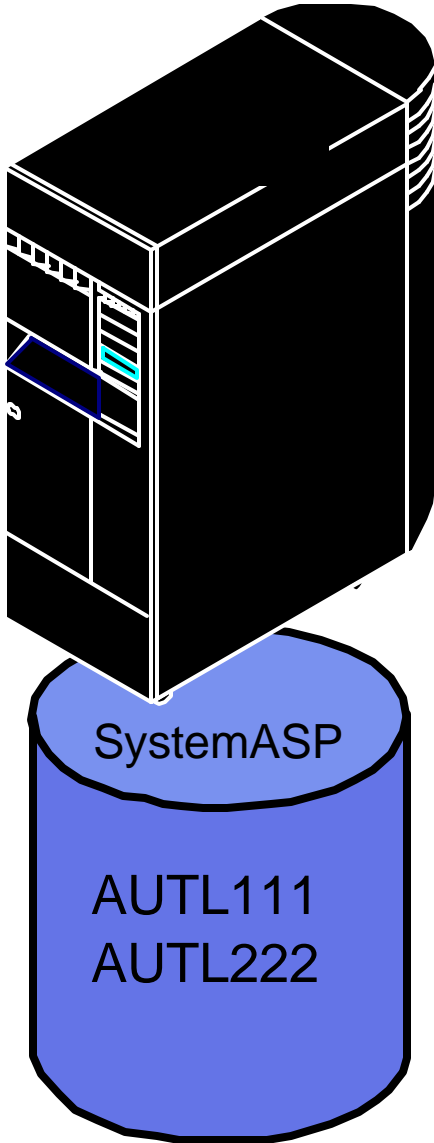
Additional storage (above that consumed by objects) is required for these system security structures. This is necessary to make the independent ASP self-contained. These structures consume disk space within the IASP. The percentage varies and depends on the size and number of objects referenced by these 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”. These user profile extensions, while not visible nor accessible to users, are stored in the IASP.

Anytime an IASP is varied on, whether it is due to a simple VRYCFG command, an IASP fail over, or being switched from another node, the system matches up 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.

Authorization lists



Objects not secured by authorization list

- ADL
- ORD

Objects secured by AUTL111

- HKG

Objects secured by AUTL222

- RST

Notes Authorizations lists



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.

This concept is illustrated in the chart. Authorization lists AUTL111 and AUTL222 are stored in the IASP, because the objects HJK and RST are secured by these lists. There is no reference on the IASP itself to the authorization list AUTL333 because the object LHR is not stored on the IASP.

As with user profiles, the 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.

iASP Migration



iASPs created on a R510 system that is upgraded to R520

- iASP hardware can be switched back and forth between R510 and R520 systems.
- Once VRYCFG has been used to Varyon an iASP on a R520 system, VRYCFG can no longer be used to Varyon the iASP on a R510 system.
 - This is true even if the iASP contains only UDFS's.
- This process will take some time to complete, as objects must be converted to the new release level at the first Varyon.

Notes iASP Migration



IASPs created on a R510 system that is upgraded to R520

- IASP hardware can be switched back and forth between R510 and R520 systems.
- Once VRYCFG has been used to Varyon an IASP on a R520 system, VRYCFG can no longer be used to Varyon the IASP on a R510 system.
 - This is true even if the IASP contains only UDFS's.
- This process will take some time to complete, as objects must be converted to the new release level at the first Varyon.



Disk Cofigurations

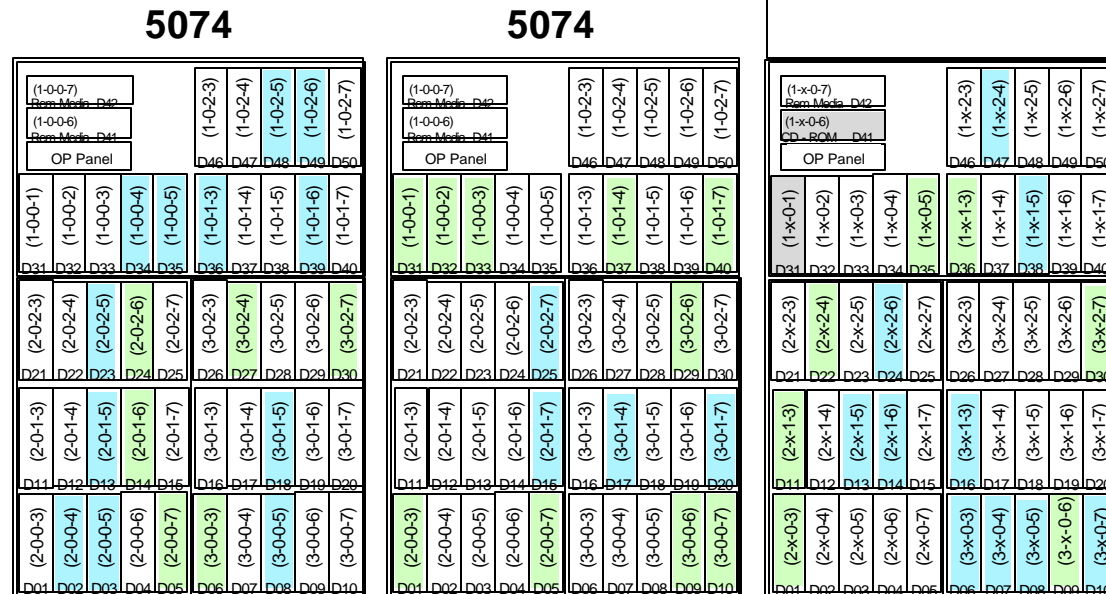
Non-Switchable Independent Auxiliary Storage Pool



- Select any Disk Unit(s) except the Load Source Unit
- Implications with random (unplanned) selections
 - Protection
 - Recovery

 IASP 33

 IASP 34



Notes Non-switchable iASP



Creating a non-switchable Independent Auxiliary Storage Pool (IASP) for use on a single system is intended for, but not limited to, support of multiple databases on the same system, as described in this presentation and the Database presentation. Selecting disks for this IASP can be done without taking any rules into account ("random or unplanned selection") - except for the fact that the load source of the system is required to be in the system ASP.

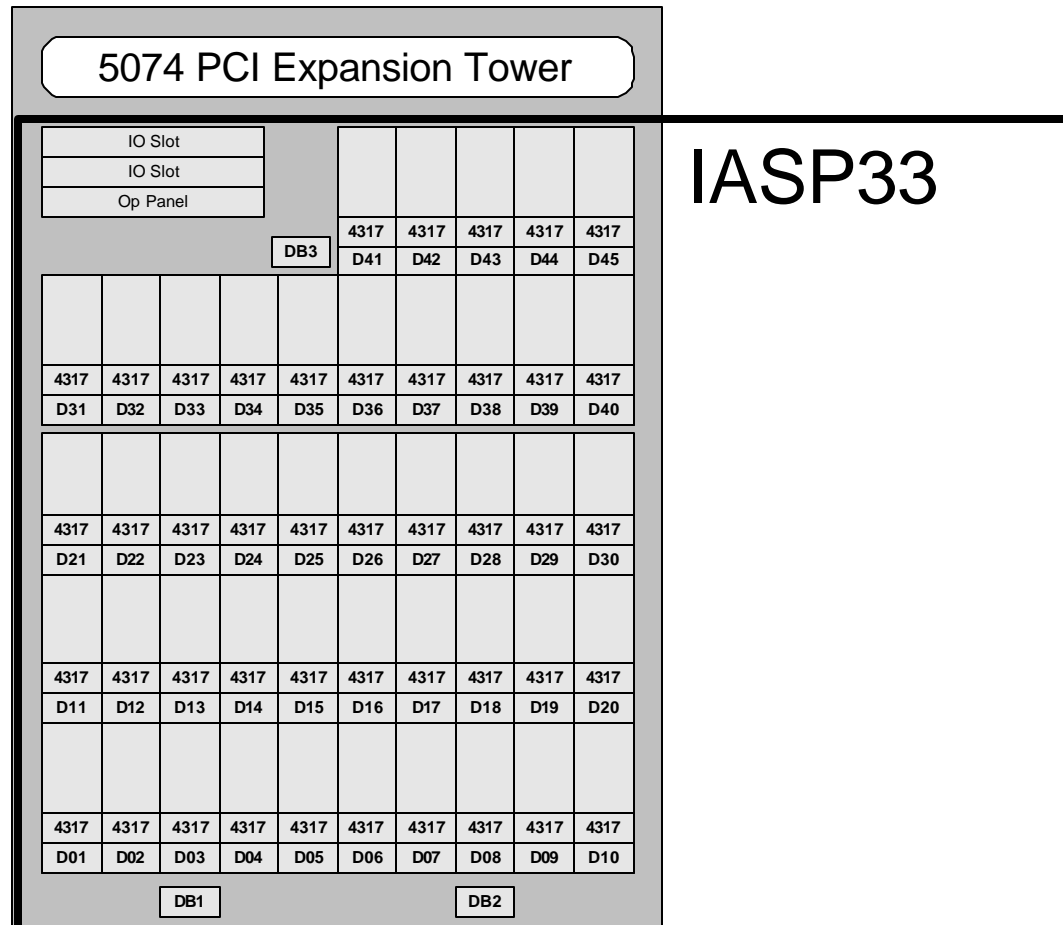
Although, not required, we recommend some logic be applied in determining which disks are placed into an independent ASP. When one understands the mechanics and rules of RAID5 protection and the physical I/O behavior using this protection method, it makes sense to select one or more Raid5 sets to form a group of disk units to be configured into the IASP. The high reliability of Raid5 protection with the capability of concurrent replacement and rebuild of a failing unit in a RAID5 set, makes the reload of data due to a hardware failure a very rare event. However, in the exceptional case that two disks in a single RAID5 set would fail, the alignment of an ASP (whatever type) with one (or more) RAID5 set(s) will prevent data loss in the system ASP or any other ASP.

Starting mirroring in an ASP (any type) is easy. However, the system licensed internal code will automatically obtain the highest protection level possible with the disk units selected for a particular ASP when mirroring for that ASP is started. Disk units must be selected depending on the level of mirroring protection that needs to be achieved: Disk level, IOP level, Bus level or Tower level.

For acceptable performance, the appropriate number of disk arms should be selected to configure the ASP. The number of disk arms in an IASP can influence the performance behavior of an application using the data in that IASP. Mixing disk sizes in a single ASP can also effect the disk I/O behavior. Therefore, it is recommended to use the same type and size of disk units when configuring disk units in an ASP. Correct sizing must be done based on an accurate interpretation of performance measurements.

For example, you could review disk activity over time with the WRKDSKSTS command or viewing the Disk Resource section of the System Report available with Performance Tools for iSeries, 5722PT1, to understand which disks are the most busy in your application environment.

iASP config 1 - all units

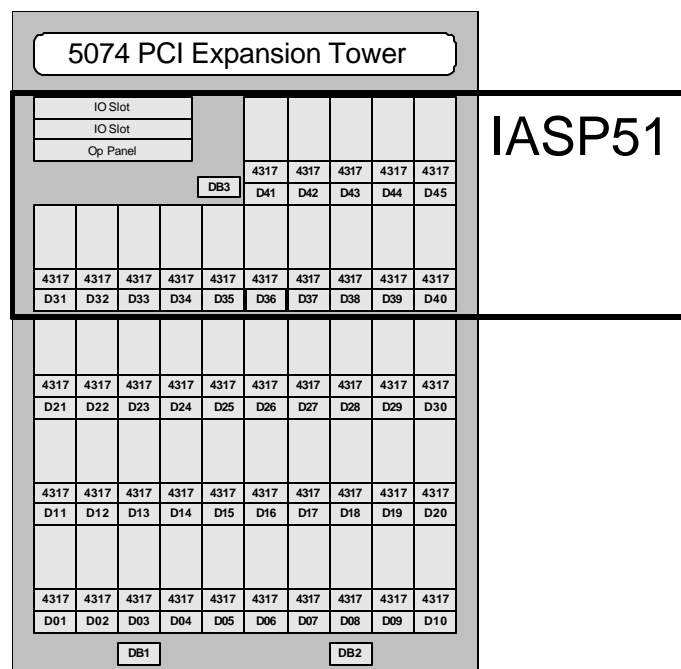


Notes iASP config 1 - all units



In this configuration all the disk in the tower are configured as one iASP. If this tower was switchable the disk and all accomodated IOPs/IOAs would also switch.

iASP config - subset of disk



Notes iASP config - subset of disk

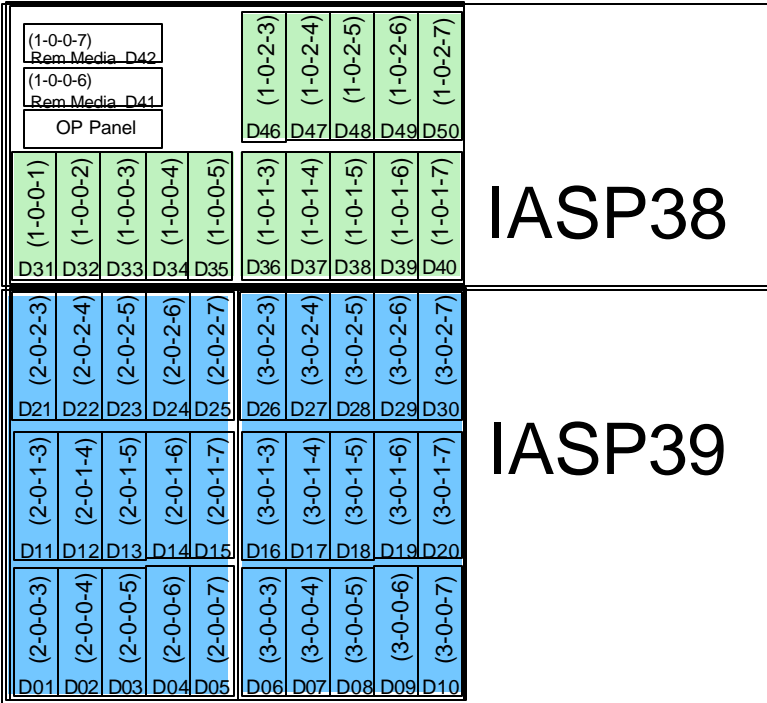


In this configuration iASP 51 is configured with 15 drives in the tower. This would be one complete RAID set within the tower. In this configuration the disks could not be switchable between systems as the remaining disk are not defined as switchable resources.

iASP config - multiple iASP in a tower



5074

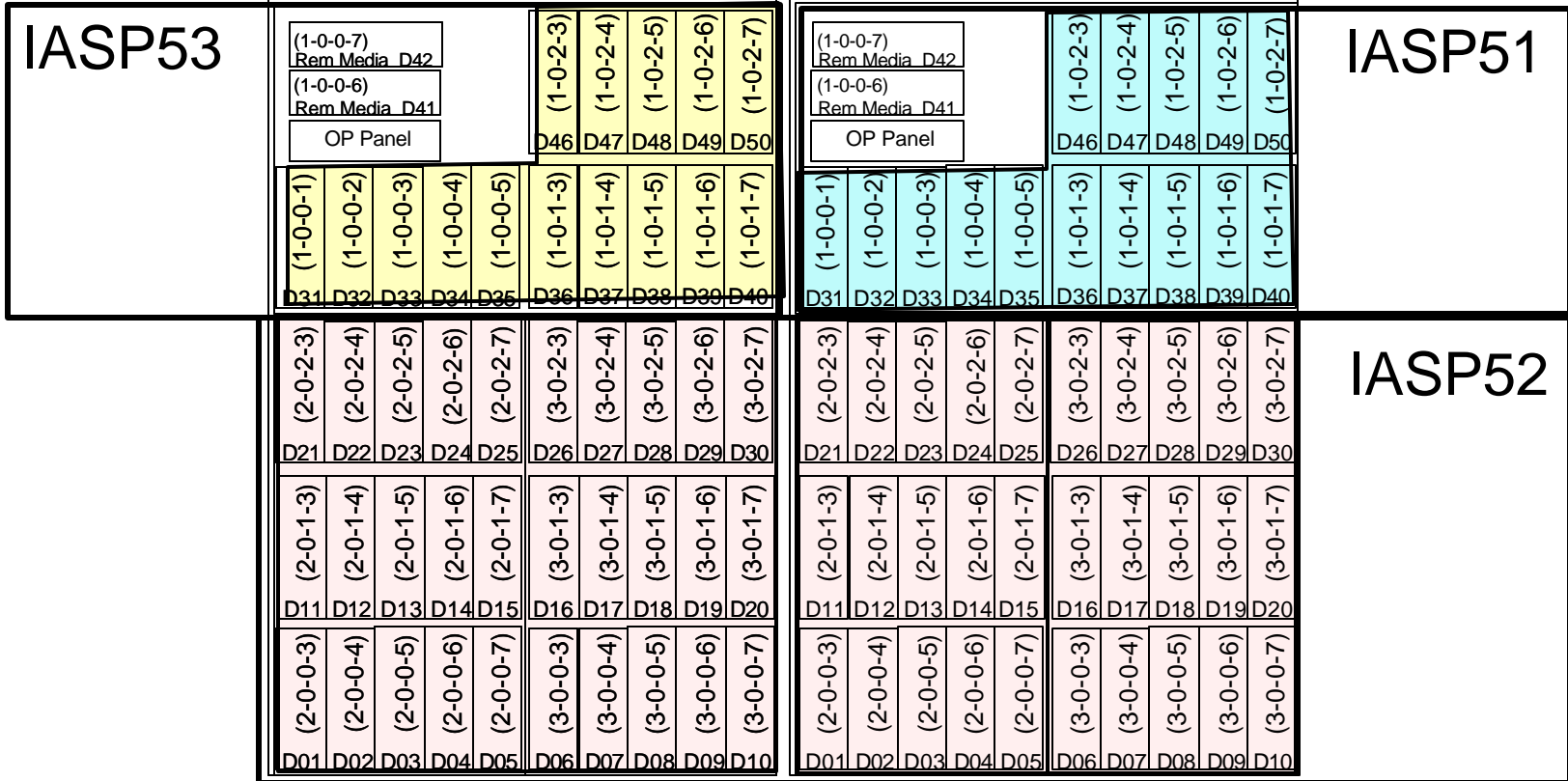


***Notes* iASP config - multiple iASP in a tower**



In this scenario all disks are configured into two iASPs, 38 and 39. These disks could be made switchable, but they would have to switch together.

iASP config - disk units in multiple towers



Notes iASP config - disk units in multiple towers



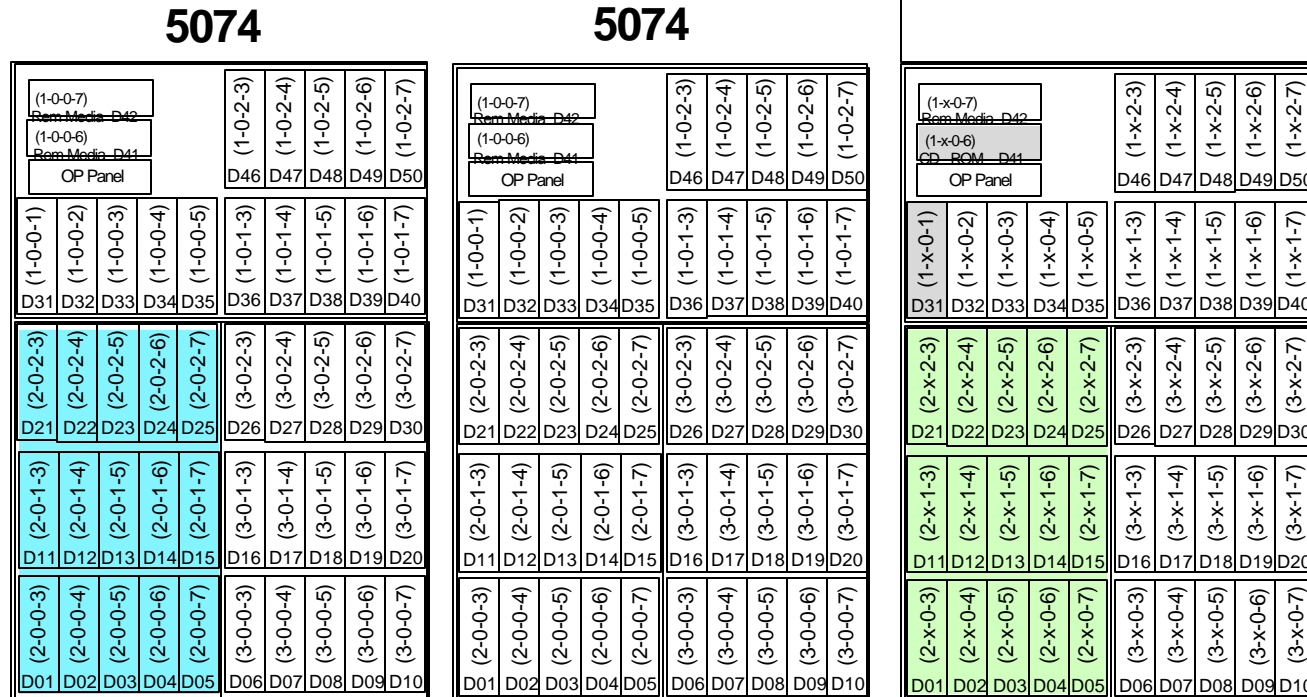
Now we have increase our storage to two towers. This example shows that all disks are allocated to independent ASPs. These disks could be defined as switchable resources, and must all switch together.

Switchable IASP with LPAR



- Cluster requirement between partitions
- Select Disk Units on IOP level
- Shared bus rules
- Size Ratio of *SYSBAS to iASP

IASP 33
IASP 34



Notes Switchable iASP with LPAR



Switchable Independent Auxiliary Storage Pools can be used with a cluster of partitions within a single system. The configuration of that type of switchable IASP can only be done at the IOP level. This is simply because the IOP is the lowest possible hardware level that can be removed from or moved (switched) to partitions that share the bus the IOP resides in. This rule implies that if a number of adapters or disk controllers are assigned to a certain IOP and that IOP is removed from partition A and moved into partition B, all associated assigned adapters and controllers move together with that IOP.

Placement or assignment of disk controllers that drive the disk units configured in a switchable IASP must be carefully planned. There are a number of other considerations such as bus ownership in the logical partition setup, cluster rules, disk controller capabilities and more that need proper investigation before configuring a switchable IASP.

For performance, the same rules and considerations as for the non-switchable IASP apply. In addition to these considerations, the way the switchable IASP will be used on the clustered nodes has to be considered.

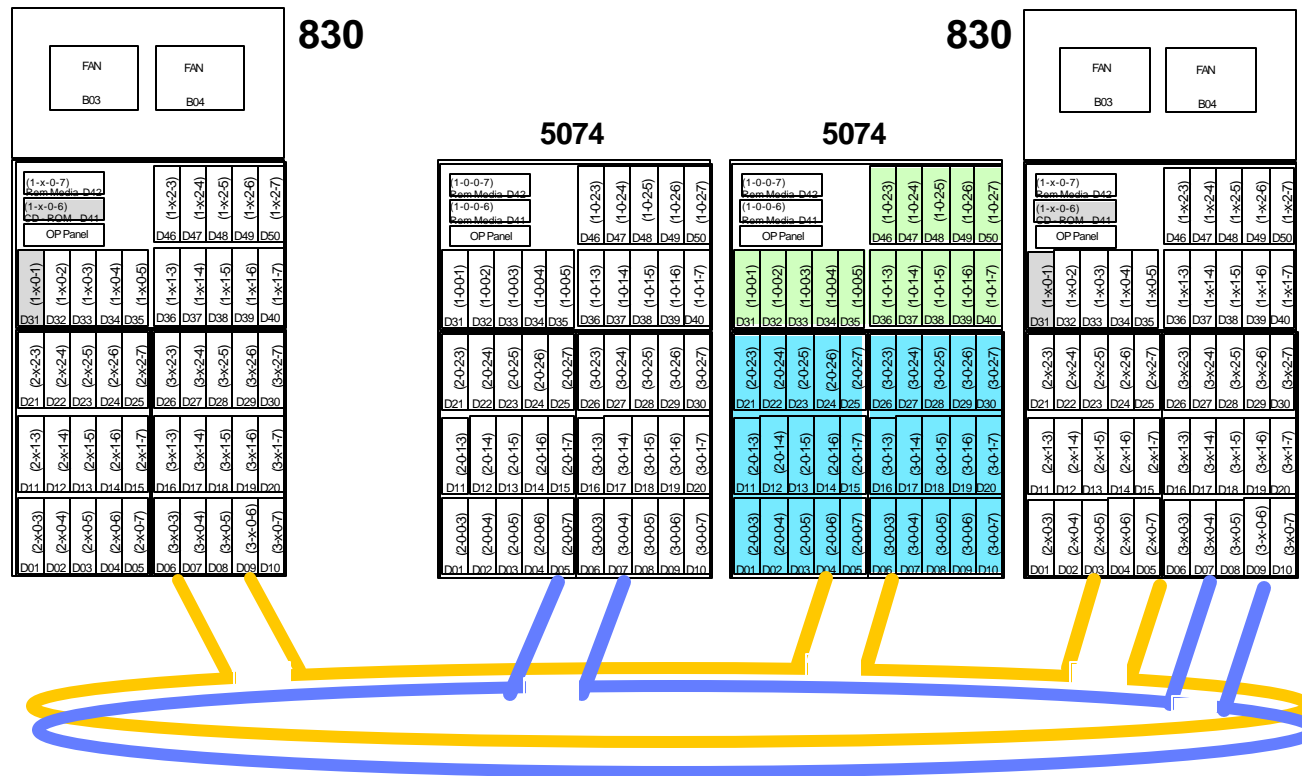
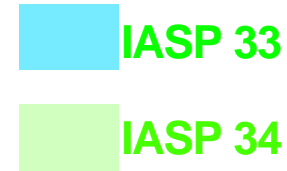
As described elsewhere in this presentation, the vary-on time for the IASP device can be considerably reduced by minimizing the number of database objects in the *SYSBAS environment. Recall that SYSBAS means the system ASP and any user (dependent) ASPs (ASP 1 to 32). This is because the cross reference of the *SYSBAS environment (primarily database objects and object authorities) must be propagated into the IASP that is brought online. The necessary checking must be performed and ownership and authorization information, called user profile extensions, must be written into the IASP.

In a clustered environment, after the IASP has been switched, the time needed to make the data in a Switchable Independent Auxiliary Storage Pool available to a cluster node is directly impacted by the way the applications and data are distributed over the *SYSBAS ASPs and the "newly arrived" IASP. That is, if there are a large number of database objects in SYSBAS on the switched to system and several database objects in the "newly arrived" IASP, then the vary on (make available) process of this IASP on the target system could be longer than expected.

Multiple server implementation of switchable IASP



- Cluster configuration requirement
- Select disk units on Tower level
- Respect switchable tower rules
- Number of disk units consideration



Notes Multiple server implementation of switchable IASP



Configuring a Switchable Independent Auxiliary Storage Pool for use on two clustered iSeries servers is essentially the same as configuring the switchable IASP for use on two clustered iSeries partitions on a single iSeries server.

The only additional consideration here is that the IASP needs to physically reside in one or more switchable HSL towers. The HSL tower is considered as the lowest switchable hardware level resource that can be switched between two iSeries clustered servers. The complete tower(s) is switched from one clustered iSeries server to the other. Therefore, in this example configuration both IASP33 and IASP34 would be switched between systems.

Note also, all hardware resources contained in the tower are brought over as well as the IASP resources. To use the hardware resources other than the IASP hardware, a configuration object for each of these hardware resources must exist.

The HSL rules for clustered iSeries servers must be respected. Only four I/O towers can connect to a clustered loop with two iSeries servers and the maximum of towers is three per HSL segment. All other aspects mentioned in the two previous foils should be considered also.

We did not cover IASP configurations on external DASD (ESS) attached to the #2766. The rules are essentially the same. Since a #2766 requires a dedicated IOP, all LUN's attached to the #2766 will be moved together with the IOP the #2766 is assigned to. Moving a tower containing multiple #2766 adapters will of course move all associated hardware attached to and contained in that tower to the other iSeries cluster node.

It is also possible to "randomly" configure IASPs with a selected number of LUN's that are made available for use on a single iSeries server.



Operating iASPs

New iASP setup

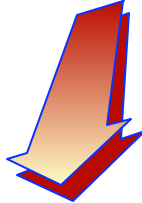


5250 workstation

CRTDEVASP DEVD(MYCPY1)
RSRCNAME(MYCPY1) RDB(*GEN)



- Go to SSP to add unconfigured disks to this IASP*



- ADDRDBDIRE entry added by the system.
- In iSeries Navigator "make available" or in OS/400 "vary on;" populate with library with objects; the library content is then available to applications.
- Repeat for each IASP<-> Database required.

iSeries Navigator Configuration and Service -> Hardware -> Disk pools - > New Pool *

New Disk Pool

Type of disk pool: Primary

Disk pool: MYCPY1

Database: Generated by the system

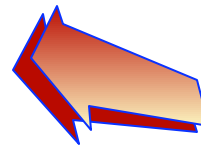
Note: If you want to create a switchable disk pool, be sure to use the appropriate clustering function before using this wizard.

Protect the data in this disk pool

OK Cancel Help ?



- Continue in iSeries Navigator to add unconfigured disks to this IASP



* System Service Tools signon required

Notes New iASP setup



This is an overview of the process to create an Independent ASP being used to support multiple databases on the same system. In our example, as shown on the following foils, we use 3 IASPs that are named Mycpy1, Mycpy2, and Mycpy3.

We show examples of creating a new IASP from either a 5250 command interface or an iSeries Navigator interface. We do not show the steps to add unconfigured disks to an IASP but do note authority to use System Service Tools (SST) is required to add the disks to the IASP. V5R1 introduced this SST sign-on requirement for disk management. For a user not familiar with the SST functions, we recommend using the iSeries Navigator interface to create the new pool and assign disks to that pool.

In the example shown we are defaulting to the database and the IASP having the same name. When you define a new IASP and associated database name, the system implicitly invokes the Add Relational DataBase Entry (ADDRBDIRE) function.

iASP Library management



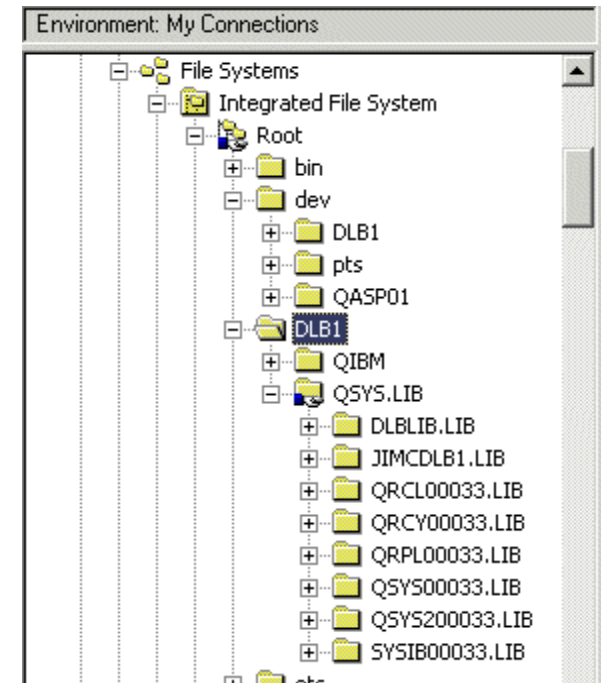
OS/400 commands:

- CRTLIB LIB(JIMCDLB1) ASP(*ASPDEV) ASPDEV(DLB1)
- WRKLIB/DSPLIB LIB(*ALL) ASPDEV(*ALLAVL , ASPname, SYSBAS, *ASPGRP,)
- SAVLIB, RSTLIB, ...

DSPLIB *ALL *ALLAVL example

Opt	Library	Type	ASP	Device
	DLBLIB	PROD		DLB1
	JIMCDLB1	PROD		DLB1
	QRCL00033	PROD		DLB1
	QRCY00033	PROD		DLB1
	QRPL00033	PROD		DLB1
	QSYS00033	PROD		DLB1
	QSYS200033	PROD		DLB1
	SYSIB00033	PROD		DLB1
	#LIBRARY	PROD		
	#RPGLIB	PROD		
	ISALIB	PROD		
	MRK	PROD		

Operations Navigator - IFS



Notes iASP Library Management



There are standard parameters that allow you to work with iASPs.

CRTLIB LIB(JIMCDLB1) ASP(*ASPDEV) ASPDEV(DLB1)

- Create Library, the following parameter work with iASPs
- ASP Name:
 - The parameter ASP, specifies the auxiliary storage pool (ASP) from which the system allocates storage for the library. For libraries created in a basic user ASP, all objects in the library must be in the same ASP as the library. When a value other than *ASP is specified for the ASP device (ASPDEV parameter), *ASPDEV is the only valid value for the ASP number (ASP parameter), if specified.
 - 1- The storage space for the library is allocated from the system auxiliary storage pool ASP 1.
 - *ASPDEV - The storage for the library is allocated from the ASP specified for the ASP device (ASPDEV parameter).
 - auxiliary-storage-pool-number - Specify a value ranging from 1 through 32 to specify the number of the system or basic user ASP.
- ASPDevice
 - *ASP The storage for the library is allocated from the ASP specified for the ASP number (ASP parameter).
 - *ASPGRPPRI The storage for the library is allocated from the primary ASP of the thread's ASP group. If no ASP group is associated with the thread an error message will be issued.
 - *SYSTEM The storage for the library is allocated from the system ASP (ASP 1).
 - auxiliary-storage-pool-device-name The storage for the library is allocated from the primary or secondary ASP. The primary or secondary ASP must have been activated (by varying on the ASP device) and have a status of 'Available'.
 - ASPDEV(*ALLAVL , ASPname, SYSBAS, *ASPGRP,)

WRKLIB/DSPLIB LIB(*ALL)

- ASP Device The possible values are:
 - *ALL *ALL will include all the libraries in the thread's library name space depending of the value defined in the ASP device (ASPDEV parameter).
- ASP Name
 - auxiliary-storage-pool-number Valid values are the numbers 1 through 32, which must designate an ASP that is configured on the system. ASP 1 is the system ASP. The user ASPs are 2-32. For information on configuring an ASP, see the Backup and Recovery book, SC41-5304.

SAVLIB, RSTLIB, are mention later in this section



Backup and Recovery of iASPs

Saving iASPs



OS/400 SAVxxx and RSTxxx commands

- Use the SETASPGRP command or
- SAVxxx ASPDEVparameter
 - *, *SYSBAS, *CURASPGRP, ASP device name
 - Special considerations when saving to a save file

On a full-system save (Option 21) or a save of all user data (Option 23)

- Independent disk pools are saved alphabetically
- Secondary ASPs are saved along with their primary

Notes Saving iASPs



The native OS/400 SAVxxx and RSTxxx commands have been enhanced to provide support for IASPs. Using these commands in your own CL programs to backup the system is relatively straightforward, as you are in control of the environment when these are running. In general, these native commands must have access to the namespace where the objects to be saved reside. This can be achieved by using the SETASPGRP command prior to issuing the Save command or by using the new with V5R2 ASPDEVparameter on the SAVxxx and RSTxxx commands.

If you have grasped 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 will 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.

Special considerations on save commands

The ASPDEV parameter will allow you to save the independent ASP or object in SYSBAS and, if active in the job/thread, an IASP - without changing your job thread, depending on the value you use for this parameter. If you are saving to save files, this parameter does not affect the DEVICE parameter of the save commands. You must use the SETASPGRP command if you are saving to a save file that exists in an independent ASP. This also allows you to save to a save file that exists in a different independent ASP than the one you are saving. Rather, the ASPDEV parameter acts as a filter on the SAV command. The following is an example:

There is a file called MOLN and a save file called CORZ in a library called BER in an IASP called ROCHESTER.
There also is a save file called ADMS in library QGPL.

The following command **will not work unless** SETASPGRP has been previously run:
SAVOBJ OBJ(MOLN) LIB(BER) DEV(*SAVF) SAVF(BER/CORZ)

Even had we added the ASPDEV such as in the following command, it still will not find the save file.
The following command **will not work**:
SAVOBJ OBJ(MOLN) LIB(BER) DEV(*SAVF) SAVF(BER/CORZ) ASPDEV(ROCHESTER)

Continued.....

Notes Saving iASPs 2



To save the file MOLN to the save file CORZ, the SETASPGRP command has to be used.

The following is the correct sequence to use the save file:

```
SETASPGRP ASPGRP(ROCHESTER)
SAVOBJ OBJ(MOLN) LIB(BER)DEV(*SAVF) SAVF(BER/CORZ)
```

Here is a quick summary of the supported values for ASPDEV parameter using Save Library command as an example:

- *: Saves a library located in the system ASP (ASP number 1) and in any basic user ASPs (ASP numbers 2-32). If the current thread also has an ASP group, all independent ASPs in the ASP group will also be saved.
- ***SYSBAS**: Saves a library located in the system ASP (ASP number 1) and in any basic user ASPs (ASP numbers 2-32).
- ***CURASPGRP**: If the current thread has an ASP group, saves a library in all independent ASPs in the IASP group are included in the save operation.
- **auxiliary-storage-pool-device-name**: saves a library in the specified IASP group.

Notes Saving iASPs 3



You can save your independent ASPs as part of a full system save (GO SAVE: Option 21), or when you save all user data (GO SAVE: Option 23). You can also use BRMS to save your independent ASPs.

In either case, you must make the independent ASPs available before you perform the save. You can also just save the current ASP group. Saving the ASP group saves both the primary ASP and any associated secondary ASPs. If you make independent ASPs available, they will be included in an Option 21/option 23 save. Before you end subsystems and restrict your server, make sure that your current job does not use integrated file system objects in the independent ASP and do not perform a SETASPGRP cmd.

The following scenarios show examples of using different options to save a library or object within an independent ASP.

Save the current ASP group (Following are the commands to save the current independent ASP group)

- SETASPGRP ASPGRP(primary-ASP-name)
- SAVSECDTA ASPDEV(*CURASPGRP)
- SAVLIB LIB(*ALLUSR) ASPDEV(*CURASPGRP)
- Unmount any QDEFAULT user-defined file systems in the current Independent ASP group
- SAV OBJ(/dev/*) UPDHST(*YES) ASPDEV(*CURASPGRP)
- Mount any QDEFAULT user-defined file systems that were unmounted in an earlier step

Save UDFS ASP (Following are the commands to save an available UDFS ASP)

- SAVSECDTA ASPDEV(ASP-name)
- Unmount any QDEFAULT user-defined file systems in the current Independent ASP group
- SAV OBJ(/dev/*) UPDHST(*YES) ASPDEV(*CURASPGRP)
- Mount any QDEFAULT user-defined file systems that were unmounted in an earlier step

Save libraries and objects within an Independent ASPs as part of a full system save (Option 21) or when you save all user data (Option 23)

In addition to the commands listed in the GO SAVE Options, the server performs the following commands for each available ASP:

- SETASPGRP ASPGRP(asp-group-name)
 - SAVLIB LIB(*ALLUSR) ASPDEV(*CURASPGRP)
 - SAV OBJ(/dev/*) UPDHST(*YES) ASPDEV(*CURASPGRP)
- The server then performs the following command for each available user-defined file system (UDFS) ASP.
- SAV OBJ(/dev/*) UPDHST(*YES) ASPDEV(udfs-asp-name)

The system will also perform a CHKTAP ENDOPT(*UNLOAD) command after the last SAV command it processes.

Restoring iASPs



The system ASP must be restored first

- Then restore any independent disk pool
- Or pools must be manually created through iSeries Navigator

Special consideration for the RSTAUT command

- Use after you have recovered all of your independent ASP's ASPs

Notes Restoring iASPs



Restoring an entire system that is using independent disk pools becomes a more complicated matter. The system ASP must be restored first, then the any independent disk pool. Or an IASP must be manually created through iSeries Navigator or the Create Device ASP (CRTDEVASP) command must be issued and a restore into the IASP be performed.

This requires knowledge of the original disk pool sizes and names. Also, if you are using the restore menu to recover user data, you should consider not using the RSTAUT command until you have recovered all of your independent ASPs. An example recovery procedure of your system would be as follows (do not forget to review the V5R2 Backup and Recovery Guide, SC41-5304):

Restore your SAVSYS

If you are recovering your system from an option 21 save, you may perform option 21 restore, which does the following steps: (You may want to prompt for commands)

```
RSTUSRPRF
RSTCFG OBJ(*ALL)
RSTLIB SAVLIB(*NOSYS)
RSTDLO DLO(*ALL)FLR(*ANY)
RST ((/'*')('/QSYS.LIB ' *OMIT)('/QDLS ' *OMIT))
```

Perform the following steps for each IASP you need:

In OS/400 create your independent storage pools from iSeries Navigator or use the CRTDEVASP command.

```
RSTLIB SAVLIB(*NONSYS) ASPDEV(youriasp)
RST OBJ('/dev/*')
```

Then perform RSTAUT USRPRF(*ALL). Note, If RSTAUT has been performed prior to recovering your independent ASPs, you must RSTUSRPRF first, then use RSTAUT.

Linux & Windows NWSSTG on iASP



You have additional save and restore considerations compared to an OS/400 partition:

- Information under the QFPNSSTG in the system ASP
- The network storage space ((NWSSTG)) named under the /dev/IASPname directory

Restore the network server description as last recovery step

- Automatically link the network server description (NWSD) to the network server storage space (NWSSTG)

If the NWSD is on a switchable IASP

- You may save the pointers and the network server description
- Restore NWSD to the other system in the cluster
- By switching the IASP, Linux & Windows becomes usable on the switched to system

Notes Linux & Windows NWSSTG on iASP



When a network server storage space is created on an IASP, it still creates its pointers in the QFPNSSTG directory which is stored 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 the QFPNSSTG in the system ASP, and also the network storage space named under the /dev/IASPname directory. The following is an example creating a Linux storage space:

```
CRTNWSSTG NWSSTG(LINUXSTG)NWSSIZE(3000)FORMAT(*OPEN)ASP(40)
```

This creates a network storage space called LINUXSTG in ASP40, which, in our example we have previously named ROCHESTER. The actual storage space resides in User Defined File System - /dev/rochester/linuxstg.udfs. It also creates an entry in /qfpnsstg/linuxstg/qfpcontrol along with a /mount directory under /qfpnsstg/linuxstg.

To use a network storage space, you must also create a network storage description to link to the storage space. The following is an example creating a Linux network storage description and adding the link to the network server storage space:

```
CRTNWSN NWSN(LINUXSVR)RSRCNAME(*NONE)TYPE(*GUEST)PARTITION(LINUX) +  
ADDNWSSTGL NWSSTG(LINUXSTG)NWSN(LINUXSVR)
```

If the network storage space was created on a switchable IASP, you may save the pointers and the network server description and restore it to the other system in the cluster, and by switching the IASP, Linux would be usable on the switched to system.

The following commands (shown on next page) will save the objects needed for Linux from System A, put them in a save file in the switchable IASP, restore the objects to System B, then allow Linux to be active in the partition on system B. Note that System B would have to already have a Linux partition configured, but the network server space and the network server description will be restored from System A.

Notes Linux & Windows NWSSTG on iASP



On system A:

```
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)
```

On system B:

After the IASP has ben switched to System B, the following commands will make Linux usable on System B (assuming that the partition already exists):

Vary on or Make Available (iSeries Navigator

```
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 very important to restore the network server description last, as this will automatically link 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.

Using BRMS for iASPs



Updated Edit Backup Control Group Entries screen

- Default values should not affect your current backup strategy

IASP recovery steps are included on the System Recovery Report

```

                                Edit Backup Control Group Entries                                RCHASM20
Group . . . . . : TEST
Default activity . . . . . *BKUPCY
Text . . . . . test
Type information, press Enter.

      Auxiliary Weekly Retain Save SWA
Backup List Storage Activity Object While Message
Seq  Items  Type Pool Device

10 *EXIT
20 *SAVSYS
30 *IBM
40 *ALLUSR      *SYSBAS
50 *ALLDLO
60 *LINK      *ALLAVL
70 *EXIT

Bottom
F3=Exit      F5=Refre
F11=Display exits  F12=Canc

```

Items to Back Up or Action	Type	Track Object Detail	Disk Pool
Security data	None		System and basic disk pools
Configuration data	None		Libobjects
All user data	Errors		Libobjects
All document data	None		Libobjects
All directory data	All		Libobjects

Notes Using BRMS for iASPs



The V5R2 BRMS (5722-BR1) Edit Backup Control Group Entries screen has been updated as shown to include a new Auxiliary Storage Pool Device field. In the screen example, the IASP name is Libobjects.

This field will not appear on some backup items entries. Typically this occurs for backup items which cannot reside on auxiliary storage pool devices. The Auxiliary Storage Pool Device prompt will be automatically filled in for entries of your existing backup control groups to reflect the scope of the save across auxiliary storage pool devices.

These default values should not affect your current backup strategy and should be consistent with what was being 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 (2-32) auxiliary storage pools.
- The *ALLAVL value for the *LINK backup items saves the links on the system (1) and any basic user (2-32) auxiliary storage pools as well as the links on all available auxiliary storage pool devices.

Note: When saving directory and files, you should unmount any mounted user-defined file systems (UDFSs) prior to the save to assure the objects in the mounted over directories are saved. UDFSs are automatically unmounted on auxiliary storage pool devices when the system is in restricted state. UDFSs on the system or basic user auxiliary storage pools need to be explicitly unmounted. Any unmounted UDFSes need to be remounted after the save.

iASP and Availability



iASPs represent a single point of failure in the system

Not a disaster recovery solution

Pre-V5R1 HSL adapters do not work with iASPs

iASP and balancing workloads

iASPs can co-exist with HABP solutions

- HABP solutions provide geographical dispersal of the data
- The IASP cannot be used for balancing workload

Notes iASPs and Availability



Using only IASPs, there are some deficiencies in total system availability remain: for example:

IASPs represent a single point of failure in the system: If the disks in the IASP are permanently damaged, then the data is unrecoverable so special care is need to ensure that the correct level of disk protection is chosen for the IASP. For critical data, tower level mirroring may be appropriate. For less critical data, RAID protected disks may be the best solution. Additionally, a secondary IASP used for journaling should exist in a separate RAID set to ensure that the latest updates are available if the data in a primary IASP becomes damaged or lost.

Not a disaster recovery solution: Because of loop limitations with HSL, the systems must be within 250 meters of each other (15 meters with some configurations). The production and backup systems can be several thousand kilometers apart when replication is used. IASPs are, therefore, not useful as a disaster recovery solution.

Pre-V5R1 HSL adapters do not work with IASP: If the IASP configuration involves an HSL loop, a V5R1 supported HSL adapter (such as feature #2739 Optical Bus Adapter) is required. HSL adapters available prior to V5R1 (for example the #2695 Optical Bus Adapter) do not work with IASPs. Models 830 and 840 systems with original HSL cabling can be upgraded to newer HSL features. Models 270 and 820 will require a model upgrade to 830 or higher to support the required HSL technology.

IASP and balancing workload: Careful planning should be done to balance workloads across systems with IASP's when ever possible. The way to achieve the is to use multiple IASP groups with separate applications or application instances. For example, if used with Domino, split the Domino servers across at least 2 IASP groups, with each system in the loop acting as the primary for only 1 of the groups.

IASPs coexists with HABP solutions: While it is true that IASPs provide additional availability tools on the iSeries server, IASP functions do not replace High Availability Business Partner (HABP) solutions. Customers who simply want to increase availability at a lower cost can use independent ASPs without adding disks for the backup system. Independent ASPs coexists 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 the limitations of the HSL Loop. With some V5R1 configurations, the distance is limited 15 meters.
- **An HABP solution provides switchover capability between two systems:** The HABP approach for switchover and failover can be complex. Monitoring is often performed at a high level, although newer versions of the HABP products now use IBMs cluster monitor. In comparison, using switchable IASP with clustering provides a means to handle a complex requirement in a relatively simple way. The heartbeat monitoring that is implemented with IBM clustering is very sophisticated. Once properly setup, the switchover or failover to the backup system can be nearly seamless.

iASP Scenarios



Non-switchable simple independent disk pool

Switchable independent disk pool

- Switchable tower
- Switchable IOP with logical partitions
- Switchable tower with logical partitions

Integrated xSeries Servers and independent disk pools

Linux and independent disk pools

Notes iASP Scenarios



The following foils provide several implementation and usage examples of independent disk pools. They range from the ultra-simple to extremely complex. These are meant to demonstrate the flexibility and possibilities of independent disk pools in the day-to-day iSeries world. The scenarios we will have a brief look at are:

- Non-switchable simple independent disk pool
- Switchable independent disk pool
 - Switchable tower
 - Switchable IOP with logical partitions
 - Switchable tower with logical partitions
- Integrated xSeries Servers and independent disk pools
- Linux and independent disk pools

Non-switchable simple independent disk pool



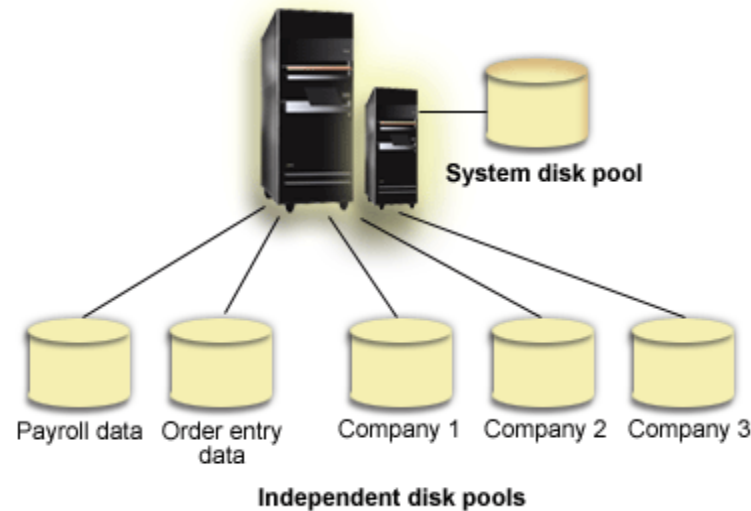
Failure in one Database does NOT disrupt other Databases

While the system is active

- Can be independently taken offline
- Can be independently brought online
 - no initial program load (IPL) required

Data can be offline until it is needed

- Shorten processing time for operations such as IPL and reclaim storage



Notes Non-switchable simple independent disk pool



In a single-system environment, a standalone, or dedicated, independent disk pool can be taken offline independent of other disk pools because the data in the independent disk pool is self-contained. In this scenario, the user has five independent disk pools. They could represent three different applications where the third application may have archived data. The system automatically creates the system disk pool (referred to as Disk Pool 1 or ASP 1) which contains all system programs and system data.

Failure in one Database does NOT disrupt other Databases

The most important advantage of this isolation is that should an iASP fail, that database will not be available. All other databases will not be impacted by this failure.

While the system is active

That is, all of the necessary system information associated with the independent disk pool's data is contained within the independent disk pool. The independent disk pool can also be brought online while the system is active; no initial program load (IPL) required. Using independent disk pools this way can be very useful, for example, if you have large amounts of data that are not needed for normal day-to-day business processing. The independent disk pool containing this data can be left offline until it is needed. When large amounts of storage are normally kept offline, you can shorten processing time for operations such as IPL and reclaim storage.

Data can be offline until it is needed

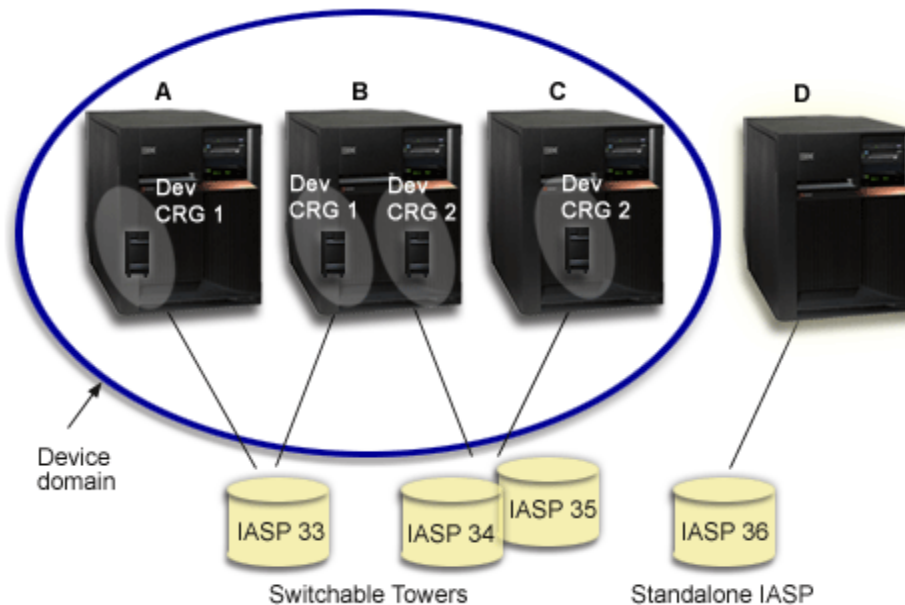
Data that does not need to be on-line can be brought offline until it is needed. This way the system does not have to maintain this data and the internal structures that are associated with it. For example the system will have smaller database cross reference files for the data that is online. Besides operational advantages, this also results in a shorter processing time for operations such as IPL and reclaim storage.

Switchable independent disk pool



Switchable towers

- First tower can be switched between nodes A and B
- Second tower can be switched between nodes B and C
- Node D can only access IASP36, a private independent disk pool



Notes Switchable independent disk pool



In this example, the figure shows a cluster consisting of four nodes. Nodes named A, B, and C are defined to be in the same device domain. There are two switchable towers - one contains IASP33 and the other contains IASP34 and IASP35. The tower containing IASP33 is on an HSL loop that also contains nodes A and B. This first tower can be switched between nodes A and B. The tower containing IASP34 and IASP35 could be on another HSL loop that also contains nodes B and C. This second tower can be switched between nodes B and C. Node D is contained in the cluster, but is not a member of the device domain and therefore can only access IASP36, a standalone, or dedicated, independent disk pool.

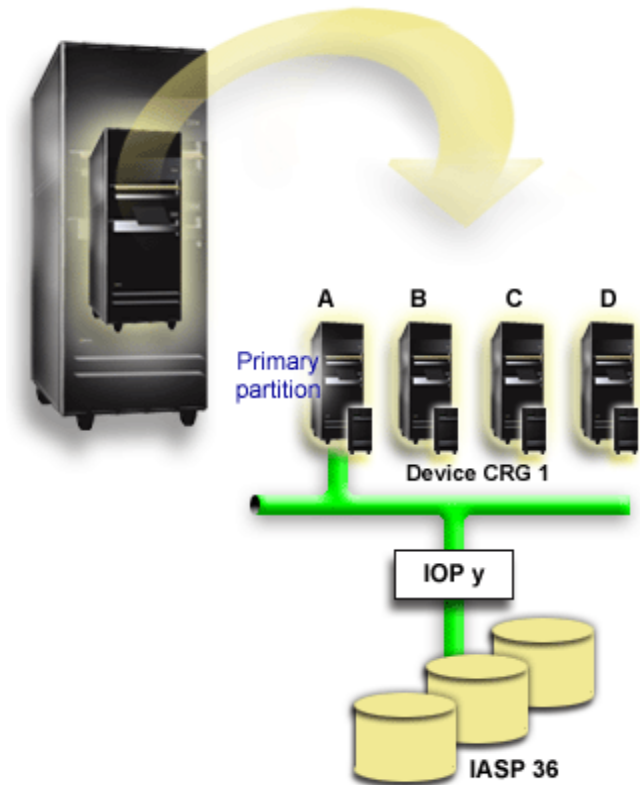
For details on cluster domains please refer to the IBM Redbook 'Clustering and IASPs for Higher Availability on the IBM iSeries Server', SG24-5149-01, or to the iSeries Information Center.

Switchable independent disk pool



Switchable IOP with logical partitions

- Four logical partitions on a single iSeries server
- IOP Y is on the shared bus
 - It can be switched between all of the nodes in the cluster.
 - When the IOP is switched
 - Everything that is physically connected to that IOP is also moved to the new primary node.



Notes Switchable independent disk pool



In this logical partition example, the following figure shows a cluster consisting of four logical partitions on a single iSeries server. All four nodes belong to the same device domain. IASP36 is composed of disk units accessible through IOP Y. IOP Y is on the shared bus so it can be switched between all of the nodes in the cluster: A, B, C, and D.

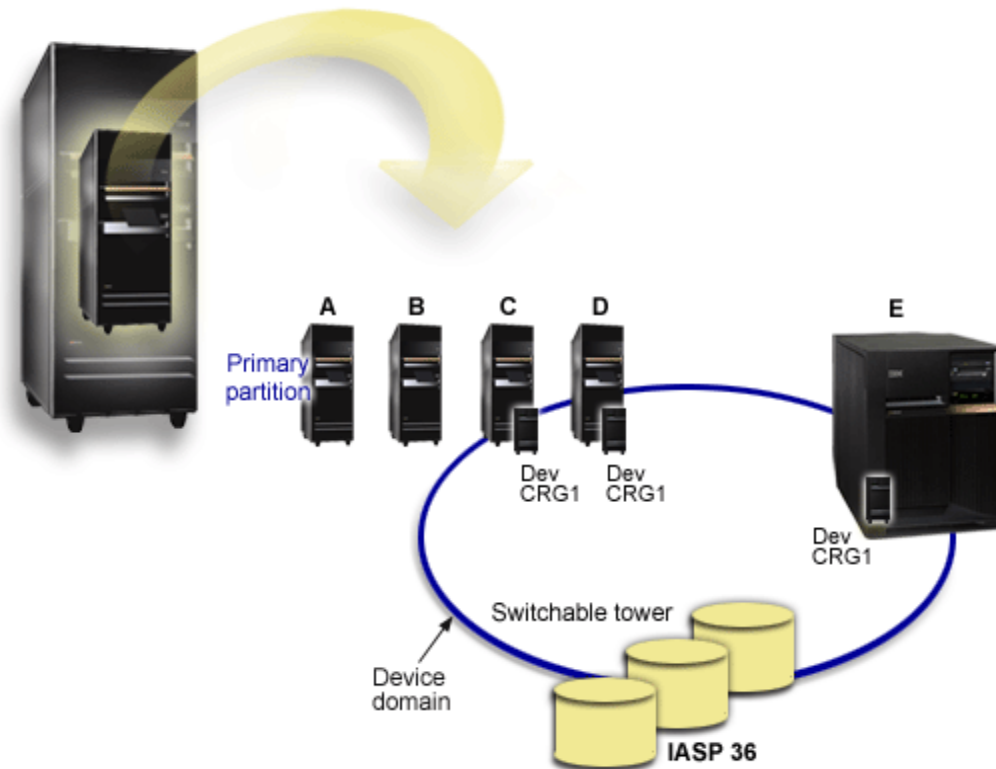
Remember when the IOP is switched, everything that is physically connected to that IOP is also moved to the new primary node.

Switchable independent disk pool



Switchable tower with logical partitions

- A combination of the previous two examples



Notes Switchable independent disk pool



The example, shown in this sheet, depicts a combination of the previous two examples. IASP36 is composed of disk units contained in a switchable tower. The tower is on the same HSL loop as two systems, one of which is made up of four logical partitions. Assuming that nodes C and D, and the second server, node E, are defined to be in the same device domain, the independent disk pool can be switched between those three nodes.

Integrated xSeries Servers and independent disk pools



A real world example



Notes Integrated xSeries Servers and independent disk pools



This is a real world example of using Integrated xSeries Adapters and independent disk pools. In this environment the disks containing the server storage for the IXAs and the IXAs 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 IXAs can be switched to the backup system. For planned outages it allows the applications running on the IXAs to continue to be available. For unplanned outages you will have a determinable time for recovery.

Linux and independent disk pools



A higher available Linux system



Notes 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 more highly available Linux "system." To use Linux with independent disk pools you need to use hosted or virtual dasd. The disk space for the Linux partition is created using network server storage spaces. One of the selling points for putting Linux on an iSeries partition was 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 happens to be on an independent disk pool then you can switch that disk pool to another system and bring up that Linux system on another partition. This sheet shows a possible configuration with Linux and independent disk pools.

If your Linux disk requirements are significantly 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 for 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.



iASP Benefits

Benefits of independent disk pools



Availability

- Keep data available to an application even in the event of a single system outage
- An alternative process of 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

Isolate low-use data with ability to bring online only when needed

- Reduce system start time

Reclaim storage by independent disk pool

Isolate data associated with specific applications

Save/restore management by independent disk pool

Perform application maintenance that does not affect entire system

Notes



The concept of “independent storage” is new to iSeries. For many years, iSeries customers, and AS/400 and System/38 customers before that, have been used to working 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 have always had a very close relationship to the System ASP and you could not have duplicate library names in different user ASPs. In effect, all the storage attached to a system was a single namespace.

In more recent versions of OS/400, various file systems such as QOpenSys, QNetWare and QNTC were introduced and these were generically known as the Integrated File System (IFS). By defining User Defined File Systems (UDFSs), 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 and as such, had limited use for most iSeries customers, being primarily limited to Domino, web-serving applications and Integrated xSeries Adapters or Servers.

Now, with the enhancements introduced in V5R2 to support the traditional library/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 of 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

Isolate low-use data with ability to bring online only when needed

- Reduce system start time

Reclaim storage by independent disk pool

Isolate data associated with specific applications

Save/restore management by independent disk pool

Perform application maintenance that does not affect entire system

Current Flash Copy



■ Local, instant point-in-time copy of all iSeries disks

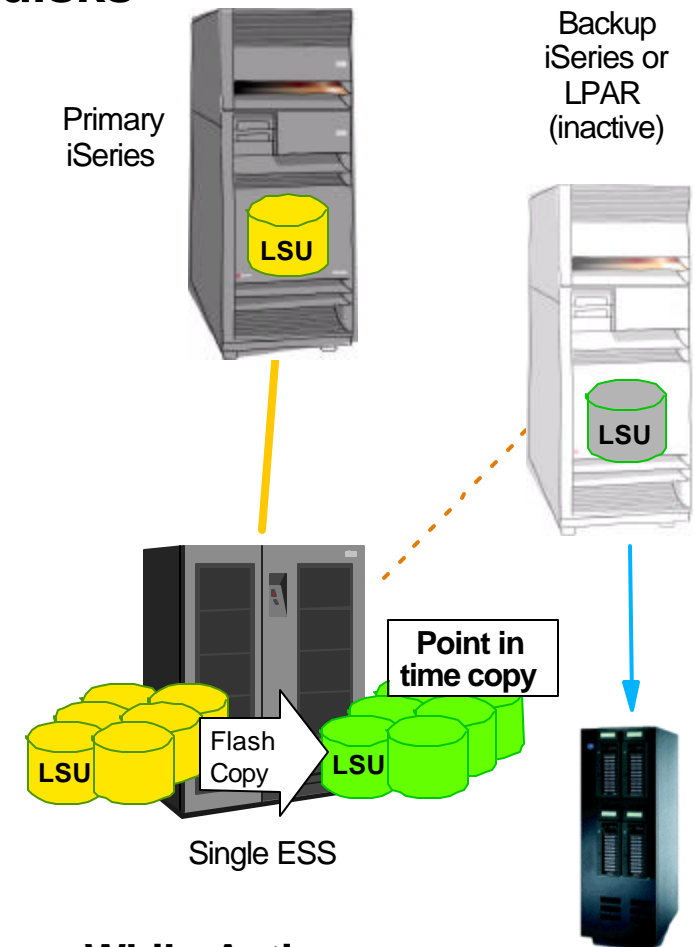
- ▶ Addresses outages due to backups
- ▶ Invoked from IBM TotalStorage ESS Specialist
- ▶ 'NOCOPY' option

■ Benefits:

- ▶ Fast checkpoint
- ▶ Full system backups with minimal downtime
- ▶ Multiple platforms supported

■ Considerations:

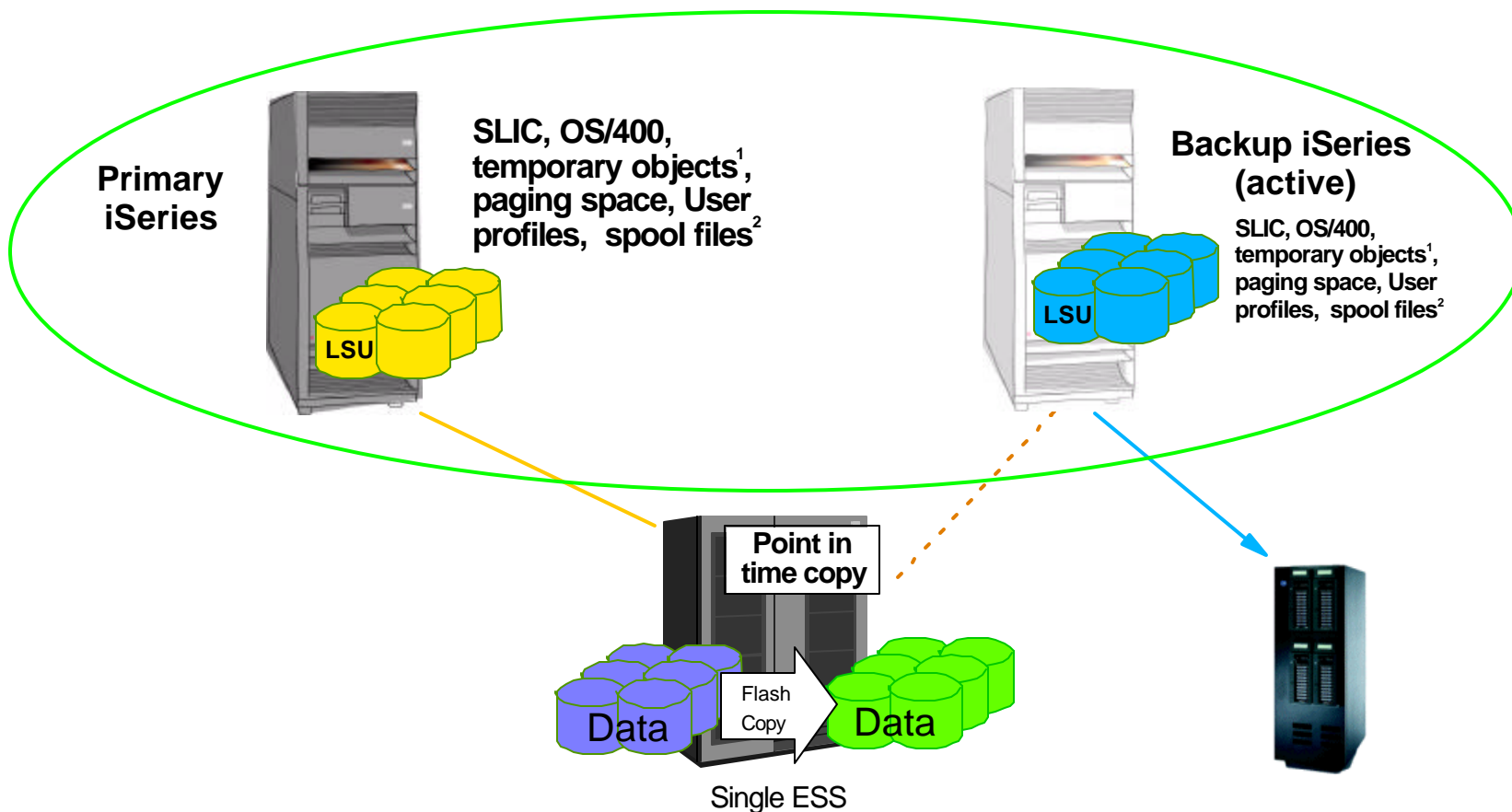
- ▶ Must quiesce (power off) primary server
 - Up to 2-hour outage or more on primary server
- ▶ New complexities
- ▶ Manual processes - no CLI
- ▶ Recovery considerations
- ▶ May impact disk performance
- ▶ For nightly backups, may not be faster than OS/400 Save-While-Active



IASP Flash Copy



iSeries Cluster and Device Domain



- 1) Temporary space includes Qtemp libraries, index build space etc
- 2) There is a statement of direction to allow Spool files in an IASP in the future

IASP Flash Copy



■ Benefits over Remote Load Source process

- ▶ No IPL to attach second copy of data to the backup server (An IPL must happen at the end of the process)
- ▶ Saving can start after around 5 - 10 minutes after the flash
- ▶ No issues with SLIC or OS/400. The second system has it's own copy
- ▶ Saves will be done through OS/400. This will take advantage of the performance of our save techniques including parallel saving.
- ▶ Replication on the production server (with HABP) to another system can continue to run without getting behind while the save is running
- ▶ Multiple IASP's can be attached and saved in a single process. This is a huge benefit compared to current remote load source mirroring technique.

■ Considerations:

- ▶ Must quiesce (vary off IASP)
 - Around 10 minute outage on primary server
- ▶ New complexities - customer must be in an IASP environment to use it
- ▶ Manual processes to Flash- no CLI
- ▶ Recovery considerations
- ▶ Disk sizing for system ASP is very important - this requires the fastest disk on the system as this is where memory paging, index builds etc happen.
- ▶ ~~IPL the backup system after the save has taken place~~



Summary

Summary



- V5R2 brings iASP to stage 2
 - 'Most' object support is enabled
- Switchable iASP adds to iSeries HA proposition
- Provides application consolidation



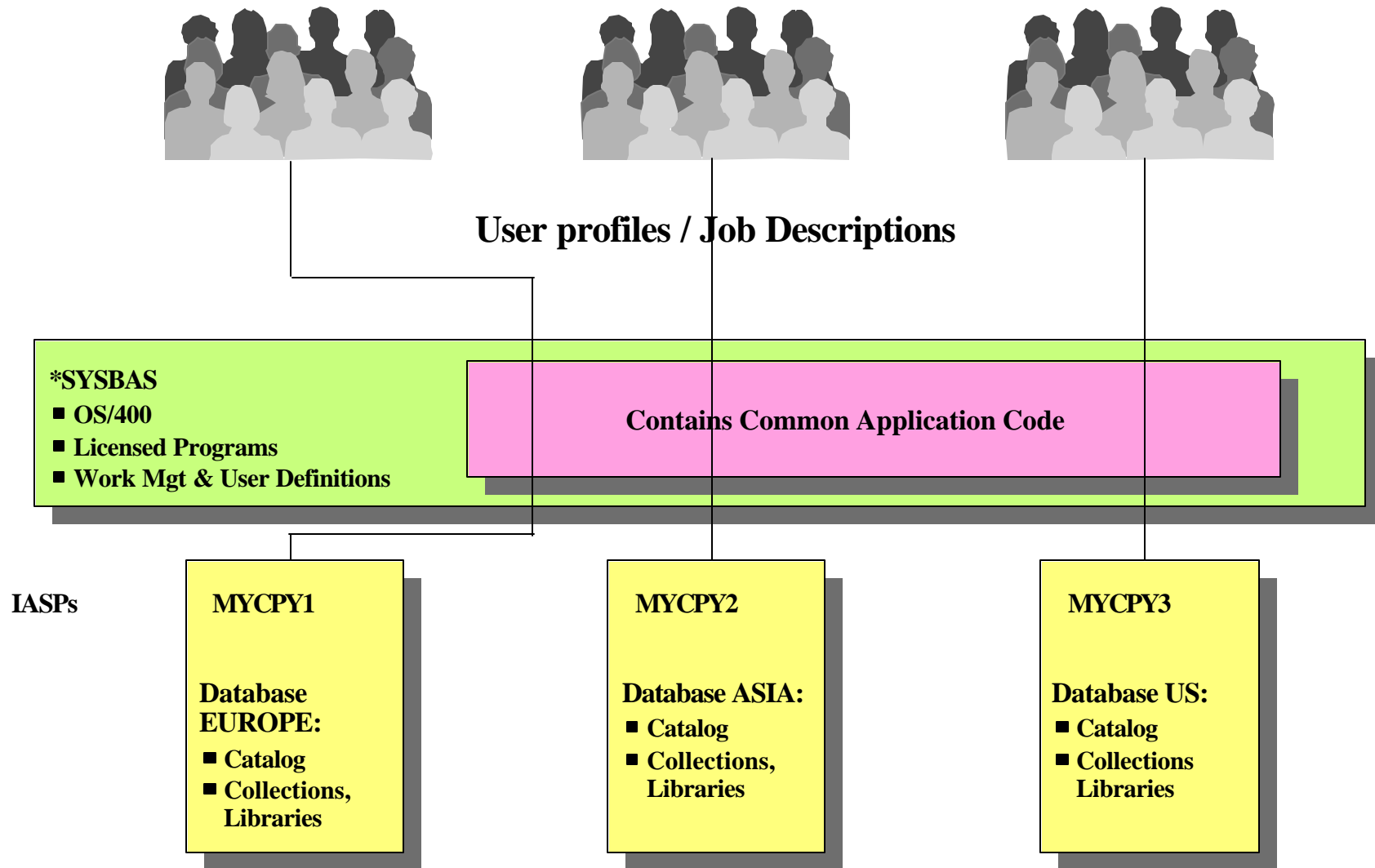
Sources of Information

Where to find more information



- Information Center
- Redbook SG24-6802 (available as Redpiece December)
 - Adding new material Feb 2003
- iTSC
 - IASP Feasibility Study & Workshop(s) for IBM iSeries server™®
- HA info at *<http://www-1.ibm.com/servers/eserver/series/ha/>*
- For customisation help CTCEMEA@FR.IBM.COM

Multiple Database Support - Setup Example



Notes Multiple database setup



This chart explains how to set up a consolidated environment on a single, not a partitioned system, where a common set of application code is used to access individual files with the same names, but of which the data has to be kept separately for business reasons.

Three IASPs are defined on this system, each of which contains the data libraries of each entity in the company. Each IASP will therefore contain the same libraries with the same tables, views and indexes, but with different data. In our example, for purposes of simplifying change management, the application programs are stored only once in the system ASP. Also, for illustrating the different data, we show our database names representing geographical areas - different than the IASP name in which the database resides.

The user profiles and their attributes are migrated from each system, and are associated with a job description that directs them to the IASP device where their data resides. This is done by specifying the name of the IASP device on the *Initial ASP Group* (INLASPGRP) parameter of the Job Description associated with each user profile.

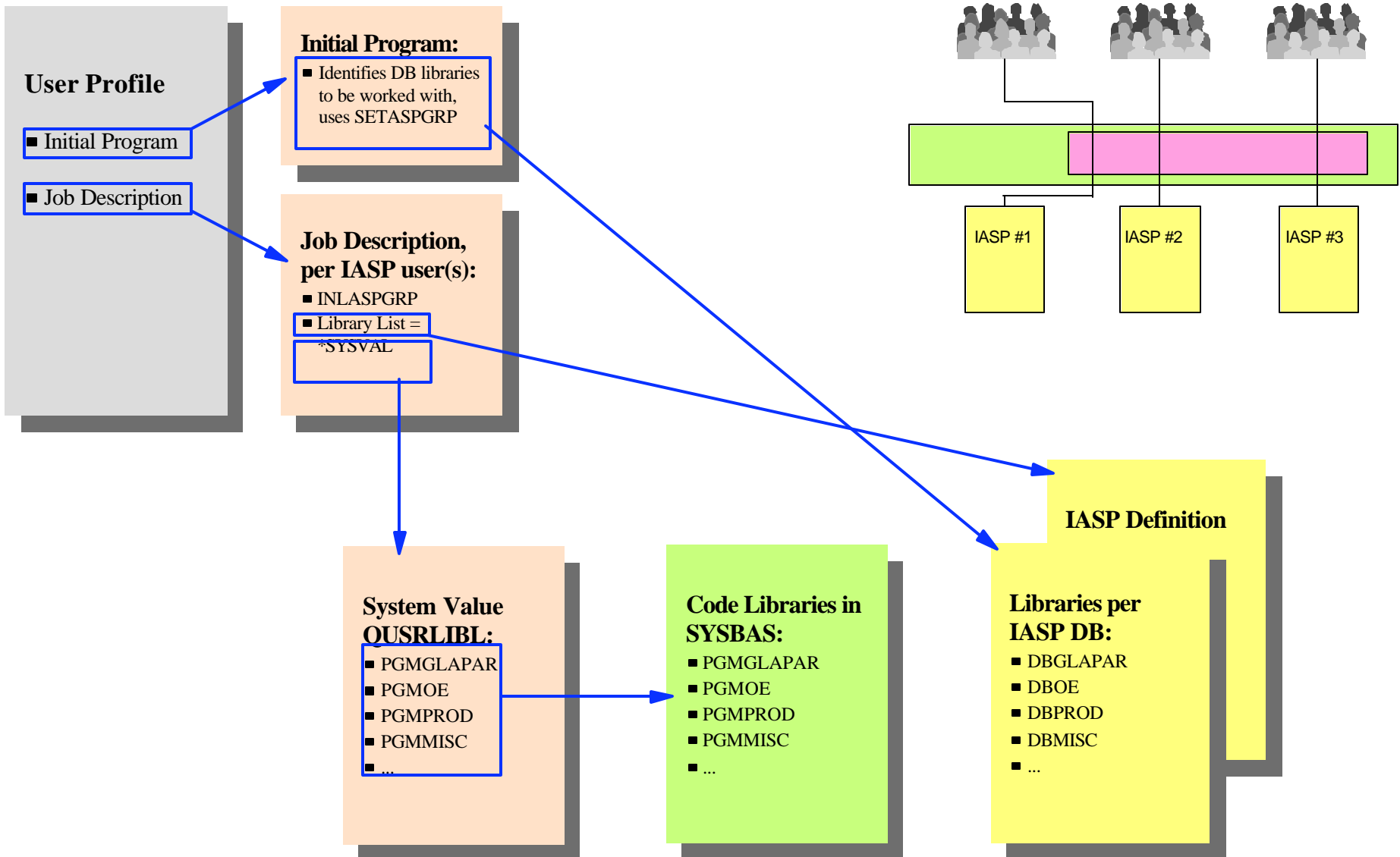
This way, a user that signs on to the system will have access to the common application code, residing in the system ASP, and to his data libraries, residing in the IASP.

If this system is running an application that needs to be enabled for support of different cultural values, it can still be achieved in this way, as long as the individual set up of the cultural objects follows the design principles as detailed in the *International Application Development* (SC41-5603) manual. In this case, multiple instances of the different cultural elements can be stored in the IASPs in their libraries, much as is the case on a system without multiple database support.

The advantages of such a consolidation, which can maintain its naming schema, include:

- No need to change the naming of all data containers, data objects or data attributes to fit in a consolidated IT service delivery environment; no secondary changes to the security setup either.
- Transparency for change and problem management including help desk support organizations.

Multiple Database Support - Setup Details



Notes Multiple Database setup



This figure shows more details of using multiple IASPs on a single system. This example is intended only as what we believe to be an implementation adhering to the principle of separate placing of processes and data.

The base assumption is to keep as much as possible the code libraries (programs, modules, environment descriptions) separated in the system ASP, while the data libraries are contained in separate IASPs, as required by the business entities. This is as discussed earlier in the notes for the foil: Multiple Database Support - Details.

The system value QUSRLIBL contains the names of the common code libraries and is used in the job descriptions associated with the different user profiles; each user profile has an initial program that sets up the data libraries the user will work with.

In a multilingual environment, you can add the libraries, containing the MRI or any other culturally dependent objects, in the system ASP also. These might also reside in the IASP.

Note: In an SQL environment use CONNECT to specify the right database. To achieve the fastest performance, make sure that the database to which you perform an SQL CONNECT corresponds with your current library name space. You may need to use the SETASPGRP command first to achieve this. If the SQL CONNECT function is not operating within the same library name space, the application will use Distributed Relational Database Architecture (TM) support which can affect performance.



Disk Pools GUI access

Make disk pool available



The screenshot shows the iSeries Navigator interface. The left pane displays a tree view of storage resources, with 'Disk Pool 33 (Flake)' selected. A context menu is open over this selection, with 'Make Available...' highlighted. The right pane shows a table of 'Disk Pool Groups'.

Disk Pool Group	Disk Pools	Status
Group Flake	1	Unavailable

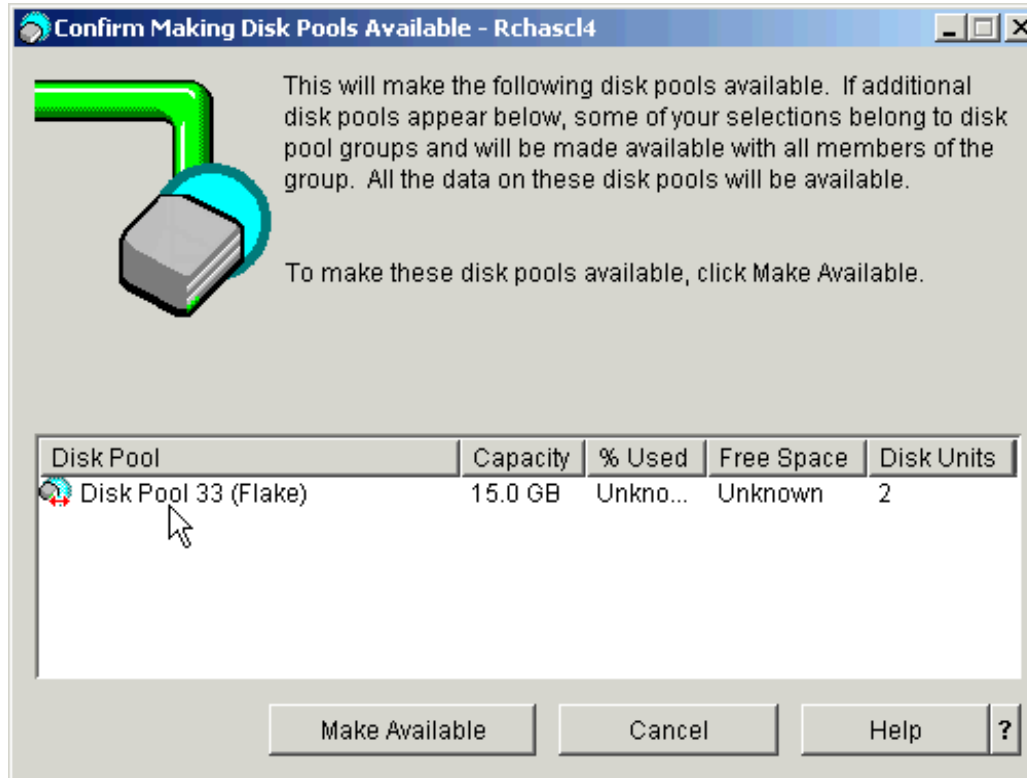
At the bottom of the window, a status bar reads: "Makes a disk pool available for use."

Notes Make disk pool available



In the window we have right clicked the disk pool 33 icon. The menu shows the option to make the disk pool available or 'vary theiASP on'.

Confirm availability



Notes Confirm availability



The confirm availability screen is then displayed. Clicking the make available button will start the vary on process. The pipe line will pump until the iASP is varied on.

Switching Disk Pools



The screenshot shows the iSeries Navigator interface. The left pane displays a tree view of system components, including 'Switchable Hardware'. A context menu is open over this component, with the 'Switch...' option highlighted. The right pane shows a table of switchable hardware.

Name	Hardwar...	Type	Status	Available at
Flake	switchabl...	Primary	Varied off	Yes

Switches the selected group to a backup node.

Notes Switching Disk Pools



Once the disk is available it can then be switched. Expand the switchable resources icon and right click the switchable hardware. The switch option is then displayed.

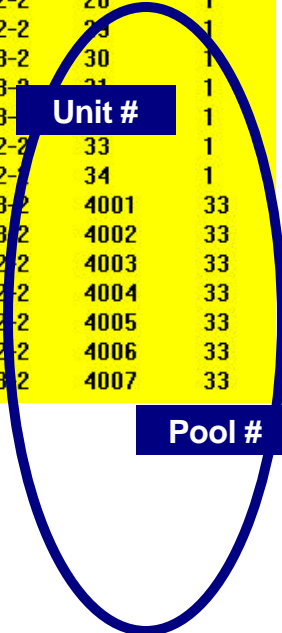
This assumes that a cluster has been setup on the system.

Disk unit numbering



Adel400a.au.ibm.com: All Disk Units

Disk U...	Status	Capac...	Free S...	Reser...	% Busy	Protection	Compression	Type-Mode...	Unit N...	Disk P
Dd020	Active	1.7 GB	1.3 GB	1.0 MB	4%	Parity	Not compre...	6606-072-4	20	1
Dd021	Active	0.7 GB	0.5 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	21	1
Dd022	Active	1.0 GB	0.7 GB	1.0 MB	4%	Parity	Not compre...	9337-023-2	22	1
Dd023	Active	1.0 GB	0.7 GB	1.0 MB	4%	Parity	Not compre...	9337-023-2	23	1
Dd024	Active	0.7 GB	0.5 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	24	1
Dd025	Active	0.7 GB	0.5 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	25	1
Dd026	Active	1.0 GB	0.7 GB	1.0 MB	4%	Parity	Not compre...	9337-023-2	26	1
Dd027	Active	0.7 GB	0.5 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	27	1
Dd028	Active	0.7 GB	0.5 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	28	1
Dd029	Active	0.7 GB	0.5 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	29	1
Dd030	Active	1.0 GB	0.7 GB	1.0 MB	5%	Parity	Not compre...	9337-023-2	30	1
Dd031	Active	1.0 GB	0.7 GB	1.0 MB	5%	Parity	Not compre...	9337-023-2	31	1
Dd032	Active	1.0 GB	0.7 GB	1.0 MB	5%	Parity	Not compre...	9337-023-2	32	1
Dd033	Active	0.7 GB	0.5 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	33	1
Dd034	Active	0.7 GB	0.5 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	34	1
Dd035	Active	1.0 GB	0.0 GB	1.0 MB	4%	Parity	Not compre...	9337-023-2	4001	33
Dd036	Active	1.0 GB	0.0 GB	1.0 MB	4%	Parity	Not compre...	9337-023-2	4002	33
Dd037	Active	0.7 GB	0.0 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	4003	33
Dd038	Active	0.7 GB	0.0 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	4004	33
Dd039	Active	0.7 GB	0.0 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	4005	33
Dd040	Active	0.7 GB	0.0 GB	1.0 MB	4%	Parity	Not compre...	9337-022-2	4006	33
Dd041	Active	1.0 GB	0.0 GB	1.0 MB	4%	Parity	Not compre...	9337-023-2	4007	33



- IASPs**
- system assigns disk unit numbers
 - 4001 - 6047
 - system assigns pool numbers
 - 33 - 99

Notes Disk unit numbering



Disk units in an independent ASP are numbered differently from normal internal drives located in SYSBAS. Drives in an iASP are numbered from 4001.

The graphic shows there are 7 drives in disk pool 33 numbered 4001 to 4007.

What if a disk pool fills up ?



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.

Notes What if a disk pool fills up ?



It is possible to fill your iASP. An iASP cannot overflow to SYSBAS. An error message will be issued and the iASP will stop. More disk must be added before the iASP can be restarted, or varied on.

This is a circumstance that you do not want to happen. There are threshold messages that can be set to warn you of the status of your iASP capacity.



iASP standalone setup with GUI

Operation Navigator disk pools view



The screenshot shows the iSeries Navigator interface. The left pane displays a tree view of system components under 'My Connections'. The right pane displays a table of disk pools for 'Rchasm20: Disk Pools'.

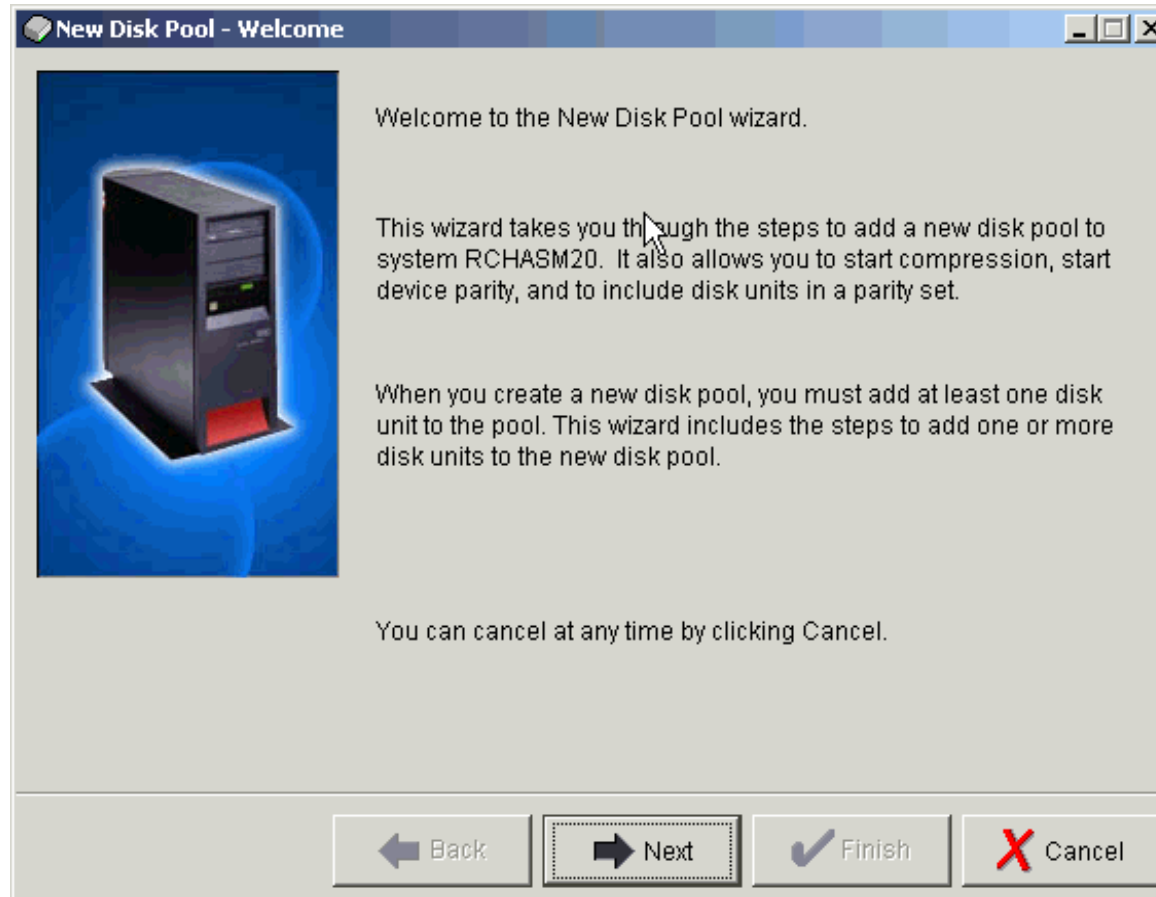
Disk Pool	Capacity	% Used	Status
Disk Pool 1	55.8 GB	49%	Available
Disk Pool 33 (Ryan)	6.4 GB	1%	Available
Disk Pool 34 (Benjamin)	6.4 GB	99%	Available

Notes Operation Navigator disk pools view



In the graphic you can see the Operations Navigator view of disk units. This shows disk pools in server IASPSYS.

Disk Pool wizard



Notes Disk Pool wizard



There is a disk pool wizard to simplify the setup of independent disk pools.

Disk Pool type



New Disk Pool [X]

Type of disk pool: Primary [v]

Disk pool: Newiasp

Database: Generated by the system [v]

Note: If you want to create a switchable disk pool, be sure to use the appropriate clustering function before using this wizard.

Protect the data in this disk pool

OK Cancel Help ?

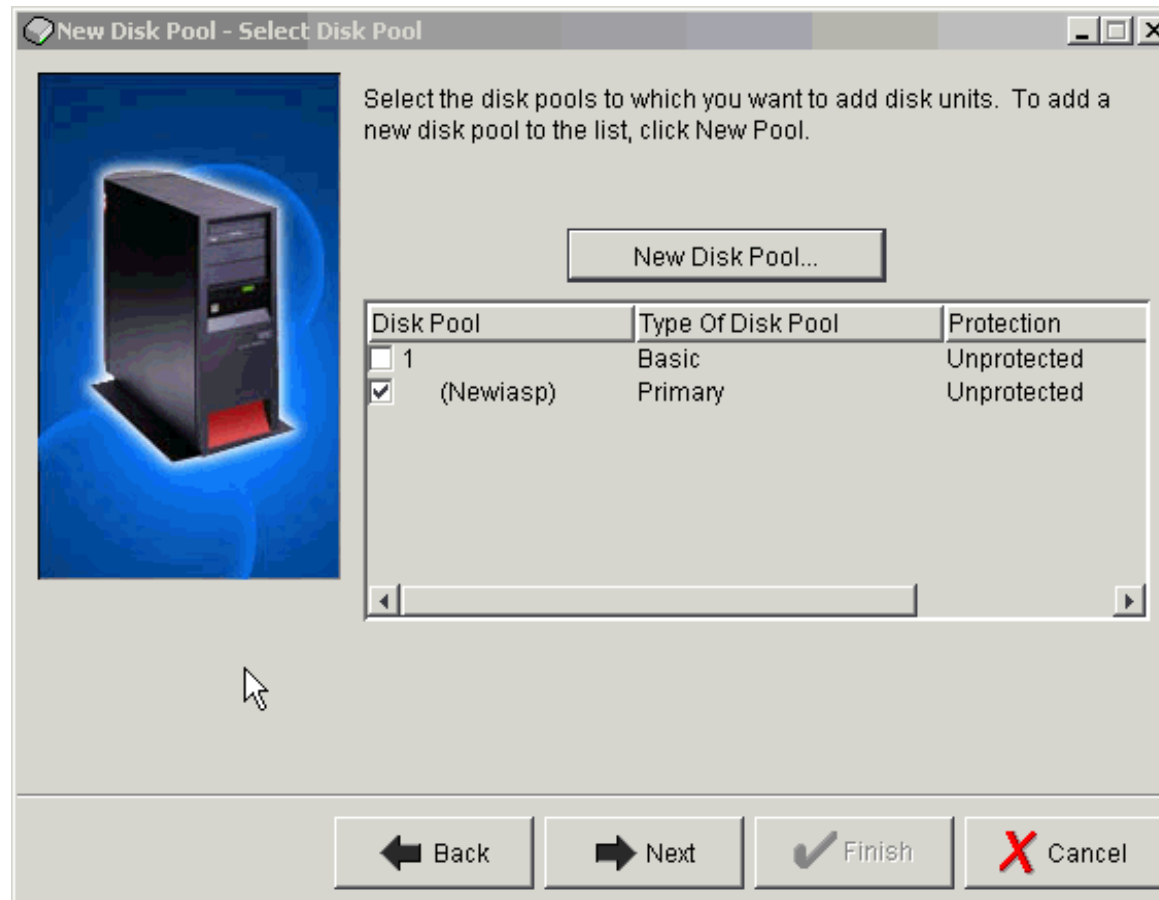
Notes Disk Pool type



The new disk pool panel appears and allows you to enter the name, the type of pool requested, UDFS, Primary or Secondary, and where the pool should be created from. In the following example we will create a primary disk pool.

You can also check the box to allow for protection to be started for this new pool.

Confirmation Window

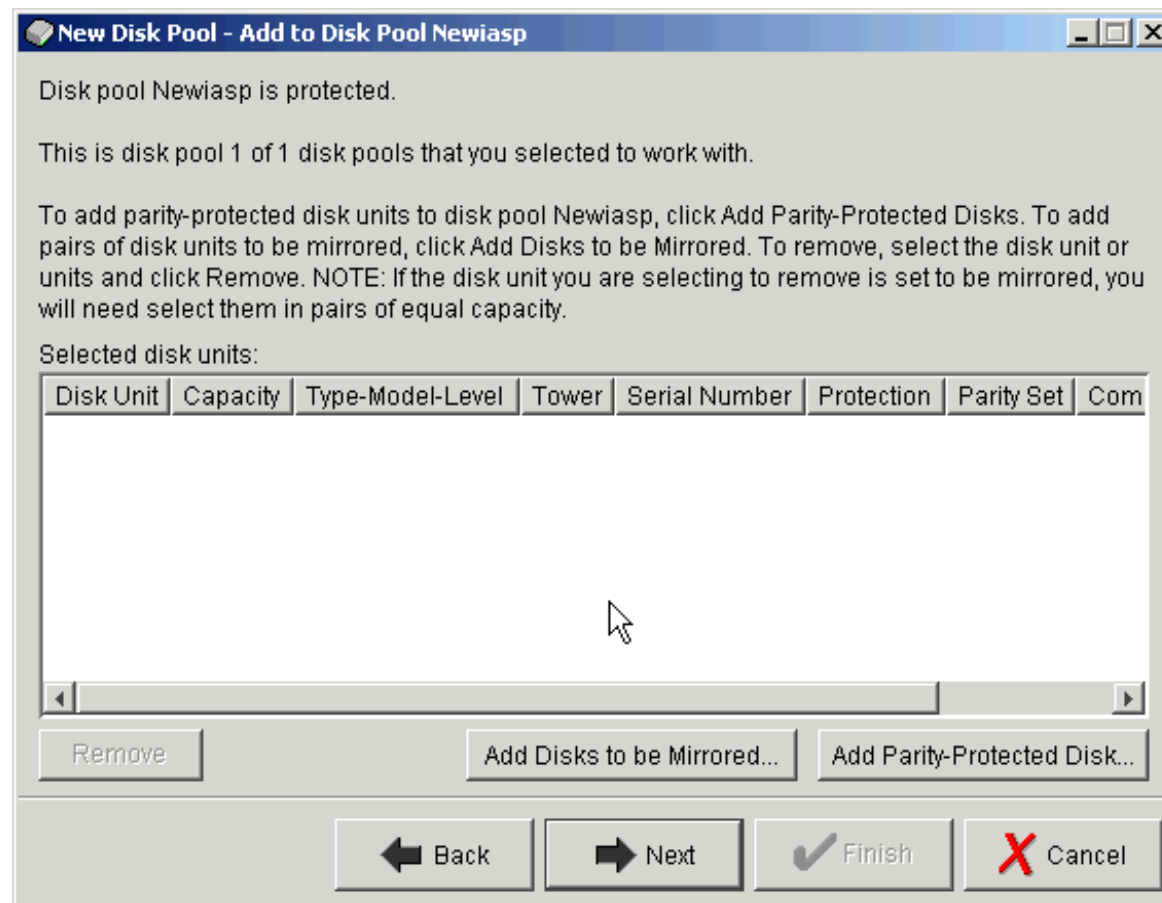


Notes Confirmation Window



The confirmation window displays the state of your new creation, disk name, type and protection.

Disk pool protection

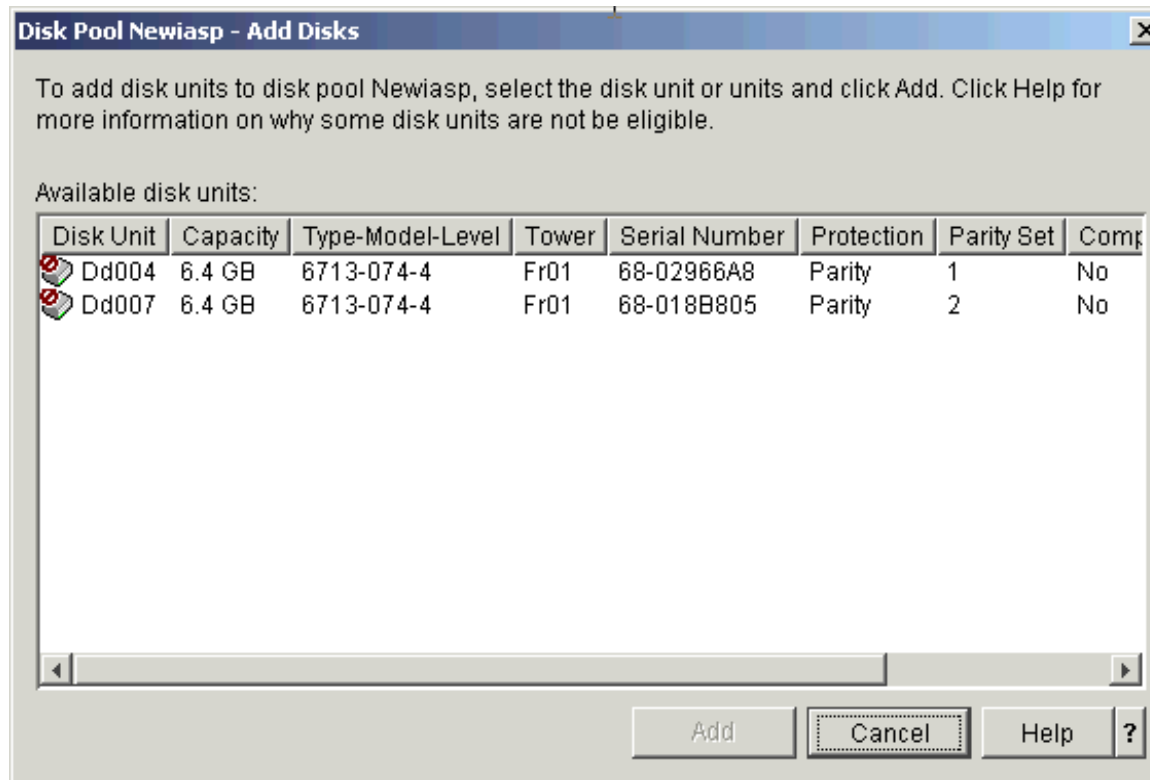


Notes Disk pool protection



Select disks for protection. Add them either with mirroring, RAID or no protection.

New disk pool -add disks



Notes New disk pool -add disks



Selected disks have been tagged to be parity protected.

New disk pool summary



New Disk Pool - Summary

The following shows how the configuration you specified will look after the disk units are added. If this is not correct, click Back and make any necessary changes.

If this is correct, click Finish to begin adding the disk units.

Disk Pool	Type Of Disk Pool	Balance	Protection	Capacity	Disk Unit
1	Basic	No	Unprotected	55.8 GB	
			Parity	6.4 GB	Dd001
			Parity	6.4 GB	Dd002
			Parity	8.6 GB	Dd003
			Parity	6.4 GB	Dd005
			Parity	6.4 GB	Dd006
			Parity	8.6 GB	Dd008
			Parity	6.4 GB	Dd009
			Parity	6.4 GB	Dd010
			(Newiasp)	Primary	No
Parity	6.4 GB	Dd004			

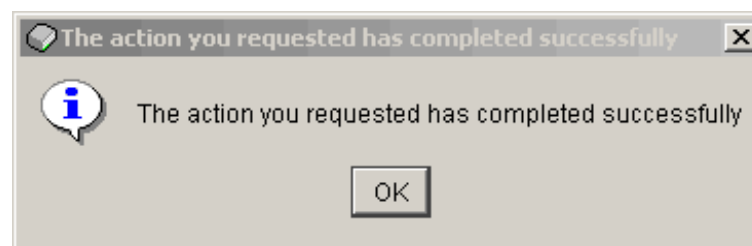
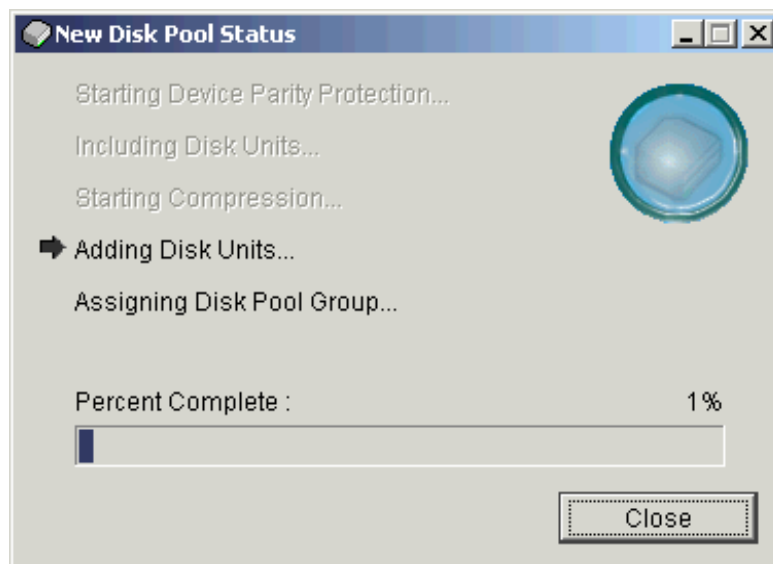
Back Next Finish Cancel

Notes New disk pool summary



A summary screen appears showing the type of protection selected and for which disks.

New Disk Pool - status/completion



Notes New Disk Pool - status/completion



A status window will appear showing disks are being added to your new pool. This process can be lengthy, similar to adding disks to an ASP and starting protection for normal disks.

Create secondary/UDFS disk pools



New Disk Pool [X]

Type of disk pool: Secondary [v]

Disk pool: Newiasp

Primary: 33 (Nonswitch) [v]

Note: If you want to create a switchable disk pool, be sure to use the appropriate clustering function before using this wizard.

Protect the data in this disk pool

OK Cancel Help ?

New Disk Pool [X]

Type of disk pool: UDFS [v]

Disk pool: Newiasp

Note: If you want to create a switchable disk pool, be sure to use the appropriate clustering function before using this wizard.

Protect the data in this disk pool

OK Cancel Help ?

Notes Create secondary/UDFS disk pools



Creation of secondary or UDFS disk pools follows the same process as for Primary disk pools.



Switchable iASP setup

Create a cluster



The screenshot shows the iSeries Navigator interface. The left pane displays a tree view of the environment, with 'Clusters' selected under 'My AS/400 Connections'. A context menu is open over the 'Clusters' folder, with 'New Cluster...' highlighted. The right pane shows a table of existing clusters.

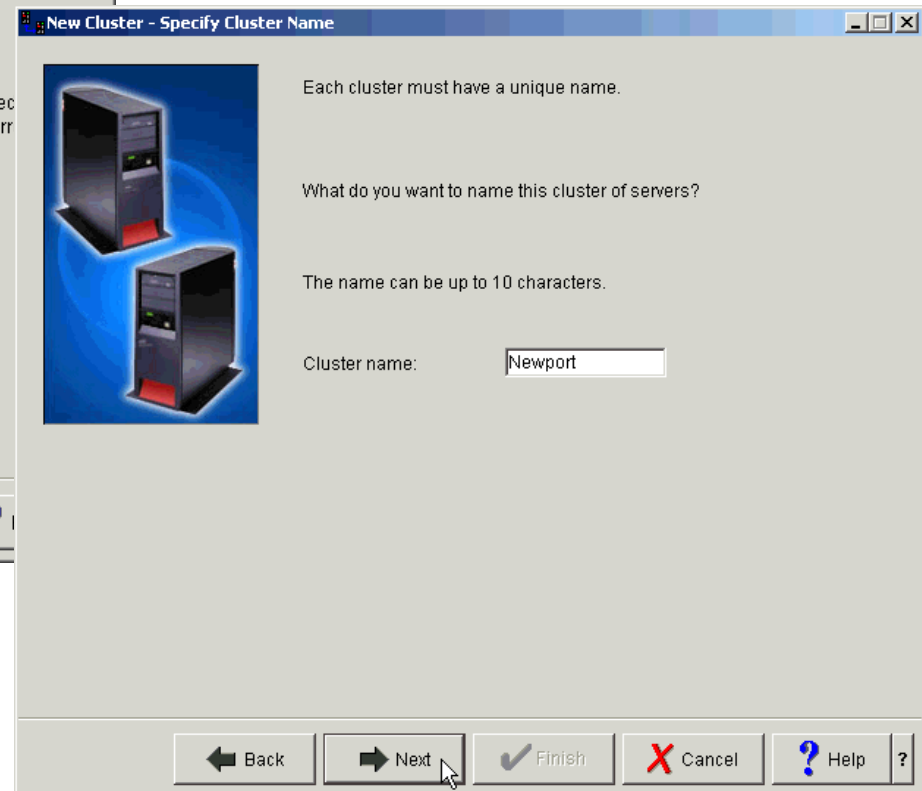
Name	Signed On User	Description
Rchascl1		Manage this server.
Rchascl2		Manage this server.
Rchascl3		Manage this server.
Rchascl4		Manage this server.
Rchascl5		Manage this server.
Rchascl6		Manage this server.
Rchascl7		Manage this server.
Rchascl8		Manage this server.

At the bottom of the window, a status bar reads: "Starts the New Cluster wizard."

Notes Create a cluster



Cluster wizard



Notes Cluster wizard



Cluster node and IP



New Cluster - Specify Node

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

What is the primary node in the cluster?

Node name:

Server:

Cluster interface IP address 1:

Cluster interface IP address 2:

Notes Cluster node and IP



Backup Node signon



New Cluster - Specify Backup Node

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

Yes
 No

Backup node

Node name: rchascl3 Browse...

Server: rchascl3 Browse...

System: RCHASCL3 Browse...

User ID: QSECOFR

Password: ***** Browse...

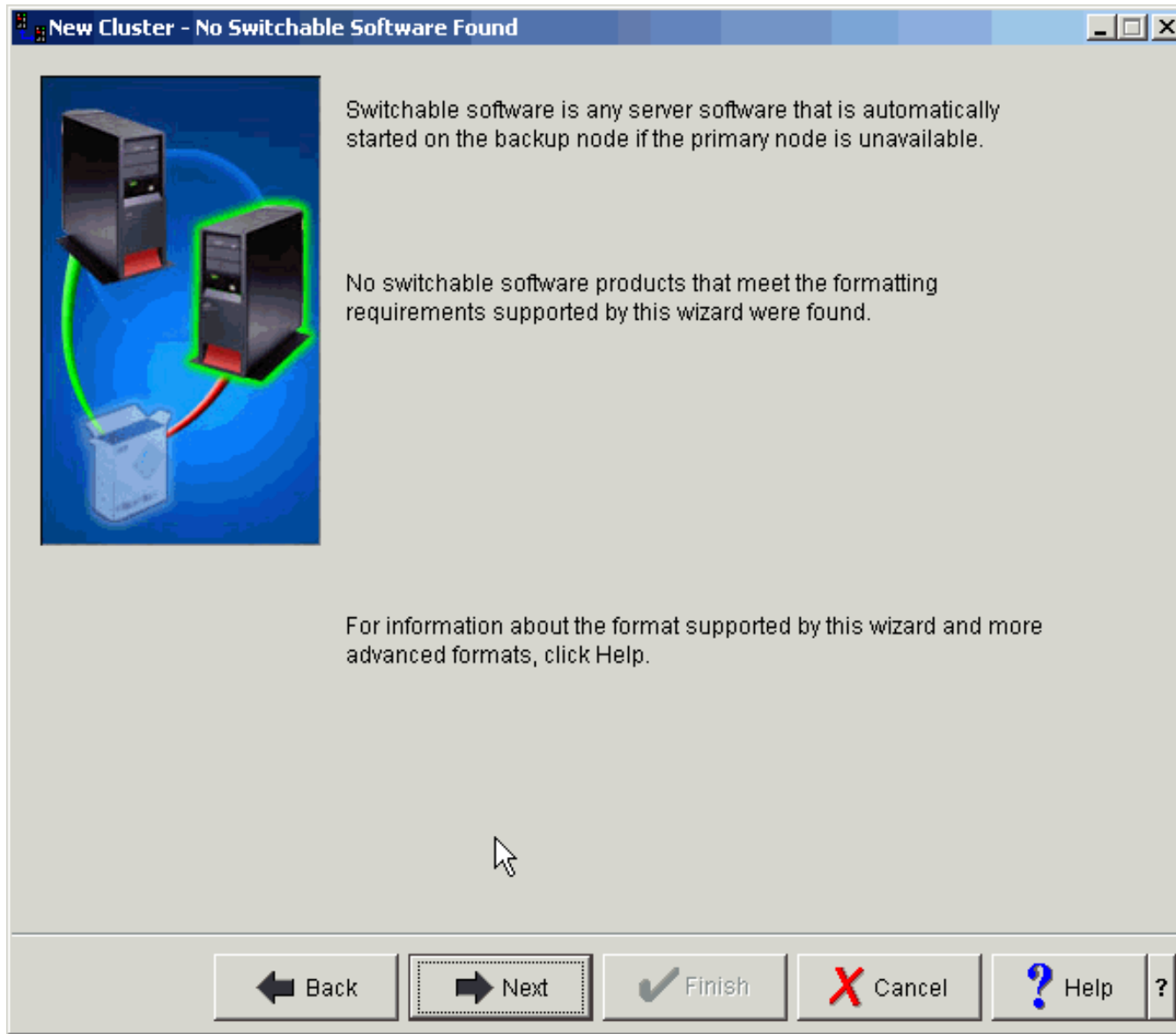
OK Cancel

Back Next Finish Cancel Help ?

Notes Backup Node signon



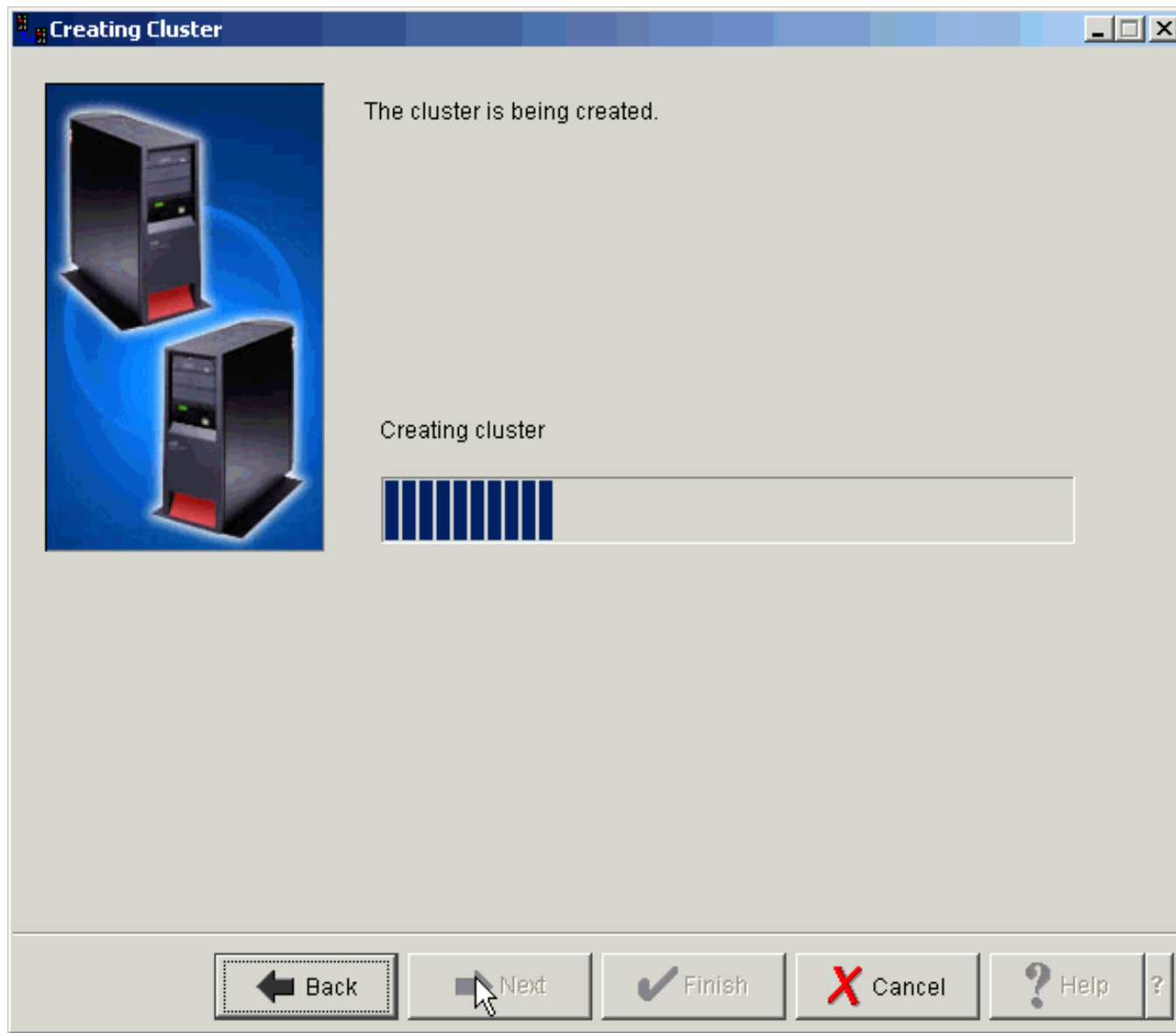
Check for switchable resources



Notes Check for switchable resources



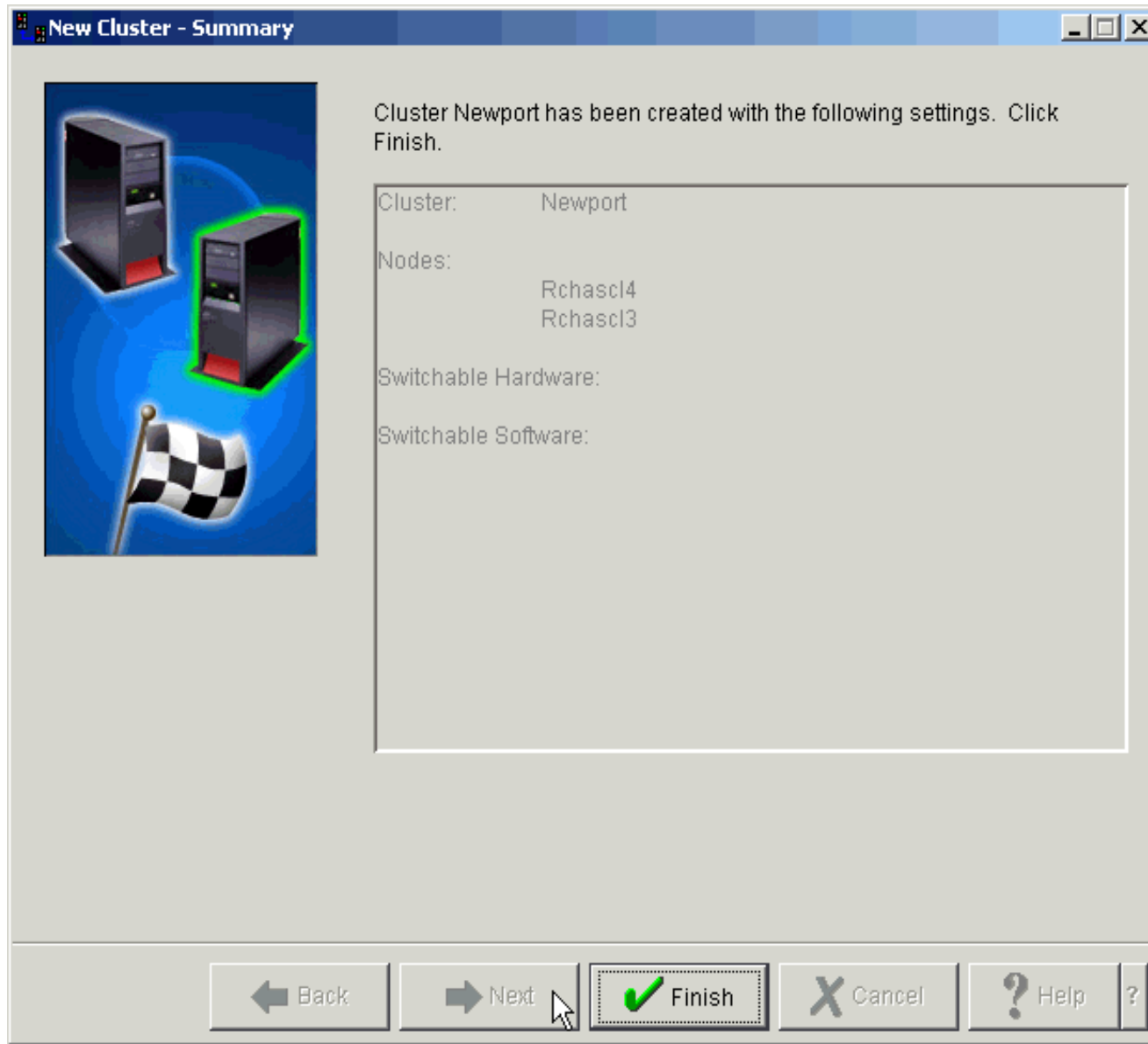
Cluster creation



Notes Cluster creation



Creation summary



Notes Cluster summary



Review cluster



Central System: Rchascl4

Clusters: Newport

Name	Description
Nodes	Configure and manage cluster nodes
Switchable Hardware	Configure and manage cluster hardware
Switchable Software	Configure and manage cluster applicati...
Switchable Data	Configure and manage cluster data

Notes Review cluster



Create switchable group



Central System: Rchascl4

Newport: Nodes

Node	Status	Address 1	Address 2
Rchascl4	Started	17.17.17.4	10.10.10.4
Rchascl3	Started	10.10.10.3	9.5.123.68

Management Central (Rchascl4)

- Task Activity
- Scheduled Tasks
- Definitions
- Monitors
- Endpoint Systems
- System Groups
- Extreme Support
- Systems with Partitions
- Clusters
 - Newport
 - Nodes
 - Switchab...
 - Switchab...
 - Switchab...

My Tasks

- Add a connection
- Install additional com...

Management Central tasks

New Group...

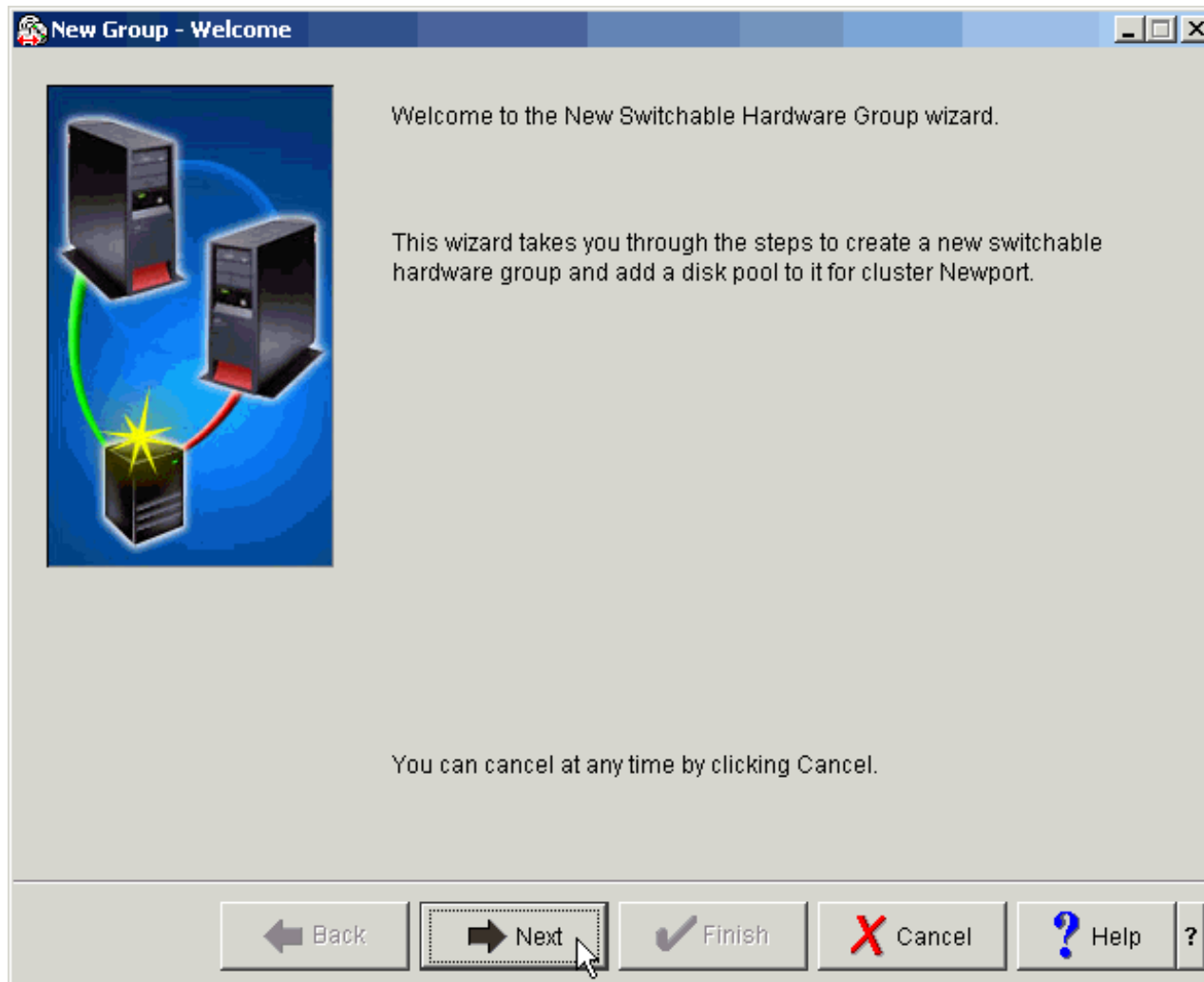
Properties

Add a switchable hardware group to the selected cluster.

Notes Create switchable group



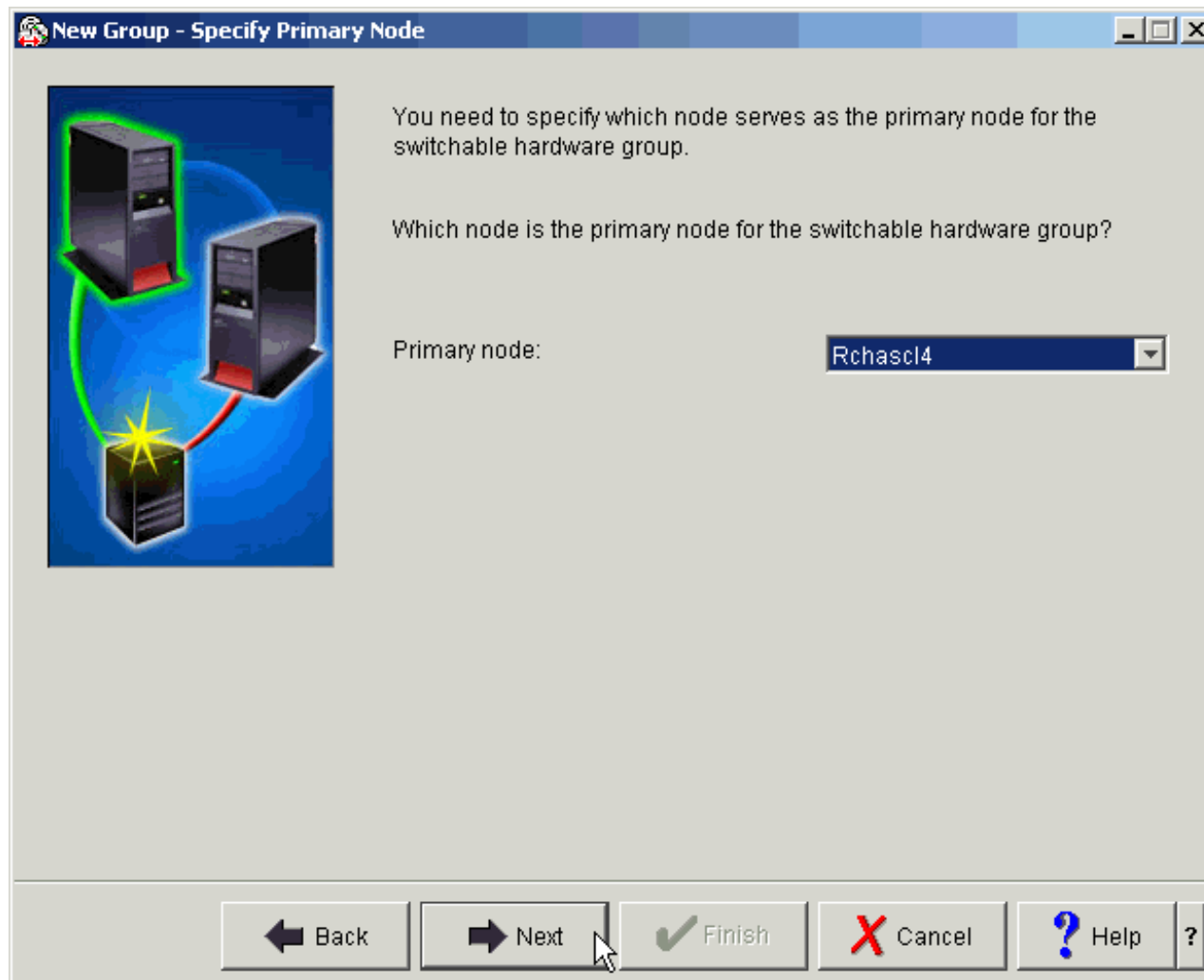
New Group wizard



Notes New Group Wizard



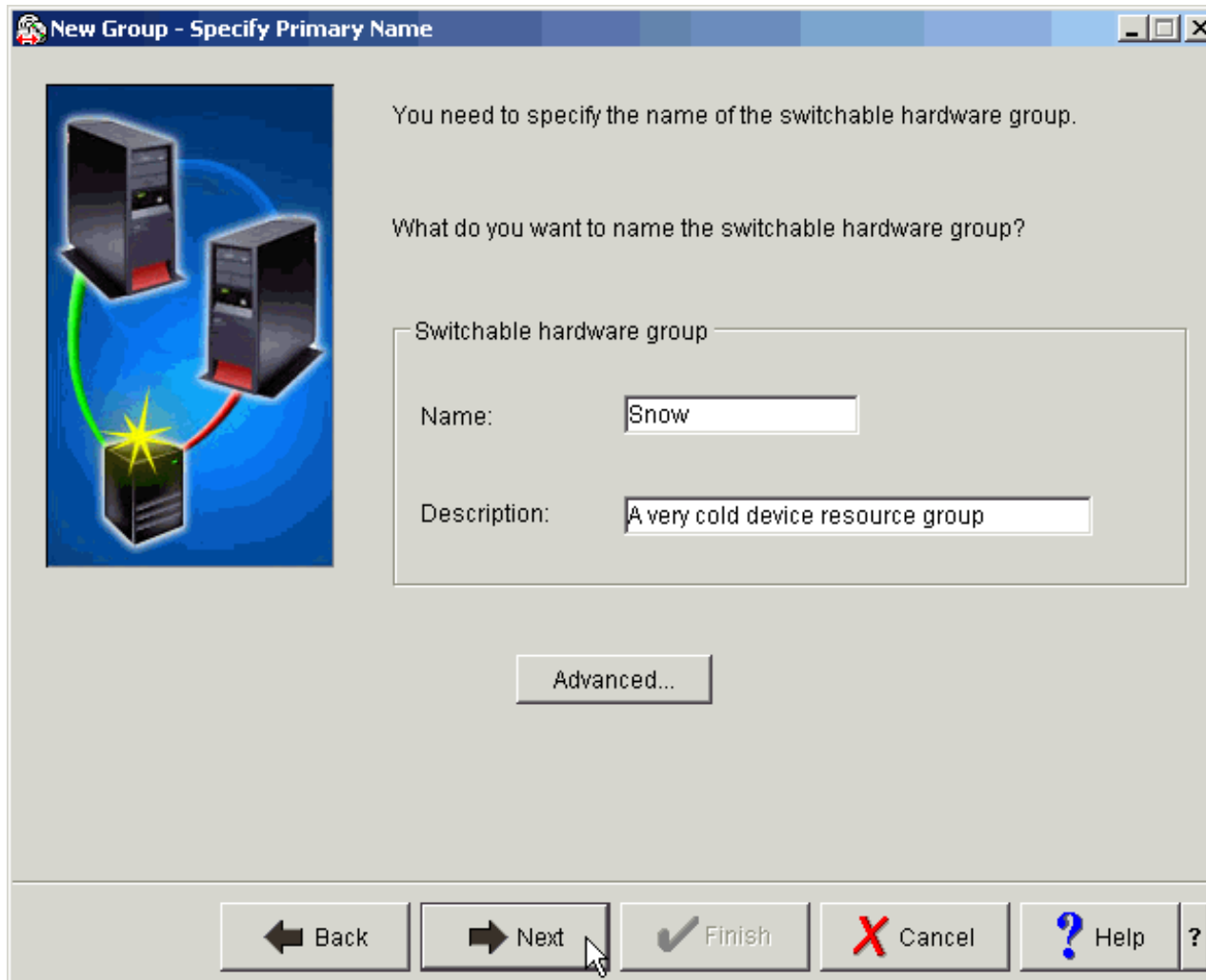
Primary Node



Notes Primary node



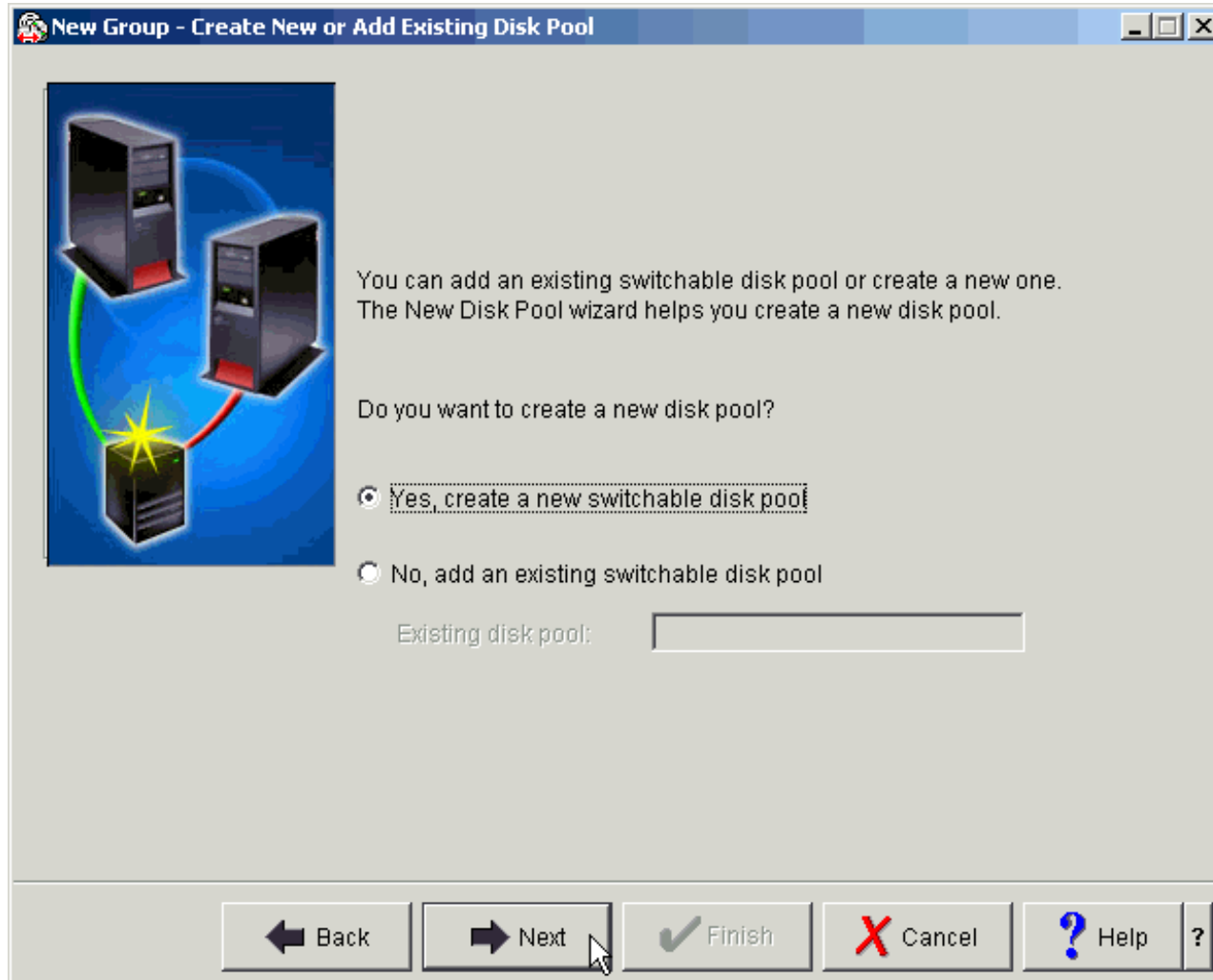
Group Name



Notes Group name



Create or add to pool



Notes Create or Add pool



Select Pool and database



New Group - Disk Pool Type

Select the type of the new disk pool:

UDFS
Name:

Primary
Name:
Database name:
IP Address:

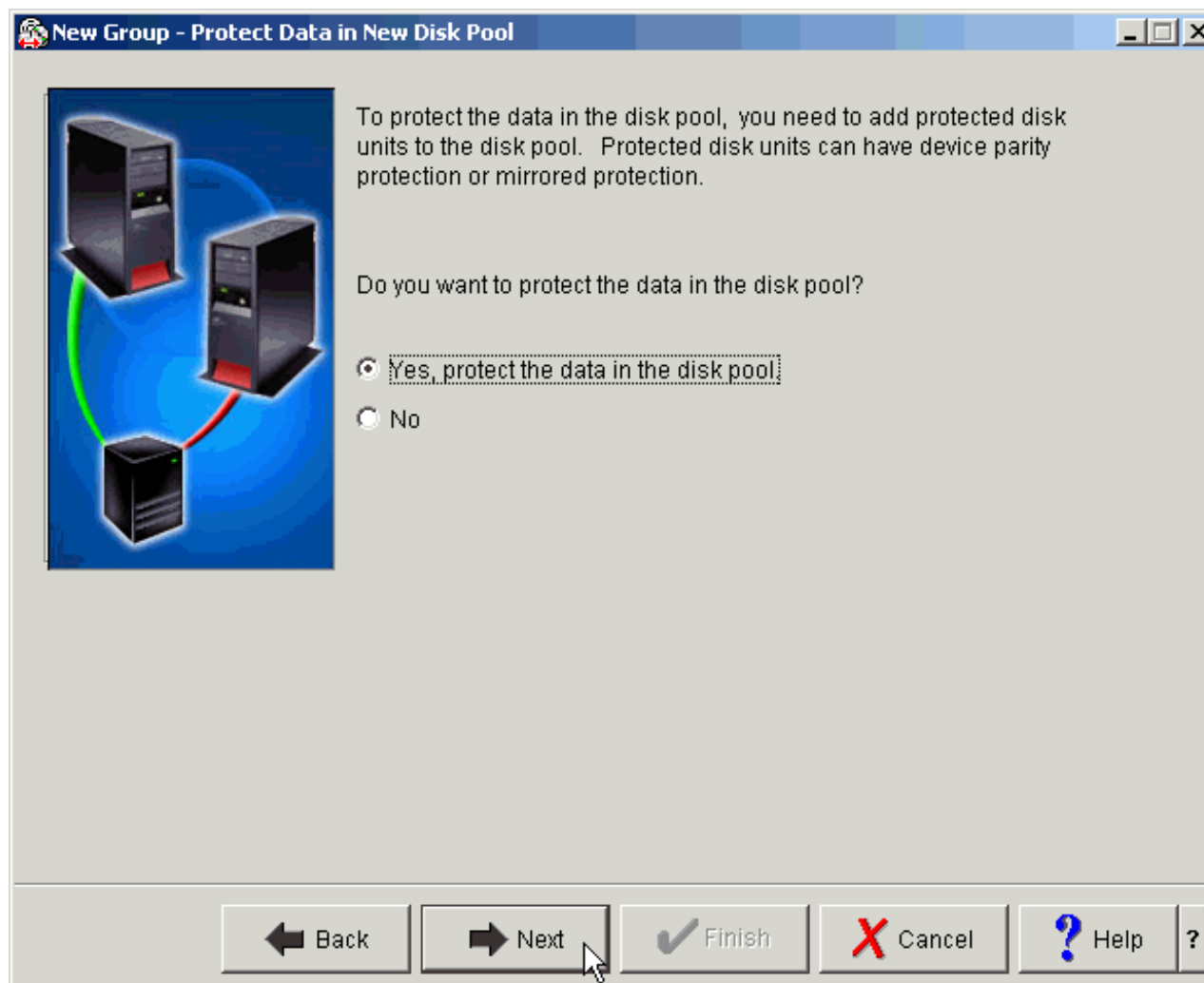
Secondary
Name:
Name of Primary:

← Back → Next ✓ Finish ✗ Cancel ? Help ?

Notes Select pool and database



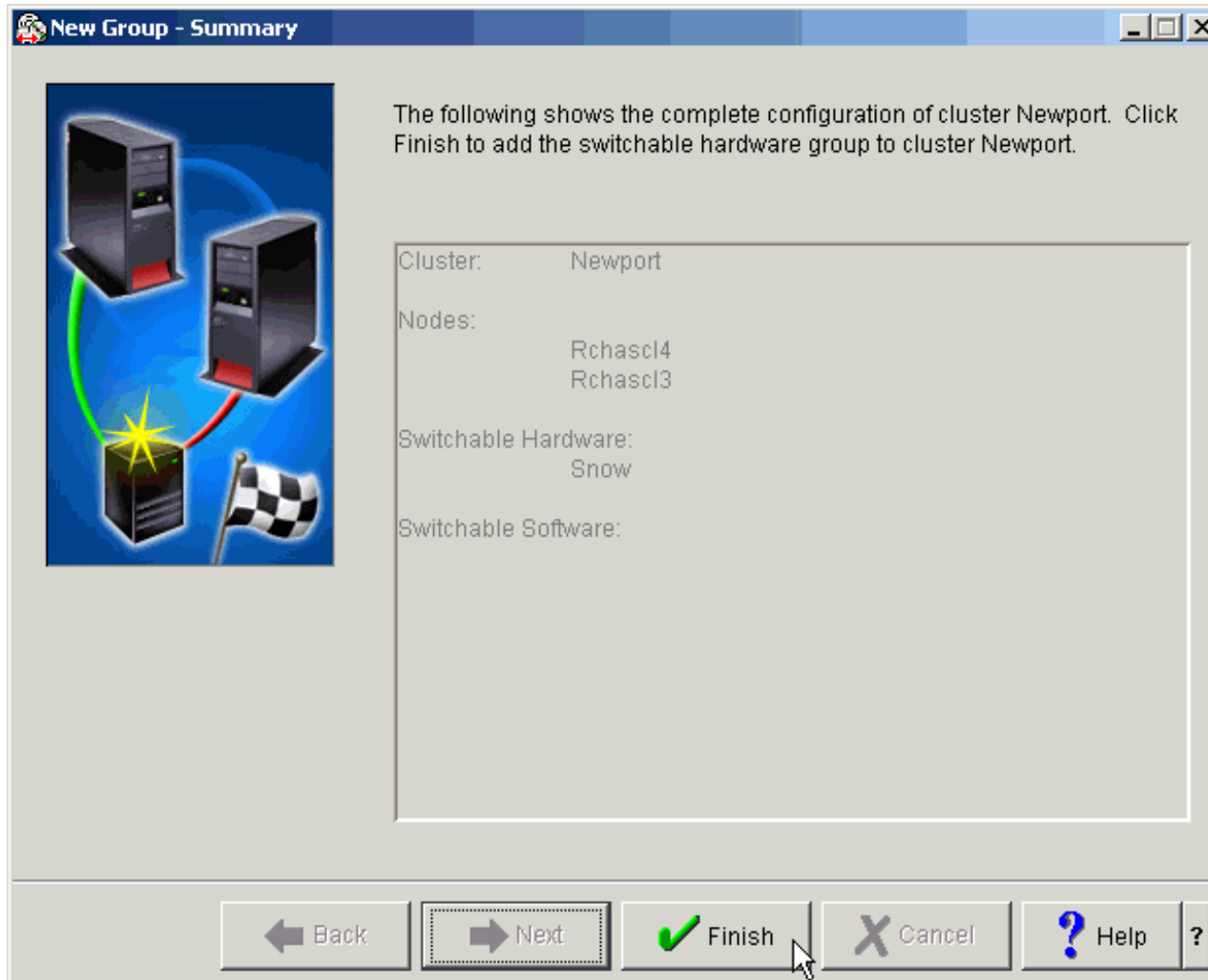
Pool Protection



Notes Pool Protection



Switchable hardware summary



The following shows the complete configuration of cluster Newport. Click Finish to add the switchable hardware group to cluster Newport.

Cluster:	Newport
Nodes:	Rchascl4 Rchascl3
Switchable Hardware:	Snow
Switchable Software:	

Navigation buttons: Back, Next, Finish, Cancel, Help

Detailed description: This is a screenshot of a Windows-style dialog box titled "New Group - Summary". On the left side, there is a graphic showing three server racks connected by a network, with a checkered flag and a starburst effect, indicating a successful or final configuration. The main text area contains a summary of the configuration for a cluster named "Newport". The summary lists the nodes as "Rchascl4" and "Rchascl3", the switchable hardware as "Snow", and the switchable software as empty. At the bottom of the dialog, there are five buttons: "Back" (disabled), "Next" (disabled), "Finish" (active, with a green checkmark), "Cancel" (disabled), and "Help" (disabled).

Notes Switchable hardware summary





Additional information

Key Interfaces



DASD Configuration

ASP Device Processing

- CRTDEVASP
- CHGDEVASP
- DSPDEV
- VRYCFG

Name Space Processing

- SETASPGRP
- QYASPOL

Device CRG Processing

- I/Fs to create cluster, add nodes, set up device domain, etc.
- CRTCRG
- ADDCRGDEVE
- CHGCRGDEVE
- CHGCRGPRI (switch over)

Library Processing

- CRTLIB
- QGYOLOBJ
- QHSMMOVL

Notes



SETASPGRP Command



SETASPGRP

Command Keywords

ASPGRP(*CURUSR,*NONE, name)

SYSLIBL(*CURRENT, *SYSVAL)

CURLIB(*CURRENT, *CURUSR,*CRTDFT, name)

USRLIBL(*CURRENT, *CURUSR, *SYSVAL,
*NONE, names)

Notes

