IBM Parallel System Support Programs for AIX

Managing Shared Disks

Version 3 Release 1

SA22-7349-00

IBM Parallel System Support Programs for AIX

Managing Shared Disks

Version 3 Release 1

SA22-7349-00

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page xiii.

First Edition (October 1998)

This edition applies to Version 3 Release 1 of the IBM Parallel Systems Support Programs for AIX (PSSP) Licensed Program, program number 5765-D51, and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for your comments may be provided at the back of this publication., or you may address your comments to the following address:

International Business Machines Corporation
Department 55JA, Mail Station P384
522 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+914+432-9405
FAX (Other Countries):
Your International Access Code +1+914+432-9405

IBMLink (United States customers only): IBMUSM10(MHVRCFS)
IBM Mail Exchange: USIB6TC9 at IBMMAIL
Internet e-mail: mhvrcfs@us.ibm.com
World Wide Web: http://www.rs6000.ibm.com

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

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

Contents

Figures

Tables

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
522 South Road
Poughkeepsie, NY 12601-5400
USA
Attention: Information Request

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

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX
DATABASE 2
DB2
ES/9000
ESCON
IBM
IBMLink
LoadLeveler
POWERparallel
POWERserver
POWERstation
RS/6000

RS/6000 Scalable POWERparallel Systems
Scalable POWERparallel Systems
SP
System/370
System/390
TURBOWAYS

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

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

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

Publicly Available Software

PSSP includes software that is publicly available:

expect	Programmed dialogue with interactive programs
Kerberos	Provides authentication of the execution of remote commands
NTP	Network Time Protocol
Perl	Practical Extraction and Report Language
SUP	Software Update Protocol
Tcl	Tool Command Language
TclX	Tool Command Language Extended
Tk	Tcl-based Tool Kit for X-windows

This book discusses the use of these products only as they apply specifically to the RS/6000 SP system. The distribution for these products includes the source code and associated documentation. (Kerberos does not ship source code.) **/usr/lpp/ssp/public** contains the compressed **tar** files of the publicly available software. (IBM has made minor modifications to the versions of Tcl and Tk used in the SP system to improve their security characteristics. Therefore, the IBM-supplied versions do not match exactly the versions you may build from the compressed **tar** files.) All copyright notices in the documentation must be respected. You can find version and distribution information for each of these products that are part of your selected install options in the **/usr/lpp/ssp/README/ssp.public.README** file.

About This Book

This book describes the shared disk management facilities of the Scalable POWERparallel Systems (SP) — the IBM Virtual Shared Disk, Hashed Shared Disk, and Recoverable Virtual Shared Disk optional components of the IBM Parallel System Support Programs for AIX (PSSP) program product. These components help you manage your SP's disks so you can let multiple nodes share the information they hold. The book includes an overview of the components and explains how to plan for them, install them, and use them to add reliability to your data storage.

For a list of related books and information about accessing online information, see the bibliography in the back of the book.

This book applies to the planning for, migration to, installation, and use of the optional shared disk management components of PSSP Version 3 Release 1. To find out what version of PSSP is running on your control workstation (node 0), enter the following:

```
splst_versions -t -n0
```

In response, the system displays something similar to:

```
0 PSSP-3.1
```

To find out what version of PSSP is running on the nodes of your system, enter the following from your control workstation:

```
splst_versions -t -G
```

In response, the system displays something similar to:

```
1 PSSP-3.1
2 PSSP-3.1
7 PSSP-2.4
8 PSSP-2.2
```

If you are running mixed levels of PSSP, be sure to maintain and refer to the appropriate documentation for whatever versions of PSSP you are running.

Who Should Use This Book

This book is intended for systems analysts, planners, installers, programmers, and admistrators of SP systems who want to have the data on the disks shared by multiple nodes in the SP system. Use this book if you are considering using the IBM Virtual Shared Disk, Hashed Shared Disk, or Recoverable Virtual Shared Disk optional components of PSSP to help you manage the sharing of data on your disks.

You can use the information in this book to plan for, install, and use these shared disk components to manage the sharing of information on your disks and to write application programs that use virtual shared disks.

It assumes that you are, and it is particularly important that you be, experienced with and understand the AIX operating system, especially the Logical Volume

Manager subsystem, as explained in the book *IBM AIX Version 4 System Management Guide: Operating System and Devices* or *IBM AIX Version 4.3 System Management Guide: Operating System and Devices*.

You need to be familiar with and might need the documentation for the following additional information:

- The PSSP software system.
- Planning information in *RS/6000 SP Planning, Volume 1, Hardware and Physical Environment* and in *RS/6000 SP Planning, Volume 2, Control Workstation and Software Environment*.
- The installation and configuration information that came with your twin-tailed disk hardware.
- Installation procedures for this release in *PSSP: Installation and Migration Guide*.
- The virtual shared disk commands in *PSSP: Command and Technical Reference*.
- Information that is useful for programming your applications that use virtual shared disks in *AIX Kernel Extensions and Device Support Programming Concepts*.
- Information that is useful for system performance in *AIX Performance and Tuning Guide*.
- Information about the availability services used by the IBM Recoverable Virtual Shared Disk subsystem in *RS/6000 Cluster Technology: Group Services Programming Guide and Reference*.
- Diagnosis information in *PSSP: Diagnosis Guide*.
- The messages in *PSSP: Messages Reference*.

Typographic Conventions

This book uses the following typographical conventions:

Typographic	Usage
Bold	• **Bold** words or characters represent system elements that you must use literally, such as commands, flags, and path names.
Italic	• *Italic* words or characters represent variable values that you must supply. • *Italics* are also used for book titles and for general emphasis in text.
`Constant width`	Examples and information that the system displays appear in `constant width` typeface.
[ ]	Brackets enclose optional items in format and syntax descriptions.
{ }	Braces enclose a list from which you must choose an item in format and syntax descriptions.
|	A vertical bar separates items in a list of choices. (In other words, it means “or.”)
< >	Angle brackets (less-than and greater-than) enclose the name of a key on the keyboard. For example, <**Enter>** refers to the key on your terminal or workstation that is labeled with the word Enter.
...	An ellipsis indicates that you can repeat the preceding item one or more times.
<**Ctrl-*x*>**	The notation <**Ctrl-*x*>** indicates a control character sequence. For example, <**Ctrl-c>** means that you hold down the control key while pressing <**c>**.
→	An arrow (→) is a shorthand notation for what to click on next, which might be in the same window or the next window that appears.

The term *SP* or *SP system* refers to the current system partition, if you have partitioned your system. Otherwise, the term refers to the physical SP system.

Chapter 1. Overview of the Managing Shared Disk Components of PSSP

This chapter highlights what is new in this release and then gives a basic overview of each of the components of PSSP that help you manage the sharing of information on your SP system's disks. Later chapters give more information about the components and how to use them to manage your virtual shared disks.

The PSSP components that help you create and manage virtual shared disks are:

- IBM Virtual Shared Disk, the subsystem that provides the function so you can create and manage virtual shared disks along with a device driver that operates between your applications that use the virtual shared disks and the AIX Logical Volume Manager (LVM)
- Hashed Shared Disk, the subsystem that works with the IBM Virtual Shared Disk subsystem, offering a data striping device driver for your virtual shared disks
- IBM Recoverable Virtual Shared Disk, the subsystem that provides recoverability of your virtual shared disks if a node, adapter, or disk failure occurs

Also in this chapter is an introduction to the IBM Virtual Shared Disk Perspective graphical user interface for managing virtual shared disks.

What's New in This Release?

The following improvements have been made:

- The book has been revised and reorganized to enhance existing topics.
- The IBM Recoverable Virtual Shared Disk component can dynamically refresh the IBM Virtual Shared Disk subsystem. This means that nodes and virtual shared disks can be added to or removed from an active IBM Virtual Shared Disk configuration without having to stop all applications and unconfigure the existing virtual shared disks.
- Enhancements have been made to coexistence support. If you do maintain multiple levels of the PSSP or IBM Recoverable Virtual Shared Disk software along with PSSP 3.1 in the same system partition, you can use the new **rvsdrestrict** command to restrict the level for which the IBM Recoverable Virtual Shared Disk subsystem will run.
- Enhancements have been made to interface with the Event Management subsystem of PSSP to allow virtual shared disks to be monitored. You can use the new **monitorvsd** command to enable monitoring of virtual shared disks on a per node basis.
- Many enhancements have been made to the IBM Virtual Shared Disk Perspective graphical user interface in support of most of the new and existing virtual shared disk functions. For a list of these enhancements, see “What's New in the IBM Virtual Shared Disk Perspective?” on page 16.

IBM Virtual Shared Disk Component Overview

IBM Virtual Shared Disk is a subsystem that lets application programs executing on different nodes of a system partition access a raw logical volume as if it were local at each of the nodes. See Figure 1 for an illustration of a simplified virtual shared disk implementation. Each virtual shared disk corresponds to a logical volume that is actually local at *one* of the nodes, which is called the *server* node. The IBM Virtual Shared Disk subsystem routes I/O requests from the other nodes, called *client* nodes, to the server node and returns the results to the client nodes.

The I/O routing is done by the IBM Virtual Shared Disk device driver that interacts with the AIX Logical Volume Manager (LVM). The device driver is loaded as a kernel extension on each node. Thus, raw logical volumes can be made globally accessible.

The application program interface to a virtual shared disk is the raw device (or device special file). This means application programs must issue requests to a virtual shared disk in the block size specified by the LVM (currently, requests are multiples of 512 bytes on 512-byte block boundaries).

You can find more information on logical volumes in *AIX System Management Guide: Operating Systems and Devices*. See the list included in “Who Should Use This Book” on page xv for order numbers.

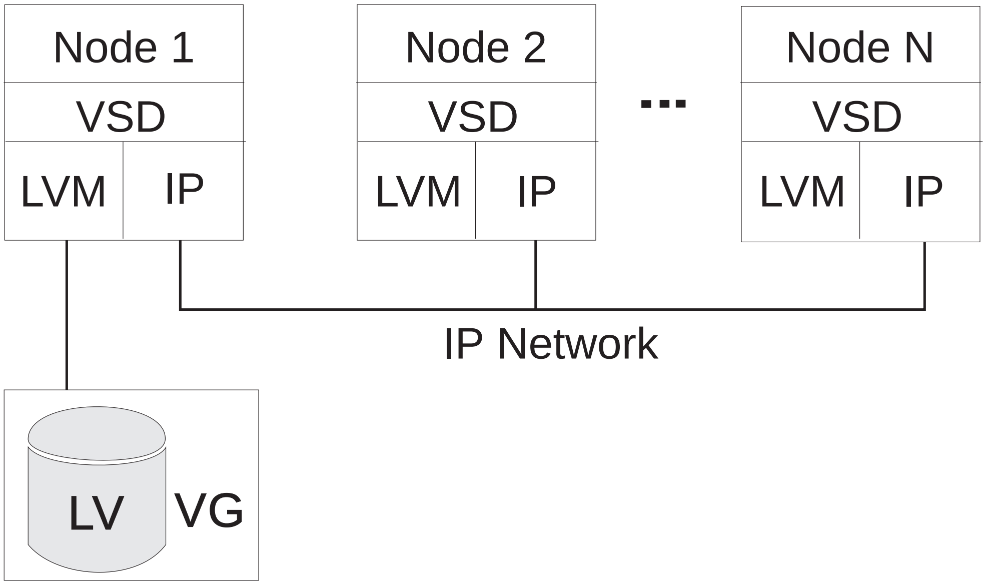


Figure 1. An IBM Virtual Shared Disk IP Network Implementation

IBM Virtual Shared Disk Restrictions

- IBM suggests that you do not create AIX Journaled File Systems (JFSs) on volume groups that contain virtual shared disks because they can interfere with recoverability. Even if you do not currently plan to use the IBM Recoverable Virtual Shared Disk component, you might want to in the future.
- The maximum number of virtual shared disks that can be defined in a system partition is 32767 (32K).

- The maximum size of a virtual shared disk is 1024 GB (one terabyte).
- Restrictions on AIX logical volumes also affect virtual shared disks; for example:
  - The maximum number of physical volumes in a volume group is 128.
  - The maximum number of physical partitions on a disk is 1016.
  - The maximum number of logical volumes in a volume group is 512.

Hashed Shared Disk Component Overview

The Hashed Shared Disk component has a data striping device driver that distributes data across multiple nodes and multiple virtual shared disks, thus reducing I/O bottlenecks. Instead of writing all the data from one application program I/O request onto one virtual shared disk at a specific location, the data striping device driver writes blocks of the data on each of several separate virtual shared disks.

Figure 2 illustrates data striping across two or more virtual shared disks. In this case, a write request to HSD1 is made by an application program running on client node 4. When you create a hashed shared disk, the virtual shared disks that comprise the hashed shared disk are created as well. They are then collectively known as a hashed shared disk though individually, they are still virtual shared disks. At the time you create the hashed shared disk you specify certain operational parameters, among which is the *stripe size*. The stripe size is the amount of data that you want as a stripe (or block) to be written as one unit.

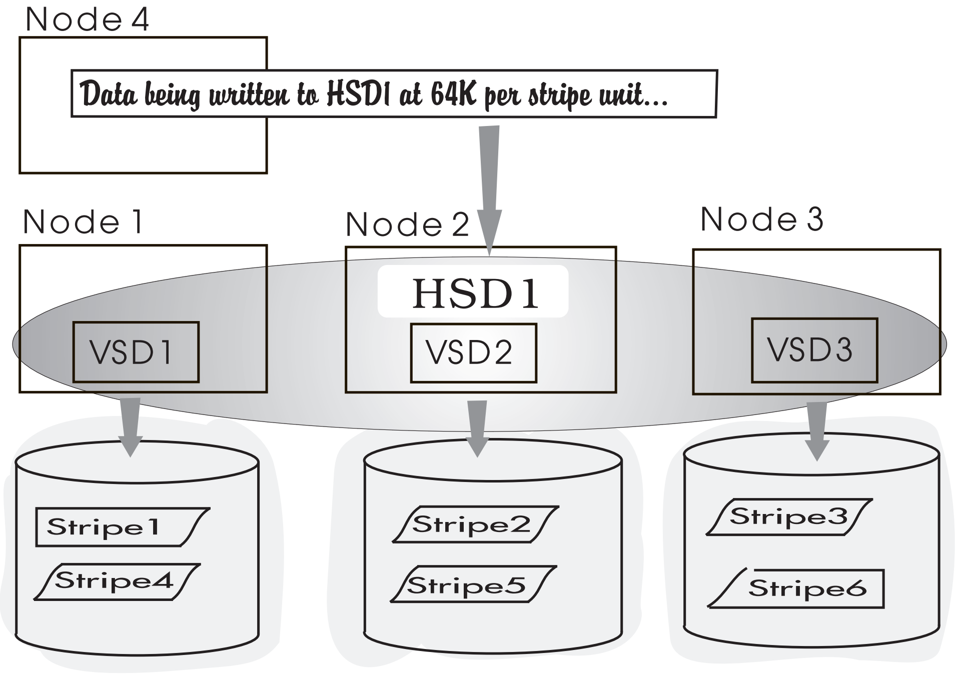


Figure 2. Hashed Shared Disk Stripes Data Across Multiple Virtual Shared Disks

The principal value of the Hashed Shared Disk component is that it provides distribution of data across physical disks and nodes while being transparent to the application program using the virtual shared disks.

Comparing Hashed Shared Disk and LVM Striping

Data striping capabilities were introduced in the Logical Volume Manager (LVM) component of AIX. A logical volume can be striped across two or more physical disks. The Hashed Shared Disk component of PSSP can stripe data across two or more virtual shared disks and nodes (a virtual shared disk corresponds to a single logical volume).

The Hashed Shared Disk component can stripe data across multiple nodes, while the LVM striping function is limited to local physical disks. The Hashed Shared Disk component allows you to use mirroring while LVM striping does not.

If you do not use mirroring, IBM suggests that you use the LVM function for local striping and the Hashed Shared Disk component of PSSP for global striping. For example, if you have two physical disks per node on a 10-node system, and you want to stripe the data across all 20 physical disks, do the following:

1. On each node, create a local logical volume that spans both physical disks
2. Create 10 virtual shared disks that point to these 10 logical volumes
3. Create one hashed shared disk that spans all 10 virtual shared disks

You now have one hashed shared disk that spans 10 virtual shared disks and each virtual shared disk correlates to a logical volume that is striped across two physical disks.

If you do use mirroring, you can use the Hashed Shared Disk component to stripe a local mirrored disk.

Refer to the book *AIX Performance and Tuning Guide*, for more information on how to tune striped logical volumes.

When to Use Hashed Shared Disk

Whether or not to use a hashed shared disk depends on your configuration of virtual shared disks and the I/O characteristics of your application programs.

If the I/O load to a specific virtual shared disk is too heavy, you can use a hashed shared disk to distribute the load to other virtual shared disks and nodes.

When you plan your application, you should be able to determine whether the bandwidth required for particular virtual shared disks is higher than the system supports. If so, you should consider using hashed shared disks.

Hashed Shared Disk Restrictions

- Striping is supported across virtual shared disks only.
- The function **lseek (SEEK_END)** is not supported because the file might be composed of several virtual shared disks.
- All the underlying virtual shared disks in one hashed shared disk must be the same size.
- The size of each virtual shared disk in one hashed shared disk must be a multiple of the stripe size in the hashed shared disk. For example, if the stripe size is 32KB, the virtual shared disk size must be a multiple of 32KB.

- The size of a hashed shared disk (*number_of_vsds* × *size_of_vsd*) cannot be greater than one terabyte (1024 GB).
- After you have written data to a hashed shared disk, do not write data to any of the underlying virtual shared disks.
- The address of your I/O buffer must be aligned on a 4KB boundary. If it is not, the Hashed Shared Disk read and write routines will **not** parallelize I/O requests to the underlying virtual shared disks.

IBM Recoverable Virtual Shared Disk Component Overview

To understand the value of the IBM Recoverable Virtual Shared Disk component, consider a system *without* it. The virtual shared disk function lets all nodes in the system partition access a given disk, even though that specific disk is physically attached to only one node. If the server node should fail, access to the disk is lost until the server node is rebooted by the administrator.

By using the IBM Recoverable Virtual Shared Disk component and twin-tailed disks or disk arrays, you can allow a secondary node to take over the server function from the primary node when certain types of failure occur.

A *twin-tailed disk* is a disk or group of disks that are attached to two nodes of an SP. For recoverability purposes, only one of these nodes serves the disks at any given time. The secondary or backup node provides access to the disks if the primary node fails, is powered off, or if you need to change the server node temporarily for administrative reasons. Both must be in the same system partition.

A *twin-tailed volume group* is a volume group that contains disks that are accessible to two nodes. Both nodes must be in the same system partition.

The IBM Recoverable Virtual Shared Disk component automatically manages your virtual shared disks by detecting error conditions, such as node failures, adapter failures, and disk failures (EIO errors), and then switching access to the disk from the primary node to the secondary node so that your application can continue to operate normally. The IBM Recoverable Virtual Shared Disk component also allows you to cut off access to virtual shared disks from certain nodes and to dynamically change the server node (using the graphical user interface or the **fencevsd** and **vsdchgserver** commands, respectively).

When your applications exploit the IBM Recoverable Virtual Shared Disk component, you can recover more easily from node failures and have continuous access to the data on the twin-tailed disks.

Virtual Shared Disk Recovery with Twin-tailed Disks

The following three figures show a simple system with one twin-tailed recoverable virtual shared disk configuration. Figure 3 on page 6 shows the basic configuration. The primary node is the *server* of information on the disk. In Figure 4 on page 6, the secondary node is acting as the server following a failure of the primary node.

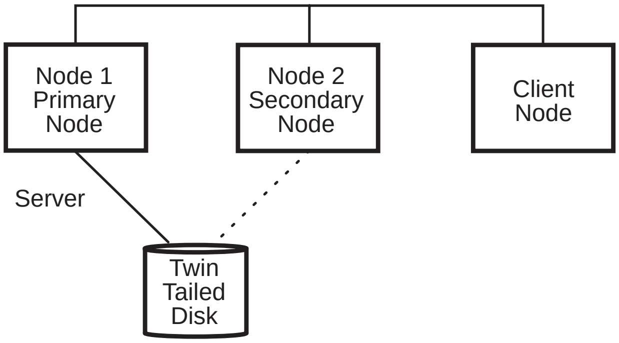


Figure 3. Simplified View of a Twin-tailed Disk

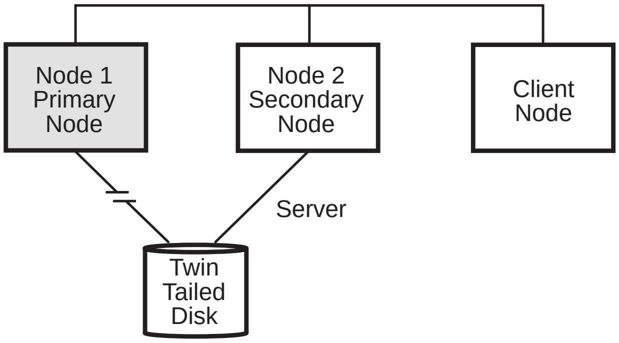


Figure 4. The Secondary Node Serves after a Primary Node Failure

In Figure 5, the primary node is again the server and has automatically taken over the disk from the secondary node.

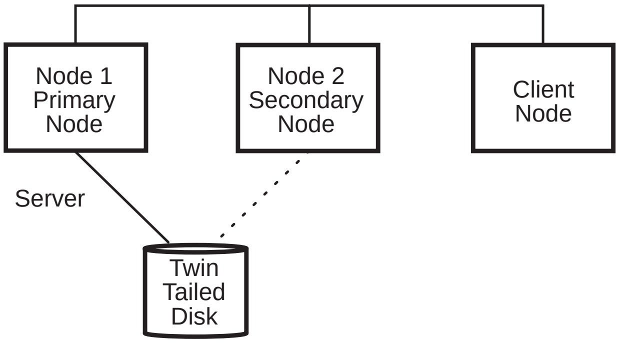


Figure 5. The Primary Node Is the Server Again after Recovery

Recovery is transparent to applications that have been enabled for recovery — there is no disruption of service, only a slight delay while takeover occurs. The IBM Recoverable Virtual Shared Disk component provides application programming interfaces, including recovery scripts and C programs, so you can enable your applications to be recoverable.

IBM Recoverable Virtual Shared Disk Restrictions

- Do not put the root volume group (**rootvg**) or any other volume group that contains bootable logical volumes on a twin-tailed disk.
- Do not use twin-tailed volume groups with non-virtual shared disk applications. Applications that access data on non-virtual shared disk logical volumes (a JFS file system, for example) will lose access to the logical volume when the virtual shared disk server node changes and can interfere with recovery.
- You cannot have recoverable and nonrecoverable virtual shared disk nodes in a single system partition. When you use the IBM Recoverable Virtual Shared Disk component, it must be installed on the control workstation and on all the nodes in the system partition that are to have or use virtual shared disks.
- Do not put a tape drive and a twin-tailed disk on the same SCSI bus.

IBM Virtual Shared Disk Perspective Overview

IBM Virtual Shared Disk Perspective is the graphical user interface of PSSP that helps you perform shared disk management tasks without you having to remember the commands and their syntax. You can view, set, or change attributes and status, and you can control and monitor the operation of the shared disk management components. The IBM Virtual Shared Disk Perspective graphical user interface has the basic actions that let you do most of your shared disk management work from within the graphical session.

Each action correlates to one or more virtual shared disk commands. You can run virtual shared disk, other PSSP, and AIX commands from within this interface as well. SMIT is also available to you for managing shared disks. Appendix A, "Interface Cross-Reference" on page 93 has a cross-reference chart including basic functions and how to perform them using each of the interfaces. It also includes a summary of the virtual shared disk commands, showing the command name and its purpose. The full path for all commands involved with managing virtual shared disks is **/usr/lpp/csd/bin/** and the path for other PSSP commands is **/usr/lpp/ssp/bin/**. The syntax and complete descriptions of all commands in PSSP are in the book *PSSP: Command and Technical Reference*.

Explanation of the IBM Virtual Shared Disk Perspective graphical user interface is limited in this book to the brief introduction in this chapter. In other chapters, you will find only references intended as task guidance and quick-start direction. This book uses an arrow (→) as a shorthand notation for what to click on next, which will be either in the same window or in the next window that appears.

Begin using the interface with either of two PSSP commands:

- Use the **perspectives** command to bring up the SP Perspectives Launch Pad window and then double-click on the **IBM VSD Perspective** icon to open the primary window.
- Use the **spvsd** command to open the IBM Virtual Shared Disk Perspective primary window directly.

Click on **Help→Tasks** at the top right-hand corner of the primary window to see an online help system that explains all that you can do and how you can use the IBM Virtual Shared Disk Perspective interface. There is information about every pane and every field. The book *PSSP: Administration Guide* also has some information about the SP Perspectives graphical user interface.

The following sections serve as an introduction and highlight the key features of only the IBM Virtual Shared Disk Perspective though most of the interface is common to all the SP Perspective interfaces.

The IBM Virtual Shared Disk Perspective Primary Window

Figure 6 shows what the primary window might look like by default, the first time you use the IBM Virtual Shared Disk Perspective.

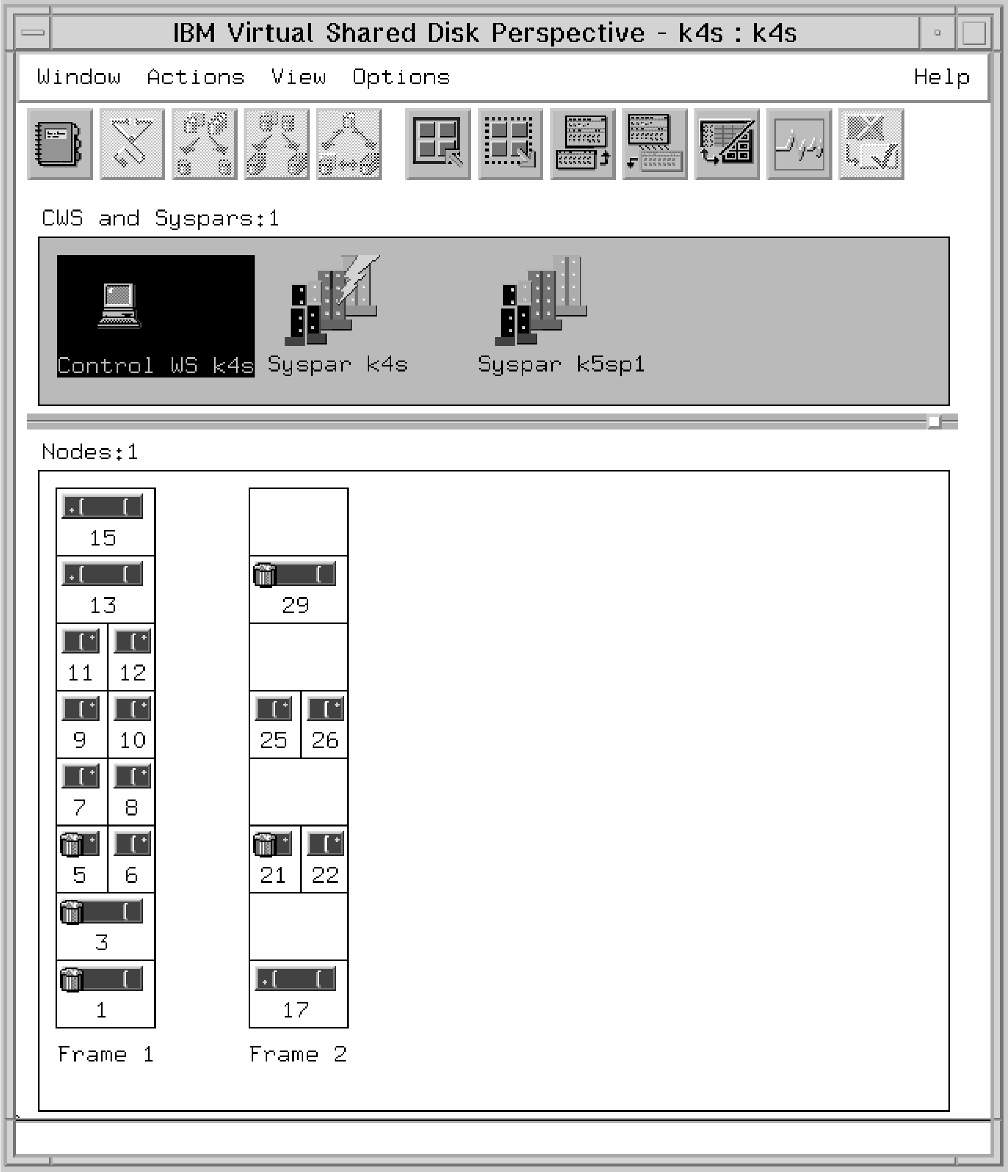


Figure 6. The IBM Virtual Shared Disk Perspective Primary Window

By default, the primary window contains:

- The menu bar
- The tool bar
- Two panes:

  - The CWS and Syspars pane
  - The Nodes pane in the frame view
- The information area

When you expect to work with virtual shared disks, you need to add the IBM VSDs pane and you might want to set the Nodes pane to the icon view to see more nodes in less space. If you want to work with hashed shared disks, you will have to add the IBM HSDs pane also. If you want, you can even remove the CWS and Syspars pane from the window and bring it back only when you need it. Figure 7 shows what the primary window might look like after you change the Nodes view, add an IBM VSDs pane, and add an IBM HSDs pane.

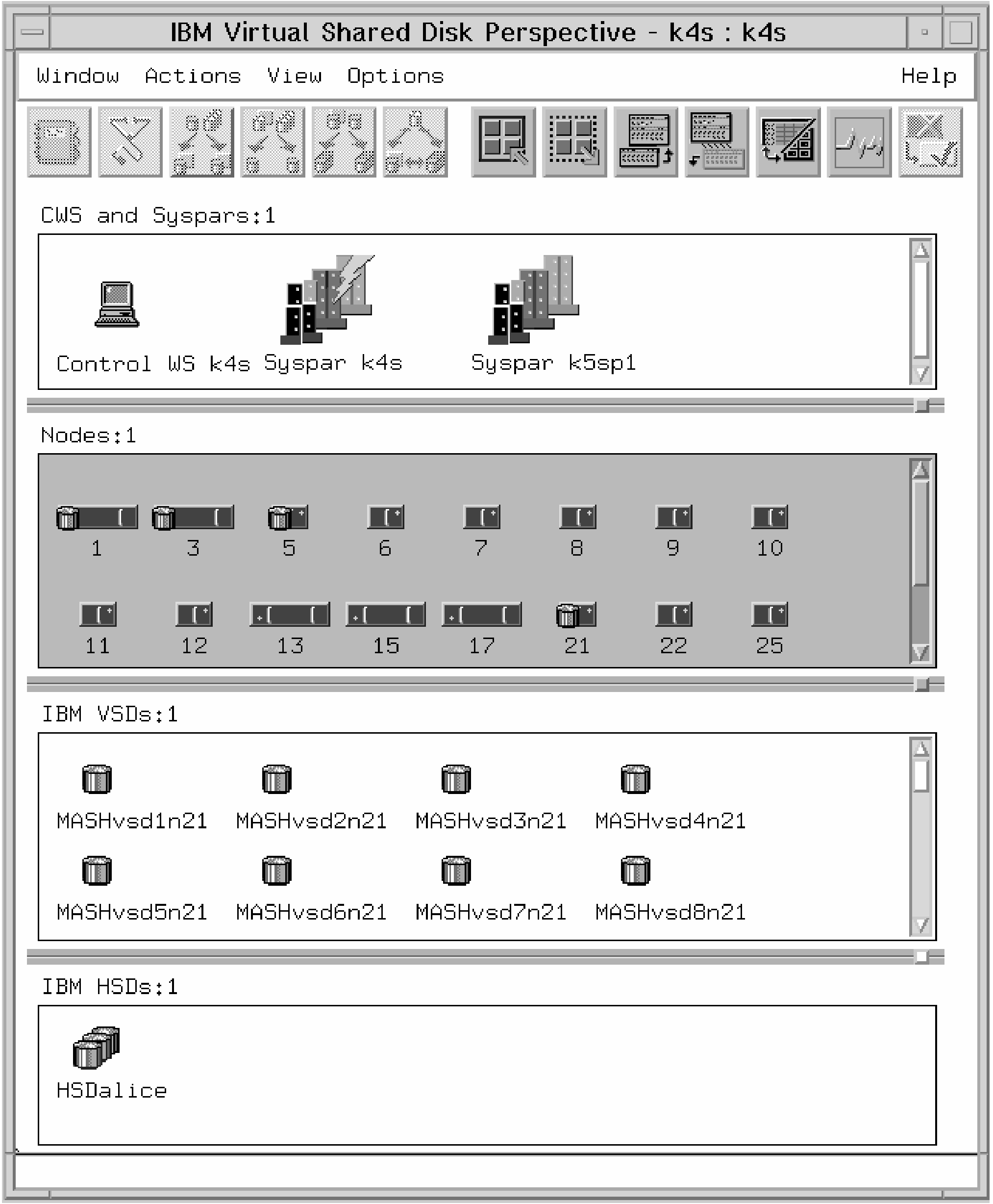


Figure 7. A Customized IBM Virtual Shared Disk Perspective Primary Window

This customized primary window contains:

- The menu bar
- The tool bar
- Panes:
  - The CWS and Syspars pane
  - The Nodes pane (was rearranged)
  - The IBM VSDs pane (was added)
  - The IBM HSDs pane (was added)
- Information area

In general, the actions that are selectable vary based on which pane is current and what object is selected. In other words, this is an object-action oriented interface. First you select what objects you want to work with, then you select what you want to do with, to or for that object.

The Menu Bar

In this area, in addition to the **Help** action , there are four menu choices: **Window**, **Actions**, **View**, and **Options**.

Figure 8. The IBM Virtual Shared Disk Perspective Menu Bar

- From the **Window** menu, you can close the current window or exit the IBM Virtual Shared Disk Perspective.
- From the **Actions** menu, you can select the actions that apply to the pane that is current (CWS and Syspars, Nodes, IBM VSDs, or IBM HSDs).

  If you are experienced with the command line interface or SMIT, you might feel more comfortable with the IBM Virtual Shared Disk Perspective interface by keeping in mind that there is often a correlation between an action, such as the **Create** action from the IBM VSDs pane, and a command, such as **createvsd**. The dialog for the action tends to contain all the same informational and data entry fields as are defined for the related command. An advantage of the graphical user interface is that most of the fields are already set to default values and you can see what they are. Also, it is so much easier and less error prone to select nodes or virtual shared disks from a pane or a selection list than to type long command line strings.

  If you do not see all the actions on your menu, you might not have authorization. See “Establishing Authorization” on page 27.
- From the **View** menu, you can change how the objects in the current pane are displayed. For example; one option lets you see the objects as entries in a table, another lets you sort the entries, you can filter the objects to be shown and set monitoring criteria, you can change system partitions even when you do not have the CWS and Syspars pane displayed.
- From the **Options** menu, you can change characteristics that affect all the windows. For example, you can set fonts, colors, and change from horizontal to vertical panes. The options you set can be saved in a *preference profile.* The windows are arranged according to your preferences each time you specify the preference profile when you start the IBM Virtual Shared Disk Perspective.

For other SP Perspectives menu choices, see the chapter on SP Perspectives in the book *PSSP: Administration Guide*. However, you will find detailed information only when you start the graphical user interface and look at the online help.

The Tool Bar

The tool bar is displayed just below the menu bar in the primary window. The tool bar consists of icons that represent the most frequently used actions from the Actions and View menus. At any given time, an icon is selectable or not, depending on what pane is current and what object is selected.

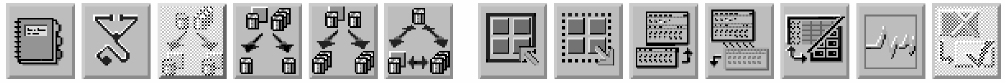


Figure 9. The IBM Virtual Shared Disk Perspective Tool Bar

For the IBM Virtual Shared Disk Perspective, the icons in the tool bar are the following:

Icon	Title and Purpose
	Properties lets you view and modify attributes of each type of object shown in the panes: control workstation, system partitions, nodes, virtual shared disks, and hashed shared disks. This is how you see what is in the System Data Repository (SDR). Depending on the selected object, you can set or change certain values.
	Run diagnostics displays information about the status of virtual shared disks.
	Filter to show related nodes in a new pane opens a new Nodes pane with the nodes that are related to the selected virtual shared disks or hashed shared disks.
	Filter to show related IBM VSDs in a new pane opens a new IBM VSDs pane with the virtual shared disks that are related to the selected virtual shared disk nodes or hashed shared disks.
	Filter to show related IBM HSDs in a new pane opens a new IBM HSDs pane with the hashed shared disks that are related to the selected virtual shared disk nodes or virtual shared disks.
	Bring up the Filter to Show Related Objects dialog brings up a dialog that lets you filter another pane based on objects selected in the current pane. For example: • You can see which virtual shared disks are configured on the selected nodes and filter the list by state (active, suspended, stopped). • You can see which virtual shared disks have the selected nodes as a primary or backup server and you can see which nodes are clients of selected virtual shared disks or hashed shared disks.
	Select all objects selects all the objects that are in the current pane.

Icon	Title and Purpose
	Deselect all selected objects deselects all of the currently selected objects.
	Add a pane lets you add a new pane to the current window or to a new window.
	Delete the current pane from this window lets you remove the current pane from the window.
	Show objects in the table view or the icon view is a toggle between icon view and table view in the current pane.
	Set up and begin monitoring lets you monitor conditions associated with nodes, the control workstation, or system partitions.
	Acknowledge triggered or unknown monitoring state of selected nodes lets you mark an object to acknowledge that you are aware of the nodes triggered or unknown monitoring state. The mark will remain until the state of the node changes.

The Panes

The objects that you can work with are displayed in the panes. Some panes can be seen in any of several arrangements or views, depending on which is the current pane. Some of your view options are pointed out in the discussion of the Nodes pane.

You can add and delete panes. For example, if you want to concentrate on your virtual shared disk nodes, you might delete the CWS and Syspars pane after selecting the current system partition to make more room for the nodes and virtual shared disk panes. You can always add the pane again if you need it later.

The panes available are:

- The **CWS and Syspars** pane.
- The **Nodes** pane.
- The **IBM VSDs** pane.
- The **IBM HSDs** pane.

The CWS and Syspars Pane

You can set the current system partition, which is indicated by a lightening bolt, in the CWS and Syspars pane. The other panes display the set of nodes, virtual shared disks, and hashed shared disks associated with the current system partition. You can set a different system partition to work with by selecting another Syspar object and then selecting **Actions→Set Current System Partition**.

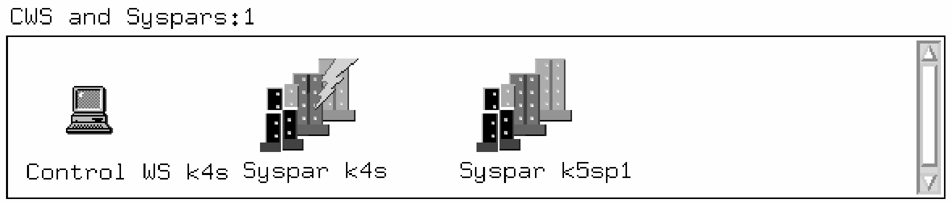


Figure 10. The CWS and Syspars Pane

The Nodes Pane

This pane contains node icons with and without virtual shared disk icons. Those without virtual shared disk icons are nodes that have not been designated as virtual shared disk nodes. You can choose to see the nodes in any of several available views.

As you might have already noticed in Figure 6 on page 8, you can see the nodes in a frame view where the nodes are arranged in their positions with respect to the SP frames that are in the current system partition. If there is an SP-attached server node, it appears outside of the SP frames but still in its position relative to the SP frames.

Another way to see the nodes is in icon view as shown in Figure 11.

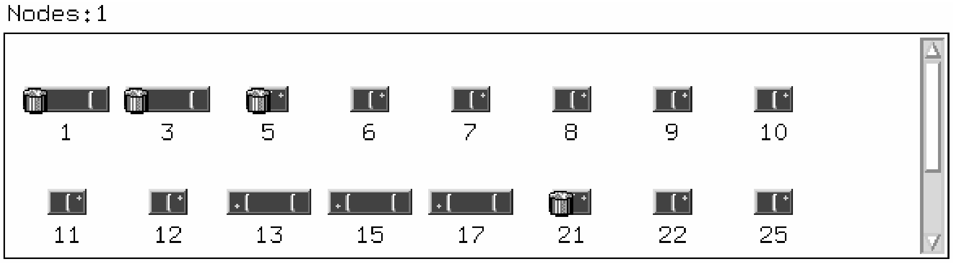
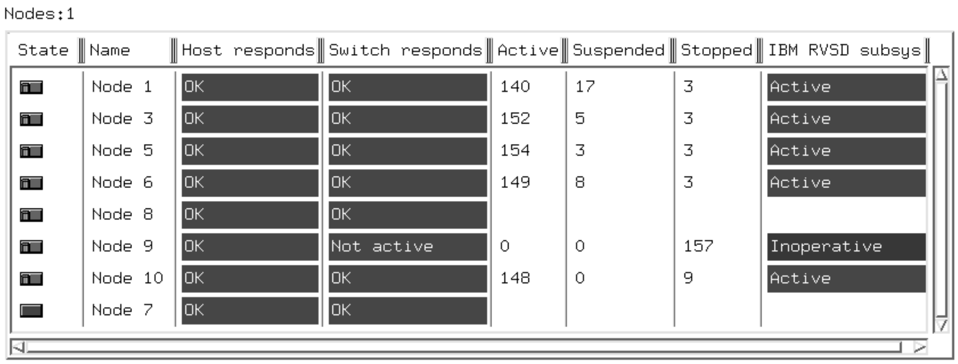


Figure 11. The Nodes Pane in Icon View

Another way to see the nodes is in table view. Table view is typically used as an alternative to looking at attributes in the Properties notebook. With the Nodes pane current, you can click on **View→Show Objects in Table View**. For example, Figure 12 on page 14 shows the attributes:

Host responds
Switch responds
Active IBM VSDs count
Suspended IBM VSDs count
Stopped IBM VSDs count
IBM RVSD subsystem



Nodes:1

State	Name	Host responds	Switch responds	Active	Suspended	Stopped	IBM RVSD subsys
▣	Node 1	OK	OK	140	17	3	Active
▣	Node 3	OK	OK	152	5	3	Active
▣	Node 5	OK	OK	154	3	3	Active
▣	Node 6	OK	OK	149	8	3	Active
▣	Node 8	OK	OK				
▣	Node 9	OK	Not active	0	0	157	Inoperative
▣	Node 10	OK	OK	148	0	9	Active
■	Node 7	OK	OK				

Figure 12. The Nodes Pane in Table View

The table view is visually more meaningful when you see it in color. Green is used for normal operation and red is used to alert you. For example, *Switch responds* is red for Node 9 and indicates *Not active*, and *IBM RVSD subsystem* is red and indicates *Inoperative*. Also notice that for Nodes 7 and 8 there are no counts of Active, Suspended, and Stopped IBM VSDs and IBM RVSD subsystem is also blank. Node 7 has blanks because, as you can tell from the node icon, it is not designated as a virtual shared disk node. Node 8 has blanks either because there are no virtual shared disks configured on the node (it is a virtual shared disk client) or because the node is not running PSSP 3.1 and the attributes are not available.

The first column for the table view is labeled State because you can also set up monitoring while in table view by selecting **View→Set Monitoring...**. The node icons are shown in their normal state when monitoring has not been started, but after monitoring is started the icons change when necessary. For instance, if you had been monitoring for the condition *switchResponds*, there would be a red X on the Node 9 icon while the rest of the node icons would still be green.

The actions available when the Nodes pane is current are:

View and Modify Properties...
Open TTY...
Run Command...
Designate as an IBM VSD Node...
Remove IBM VSD Node Designation
Display IBM VSD Diagnostic Information...
Control IBM RVSD subsystem... (initial reset, start, reset, refresh, stop)
Configure IBM VSDs...
Unconfigure IBM VSDs...
Configure IBM HSDs...
Unconfigure IBM HSDs...
Change IBM VSDs State... (active, suspended, stopped)

When you use **Run Command...**, you must type the command with its full path name unless you set your **DSHPATH** environment variable to include /usr/lpp/ssp/bin/ and /usr/lpp/csd/bin/.

The IBM VSDs Pane

You can create and select virtual shared disks in the IBM VSDs pane.

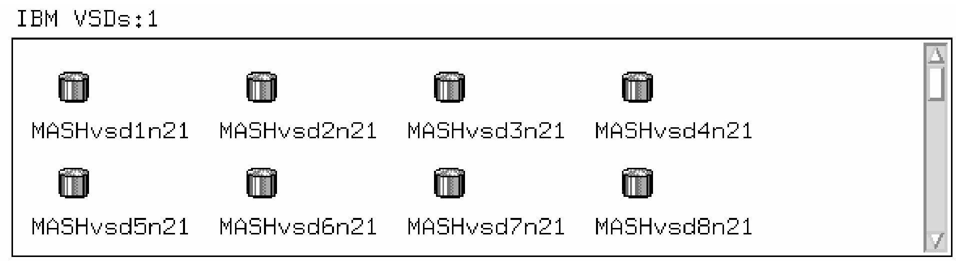


Figure 13. The IBM VSDs Pane

The actions available when this pane is current are:

View and Modify Properties...
Create...
Remove
Define...
Undefine
Configure...
Unconfigure...
Change Owner and Group...
Change State... (active suspended, stopped)

The IBM HSDs Pane

You can create and select hashed shared disks in the IBM HSDs pane.

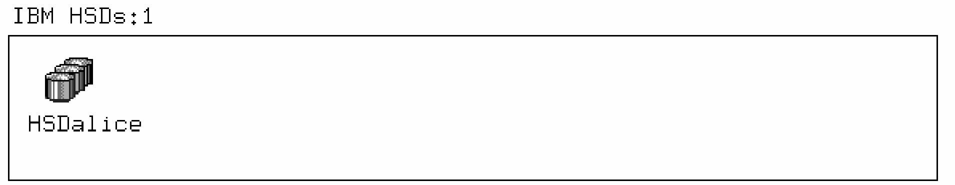


Figure 14. The IBM HSDs Pane

The actions available when this pane is current are:

View Properties...
Create...
Remove
Define...
Undefine
Configure...
Unconfigure...

The Information Area

The information area is at the bottom of the window. The information changes as you move the cursor to different areas. Also, a pop-up or bubble area appears when you place your cursor over a tool bar icon or a pane title.

What's New in the IBM Virtual Shared Disk Perspective?

Many enhancements have been made in support of new as well as most of the existing virtual shared disk functions. Significant enhancements include:

- Support for most of the necessary IBM Virtual Shared Disk, Hashed Shared Disk, and Recoverable Virtual Shared Disk commands resulting in new dialogs for:
  - Designating nodes as virtual shared disk nodes, from the Nodes pane.
  - Removing virtual shared disk designation, from the Nodes pane.
  - Controlling the state of the IBM Recoverable Virtual Shared Disk subsystem, from the Nodes pane.
  - Configuring virtual shared disks, from the Nodes pane and the IBM VSDs pane.
  - Unconfiguring virtual shared disks, from the Nodes pane and the IBM VSDs pane.
  - Configuring hashed shared disks, from the Nodes pane and the IBM HSDs pane.
  - Unconfiguring hashed shared disks, from the Nodes pane and the IBM HSDs pane.
  - Changing the state of the virtual shared disks, from the Nodes pane and the IBM VSDs pane.
  - Defining existing logical volumes as virtual shared disks or existing virtual shared disks as hashed shared disks, from the IBM VSDs or the IBM HSDs pane.
  - Creating virtual shared disks or hashed shared disks, and the underlying logical volumes, from the IBM VSDs or the IBM HSDs pane.
  - Removing virtual shared disks or hashed shared disks, and the underlying logical volumes, from the IBM VSDs or the IBM HSDs pane.
  - Undefining virtual shared disks or hashed shared disks, from the IBM VSDs or the IBM HSDs pane.
- The Run Command action is available in the Nodes pane. When you use this action, you must type the command with its full path name unless you set your **DSHPATH** environment variable to include /usr/lpp/ssp/bin/ and /usr/lpp/csd/bin/.
- New Properties notebook layouts to display attributes and controls for nodes.
- New tool bar icons with bubble help rather than text labels.
- Bubble help in the pane title area for showing statistics and what is currently being monitored.
- Control workstation icon added to the system partitions pane.
- Node icons are shown inside frames with frame numbers and, such as for SP-attached servers, outside frames.
- Nodes with virtual shared disks are visually recognizable by the icon in the Nodes pane.
- Objects in panes can be arranged in a table view, where columns can be based on attributes you select.

- Relationships among nodes, virtual shared disks, and hashed shared disks can be viewed by means of a new Filter to Show Related Objects dialog from the View menu. Three new tool bar icons are provided for the most common relationships.
- New resource variables and conditions have been added to node objects for monitoring the number of active, suspended, and stopped virtual shared disks on the node. There are also resource variables for monitoring the state of the IBM Recoverable Virtual Shared Disk subsystem.
- The objects in the IBM Virtual Shared Disk Perspective graphical user interface are automatically updated every few minutes when related system changes are made either within or outside of the Perspective, so that you can see the updated information.

Chapter 2. Installing the Shared Disk Management Components of PSSP

This chapter guides you through the following:

1. Planning to use the shared disk components, which you need to do whether you are doing a new install or a migration.
2. Migrating shared disk components to PSSP 3.1, which you need to do only if your SP already has virtual shared disks.
3. Installing the shared disk components, which you need to do either to use a shared disk component for the first time on your SP or as part of the migration process.
4. Establishing authorization to perform virtual shared disk activities.

Planning to Use the Shared Disk Components

This section describes things to consider and information that you should know about the IBM Virtual Shared Disk, Hashed Shared Disk, and Recoverable Virtual Shared Disk components before you use them. They are each optional components of PSSP with the following relationships:

- None can function without the base components of PSSP (contained in the install image ssp, file sets **ssp.basic** and **ssp.sysctl** in particular).
- The IBM Virtual Shared Disk component can function without the other two.
- The Hashed Shared Disk component cannot function without the IBM Virtual Shared Disk component.
- The IBM Recoverable Virtual Shared Disk component cannot function without the IBM Virtual Shared Disk component, while it can function with or without the Hashed Shared Disk component.
- The IBM Recoverable Virtual Shared Disk component cannot function without the RS/6000 Cluster Technology (RSCT) Group Services and Topology Services components of PSSP (contained in file sets rsct.basic and rsct.clients).
- To use the IBM Virtual Shared Disk Perspective component, you need the install image ssp.vsdgui.

Planning for IBM Virtual Shared Disk

Before you can use the IBM Virtual Shared Disk component of PSSP, you must:

1. Refer to the book *RS/6000 SP Planning, Volume 2, Control Workstation and Software Environment* for related planning considerations.
2. Check “IBM Virtual Shared Disk Restrictions” on page 2.
3. Determine which applications are to have access to the virtual shared disks you plan to create. See “Tuning Virtual Shared Disk Performance” on page 61 and “Application Programming Considerations for Virtual Shared Disks” on page 71 for considerations in selecting applications to exploit your virtual shared disks and planning your virtual shared disk configuration.
4. Consider the size of the applications you will run.

5. Consider how you want to spread data across the virtual shared disk nodes.
6. Plan your volume groups and logical volumes, using *AIX System Management Guide: Operating Systems and Devices*.
7. Install the IBM Virtual Shared Disk component of PSSP on the control workstation and all nodes that will have or use virtual shared disks (servers and clients).

Planning for Hashed Shared Disk

After planning for IBM Virtual Shared Disk, if you decide to use the Hashed Shared Disk component of PSSP as well, to stripe data across multiple virtual shared disk nodes, you must:

1. Check “Hashed Shared Disk Restrictions” on page 4.
2. Read “Tuning Hashed Shared Disk Performance” on page 68 to understand tuning considerations.
3. Determine where the I/O bottlenecks are on the virtual shared disks.
4. Determine which nodes and disks will be used to eliminate these bottlenecks.
5. Determine the stripe size.
6. Install the Hashed Shared Disk component of PSSP with the IBM Virtual Shared Disk component on all nodes that will have or use virtual shared disks (servers and clients).

Planning for IBM Recoverable Virtual Shared Disk

The IBM Recoverable Virtual Shared Disk component provides recoverability for your virtual shared disks and operates in the same manner whether or not you use the Hashed Shared Disk component. If you want recoverability, in addition to planning for the IBM Virtual Shared Disk component, and the Hashed Shared Disk component if you choose it, you'll need to consider both primary and secondary nodes for your volume groups and plan your hardware configuration. You must:

1. Check “IBM Virtual Shared Disk Restrictions” on page 2 and “IBM Recoverable Virtual Shared Disk Restrictions” on page 7.
2. Plan your twin-tailed volume groups and decide which twin-tailed disks will be in each group. Use any twin-tailed disk supported by the Logical Volume Manager running on AIX 4.3.2.
3. Ensure that the supported disks are properly installed and connected to both nodes if they are twin-tailed.
4. Before PSSP 3.1, IBM Recoverable Virtual Shared Disk was a separate licensed program product. The levels still supported as such are IBM Recoverable Virtual Shared Disk 1.2, 2.1.0, and 2.1.1. In PSSP 3.1, IBM Recoverable Virtual Shared Disk is an optional component of PSSP and is therefore at level 3.1. Each of the currently supported levels of IBM Recoverable Virtual Shared Disk (1.2, 2.1.0, 2.1.1, and 3.1) can interoperate with each of the supported levels of PSSP (2.2, 2.3, 2.4, and 3.1). *Interoperate* means that nodes in the same system partition but at different PSSP levels can access each other's virtual shared disks at the lowest functional level.

   However, to get the newest level of recovery function, the control workstation and each node in the system partition that will have or use virtual shared disks must have AIX 4.3.2 and PSSP 3.1 with the IBM Virtual Shared Disk component and the IBM Recoverable Virtual Shared Disk component installed.

The IBM Recoverable Virtual Shared Disk component uses the RSCT Group Services and Topology Services utilities so those components of PSSP must also be installed.

Migrating Shared Disk Components

Follow these procedures to prepare a node for migrating either the PSSP shared disk component software or the IBM Recoverable Virtual Shared Disk licensed program product to PSSP 3.1. Perform these procedures from the control workstation for each node that has or uses virtual shared disks.

Preparing to Migrate

Perform the following steps to prepare for migration:

1. Issue the **vsdatalst** command with the **-n** option, for example:

   ```
   vsdatalst -n
   ```

   The **vsdatalst -n** command returns parameter values similar to the following for each node:

   ```
          VSD Node Information
                                             Initial Maximum     VSD      rw      Buddy Buffer
     node                 VSD      IP packet   cache   cache request request minimum maximum size: #
   number host_name       adapter    size    buffers buffers   count   count    size    size maxbufs
   ------ --------------- -------- --------- ------- ------- ------- ------- ------- ------- -------
        1 k13n01.ppd.pok. css0         61440      64    4096     256      48    4096  262144      66
   ```

   Make a note of the **VSD adapter** name, which in this case is css0.

2. Shut down any application that is running on the node and uses virtual shared disks.

3. Shut down the virtual shared disk connection manager on the node by issuing:

   `dsh -w` *node_name* `/usr/lpp/csd/bin/hc.vsd stop`

   where *node_name* specifies the target hostname.

   Check the connection manager status by issuing:

   `dsh -w` *node_name* `/usr/lpp/csd/bin/hc.vsd query`

   Continue if all show inoperative.

4. Shut down the IBM Recoverable Virtual Shared Disk daemon on the node by issuing:

   `dsh -w` *node_name* `/usr/lpp/csd/bin/ha.vsd stop`

   where *node_name* specifies the target hostname.

   Check the daemon status by issuing:

   `dsh -w` *node_name* `/usr/lpp/csd/bin/ha.vsd query`

   Continue if all show inoperative.

5. If you are migrating the virtual shared disk software you must unconfigure all existing virtual shared disks and hashed shared disks. Issue the commands:

   ```
   dsh -w node_name /usr/lpp/csd/bin/ucfghsd -a
   dsh -w node_name /usr/lpp/csd/bin/ucfgvsd -a
   ```

6. Disable the connection manager (the hc subsystem) on the node to prevent automatic restart of database instances after shut down:

   ```
   dsh -w node_name mv /usr/lpp/csd/bin/hc.activate\
   /usr/lpp/csd/bin/hc.activate.off

   dsh -w node_name mv /usr/lpp/csd/bin/hc.deactivate\
   /usr/lpp/csd/bin/hc.deactivate.off
   ```

   where *node_name* specifies the target node.

7. Disable the node for the virtual shared disk configuration that was disabled in step 3 on page 21. This step sets the adapter name value to `none` and is to be done for nodes that have virtual shared disks, not for nodes that only use them.

   ```
   /usr/lpp/csd/bin/updatevsdnode -n node_number -a none
   ```

   where *node_number* identifies the target node.

Performing the Migration Install

Install the new PSSP file sets following the procedures in the book *PSSP: Installation and Migration Guide*. Install the file sets for the shared disk components that you want, which are listed in “Installation File Sets” on page 24 of this book, following the additional guidance in “Installing Shared Disk Components” on page 23.

You can have mixed levels of PSSP and any earlier supported level of the IBM Recoverable Virtual Shared Disk licensed program product in the same system partition with PSSP 3.1 but the control workstation must have the 3.1 level of IBM Recoverable Virtual Shared Disk. Use the **rvsdrestrict** command to restrict the level for which the IBM Recoverable Virtual Shared Disk subsystem will run. This command must be run on each node that has PSSP 3.1 installed.

Completing Migration after the Install

Perform these steps to bring your virtual shared disks back online after rebooting the node.

1. If necessary, bring the node back onto the switch by issuing the following command:

   ```
   Eunfence node_number
   ```

   where *node_number* identifies the target node.

2. Enable the node for the virtual shared disk configuration that was disabled by step 3 on page 21 and step 7.

   ```
   /usr/lpp/csd/bin/updatevsdnode -n node_number -a VSD adapter name
   ```

   where *node_number* identifies the target node and *VSD adapter name* is the name you noted in step 1 on page 21 under Preparing to Migrate.

3. Reactivate the connection manager (the hc subsystem):

   ```
   dsh -w node_name mv /usr/lpp/csd/bin/hc.activate.off\
   /usr/lpp/csd/bin/hc.activate

   dsh -w node_name mv /usr/lpp/csd/bin/hc.deactivate.off\
   /usr/lpp/csd/bin/hc.deactivate
   ```

   where *node_name* specifies the target hostname.

4. Use the **rvsdrestrict** command to set the level you want the IBM Recoverable Virtual Shared Disk system to run at. To determine the current setting, issue the following command on the control workstation:

   ```
   /usr/lpp/csd/bin/rvsdrestrict -l
   ```

   To determine what level of the IBM Recoverable Virtual Shared Disk system is installed on each node, issue the following AIX command on the control workstation:

   ```
   dsh -a lslpp -l vsd.rvsd.rvsdd
   ```

   Set the level to the lowest level of IBM Recoverable Virtual Shared Disk that you have running in the system partition. Choose a value from Table 1.

Table 1. Levels for the rvsdrestrict Command

IBM Recoverable Virtual Shared Disk Level	Value for rvsdrestrict Command
3.1	RVSD3.1
2.1.1	RVSD2.1
2.1	RVSD2.1
1.2	RVSD1.2

   For example, if you have some nodes at IBM Recoverable Virtual Shared Disk 2.1.1 and you just installed some nodes with IBM Recoverable Virtual Shared Disk 3.1 but you want them to run in a coexistence environment, you need to restrict the IBM Recoverable Virtual Shared Disk functioning level to RVSD2.1. The command to do this is:

   ```
   /usr/lpp/csd/bin/rvsdrestrict -s RVSD2.1
   ```

   The **rvsdrestrict** command does not dynamically change the IBM Recoverable Virtual Shared Disk subsystem run level across the SP. An instance of the IBM Recoverable Virtual Shared Disk subsystem only reacts to the command after it is restarted. To override the level of a running IBM Recoverable Virtual Shared Disk subsystem, do the following on each node:

   a. Stop the IBM Recoverable Virtual Shared Disk subsystem.

   b. Run the **rvsdrestrict** command.

   c. Restart the IBM Recoverable Virtual Shared Disk subsystem.

   If a node in the same system partition has a lower level of the IBM Recoverable Virtual Shared Disk subsystem than was set by the command, the IBM Recoverable Virtual Shared Disk subsystem will not start on that node.

5. Restart the virtual shared disk on the node by issuing:

   ```
   dsh -w node_name /usr/lpp/csd/bin/ha_vsd reset
   ```

   where *node_name* specifies the target hostname.

Installing Shared Disk Components

This section tells you what you need to have or know before the install, what to install, and how to do the install.

First, you should understand the general installation process, which is:

1. Install AIX 4.3.2 on the control workstation and make it operational. Then be sure you have or install an appropriate level of AIX on all the nodes.

   Note: Remember, you can run earlier supported levels of the IBM Recoverable Virtual Shared Disk software in a coexistence environment but you will not get the latest level of functionality until the control workstation and all the virtual shared disk nodes in a system partition have AIX 4.3.2 and PSSP 3.1 with the prerequisite components.

2. Install PSSP 3.1 on the control workstation, including any optional components that when chosen are required to be on the control workstation, and make it operational.
3. Install the applicable software on the nodes.

Preparing to Install

Before you begin installing the shared disk components, do the following:

- If you haven't already done it, read “Planning to Use the Shared Disk Components” on page 19 and prepare your plan.
- Read the documentation that accompanies the installation media to make sure you have the appropriate version and level of all software. The "Read This First" document lists any required service updates (APARs) for the PSSP or AIX products that must be installed along with the file sets for the IBM Virtual Shared Disk, Hashed Shared Disk, and IBM Recoverable Virtual Shared Disk components.
- Install AIX 4.3.2 following procedures in the AIX publications. It must be installed, configured, and operational on the control workstation before you proceed.
- Decide which shared disk components you want to use and select both the required and optional PSSP file sets to install respectively. Keep in mind any dependencies that the options you choose might have on other optional components. Select file sets from the list in “Installation File Sets” and the list in the book *RS/6000 SP Planning, Volume 2, Control Workstation and Software Environment*.
- Be sure you have enough space. Use the space specifications in “Detailed Space Requirements” on page 25 and in the book *RS/6000 SP Planning, Volume 2, Control Workstation and Software Environment* to help you evaluate this.
- Install the required components of PSSP 3.1 and make them operational, file sets **ssp.basic** and **ssp.sysctl** in particular, following the procedures in the book *PSSP: Installation and Migration Guide*.

Installation File Sets

The file sets involved are:

- For the IBM Virtual Shared Disk component:

vsd.vsdd	virtual shared disk device driver
vsd.sysctl	sysctl commands
vsd.cmi	Centralized Management Interface (SMIT)

- For the Hashed Shared Disk component:

vsd.hsd	data striping device driver

- For the IBM Recoverable Virtual Shared Disk component:

vsd.rvsd.rvsdd	recovery manager
vsd.rvsd.hc	connection manager
vsd.rvsd.scripts	recovery scripts

Note: The IBM Virtual Shared Disk Perspective component is in ssp.vsdgui. The PostScript file for this book and the man pages for related commands are contained in ssp.docs. They are in the **ssp** install image which should be installed on the control workstation.

Detailed Space Requirements

Use the information in this section to help you estimate the space you need in order to install the optional shared disk components of PSSP. See the book *RS/6000 SP Planning, Volume 2, Control Workstation and Software Environment* to help you estimate the space for PSSP and other optional components.

The individual components require space in directories as specified in the following table:

Table 2. Space Used

File Set	root during installation	usr during installation	var during execution
vsd.cmi	100KB	270KB	0
vsd.vsdd	8 MB for a file system created in rootvg volume group	490KB	8 MB for file system only if could not be created in rootvg
vsd.hsd	0	220KB	0
vsd.sysctl	0	490KB	0
vsd.rvsd.scripts	0	250KB	8 MB for file system only if vsdd could not create file system in rootvg
vsd.rvsd.rvsdd	0	320KB	0
vsd.rvsd.hc	0	300KB	0

Performing the Install

Use either of the interfaces described in Table 3 on page 26 for each of the components you have decided to install.

Table 3. Installation Interfaces

If using:	**Do this:**
SMIT	**TYPE** **smit install_latest** • The Install New Software Products at Latest Level window appears. **ENTER** **/spdata/sys1/install/pssplpp/pssp-3.1/lpp** for Input Device **PRESS** **Do** to display the default install parameters. **PRESS** **List** to show options. **SELECT** One or more program options, or select the header file (called **vsd** with **ALL** on the far right side) to do the full installation. **PRESS** **Do** to complete option selection and to begin installation. When the installation is complete, check the SMIT log file for the installation status. If errors occur, see the related version of the *IBM AIX Problem Solving Guide and Reference* manual.

installp	You can use **installp** to install multiple file sets. For example, to install all of the file sets in PSSP for the virtual shared disk components, enter:
	`installp -a -d  /spdata/sys1/install/pssplpp/pssp-3.1`
	`-X vsd`
	Note - Since AIX 4.2, **installp** automatically commits the packaging file set when you specify the **-a** option.
	To list all of the options for **ssp**, enter:
	`installp -l`
	`-d /spdata/sys1/install/pssplpp/pssp-3.1/pssp.installp`

You can install all the virtual shared disk components at once, even if you do not intend to use them all, you can select files using a wildcard character (like vsd.*), or you can install individual file sets. Whatever tac you take, the following steps must be completed:

1. Install the IBM Virtual Shared Disk component **on the control workstation.**

   It must be on the control workstation for configuration management. The control workstation is neither a server nor a client.

   The files to install are:

vsd.vsdd	device driver
vsd.sysctl	sysctl commands
vsd.cmi	Centralized Management Interface (SMIT)

2. Install the IBM Virtual Shared Disk component **on the virtual shared disk client and server nodes**

   The file sets required on the client and server nodes are **vsd.vsdd** and **vsd.sysctl**.

3. If you decided to use the Hashed Shared Disk component, install **vsd.hsd on the client and server nodes**.

4. If you decided to use the IBM Recoverable Virtual Shared Disk component, install it **on the control workstation and the client and server nodes**.

   The files to install are:

vsd.rvsd.rvsdd	recovery manager
vsd.rvsd.hc	connection manager
vsd.rvsd.scripts	recovery scripts

Establishing Authorization

You might have to make the authentication system operational and set user authorizations and verify that authentication and authorizations are operational before you can use the shared disk components that you just installed.

Authentication System

PSSP, the sysctl subsystem in particular, must be configured and operational. Sysctl is an authentication system for running commands remotely and in parallel. It provides:

- Least privilege capability

  Root authority can be dynamically provided to non-root users based on their authenticated identity, the task they are trying to perform, access control lists, and any other relevant criteria. The root password need not be given out to as many people so it is kept secure.

- Distributed execution

  Sysctl applications can be executed on remote hosts with full authentication and authorization.

- Parallel execution

  Sysctl applications can be efficiently executed in parallel on many hosts.

In order to run commands for managing shared disks in particular, the /etc/sysctl.conf file must have the following statement in it:

```
include /usr/lpp/csd/sysctl_vsd.cmds
```

See *PSSP: Installation and Migration Guide* for installation information and *PSSP: Administration Guide* for information about sysctl.

User Authorization

User authorization must be established. Before you can use IBM Virtual Shared Disk Perspective actions or virtual shared disk commands that operate on multiple nodes (such as **createvsd**, **vsdsklst**, and **vsddiag**) or to operate from a remote host, you must have Kerberos and sysctl authorization. If you do not already have authorization, your system administrator must run **/usr/kerberos/bin/add_principal** to put your ID into the authorized list. The administrator must also add your principal ID to the **/etc/sysctl.acl** and **/etc/sysctl.vsd.acl** files. Then the administrator must run **sysctl svcrestart** to complete the database update for the new entries. The administrator will have to perform these authorization steps on the control workstation and on all nodes that will be virtual shared disk clients or servers.

When you next logon, you might need to run **kinit** on the control workstation and each node that is to have or use virtual shared disks to get the correct Kerberos ticket. You can verify your ticket with the **klist** command and run **sysctl whoami** to check your sysctl authorization.

To check whether you can run the virtual shared disk and hashed shared disk commands that operate on multiple nodes, use **sysctl sysctl_vsdcheck**. Issue **vsdsklst** as a check before you use any other commands that operate on multiple nodes. If your authorization is correct, it will return information about the virtual shared disks that are defined on the system partition.

To check for remote authority, you could log on to the control workstation and issue commands to run on nodes. The session would consist of:

```
kinit userid.instance
dsh -w target_node /usr/lpp/csd/bin/lsvsd -l
```

For more complete information see the book *PSSP: Administration Guide*.

Chapter 3. Understanding Your Managing Shared Disks Process

This chapter explains procedural considerations and points out, at a high level, the tasks to perform after you have completed installing the software and made it operational. The tasks are listed and procedural criteria is explained. The tasks appear in other chapters that expand them into lower level steps.

Which procedural choices apply to you depends on whether you are establishing a new virtual shared disk environment, or you are adding to or changing an existing virtual shared disk environment. Which steps you must explicitly perform versus which steps are done for you at once, depends on whether you already have logical volumes and global volume groups, which shared disk components you choose to use, and which interface you choose to use.

The global tasks in the process toward fully operational virtual shared disks are generally the same in all cases. However, you need to consider where you are, where you want to be with respect to the virtual shared disks on your system, and how you want to get there. Within a global task, different base actions or commands might be necessary to complete the steps.

For instance, if yours is a new system with all physical and software components just installed and configured, you can take full advantage of the IBM Virtual Shared Disk graphical user interface actions that work on multiple nodes and perform many steps at one time. On the other hand, you might already have used the Logical Volume Manager of AIX to establish volume groups and logical volumes or you might already have virtual shared disks. You might even have scripts that run the more basic single-node commands. In each case, different steps are necessary depending on what is already done and what has yet to be done.

Generally, the tasks are:

1. **Designate node as a virtual shared disk node**

   Do this for every node that is to have or use a virtual shared disk, regardless of whether or not the virtual shared disks will have data striping or will be recoverable. With the graphical user interface, you can select all the applicable nodes and apply the action **Designate as an IBM VSD Node...** at one time. If you prefer, you can use the command **vsdnode**.

2. **Create or define the virtual shared disks or hashed shared disks**

   Do this for each node that is to be a virtual shared disk server.

   - If you do not already have logical volumes and volume groups established, do one of the following:

     – If you do not want data striping, use the action **Create...** from the IBM VSDs pane or the command **createvsd**, which also create the underlying logical volumes.

     – Otherwise, use the action **Create...** from the IBM HSDs pane or the command **createhsd**. The underlying virtual shared disks and logical volumes are created as well.

   - If you do have logical volumes and volume groups established, do the first or both of the following:

   a. Use the action **Define...** from the IBM VSDs pane or the command **defvsd**.

   b. If you want data striping, use the action **Define...** from the IBM HSDs pane or the command **defhsd**.

3. **Configure the virtual shared disks or hashed shared disks**

   - If you want recoverability you should have installed the IBM Recoverable Virtual Shared Disk component on each virtual shared disk node. In that case, you can use the action from the Nodes pane **Control IBM RVSD subsystem...**. This will automatically configure and activate all the virtual shared disks or hashed shared disks as soon as quorum is met and activate recoverability on the virtual shared disk nodes that you select when you select the **Initial Reset** state. If you prefer to use the command **ha_vsd reset**, you must run it on each virtual shared disk node.

   - If you do not want recoverable virtual shared disks, configure the virtual shared disk or hashed shared disk on each node either using an action from the Nodes pane (**Configure IBM VSDs...** or **Configure IBM HSDs...**) or using a command (**cfgvsd**, **cfghsd**, or **cfghsdvsd**) which might have to run for each virtual shared disk or hashed shared disk.

4. **Activate the virtual shared disks**

   Skip this step if you established recoverable virtual shared disks because it is already done.

   Otherwise, use the action from the Nodes pane (**Change IBM VSD state...**), which can do this at once for all the nodes you select, or use the commands **preparevsd** and **startvsd** which must be run on each virtual shared disk node for each virtual shared disk.

Your applications can then begin to write to and read from virtual shared disks or hashed shared disks. To understand what you must do for applications to efficiently use them, see:

- Chapter 8, “Performance and Tuning Considerations for Virtual Shared Disks and Hashed Shared Disks” on page 61
- Chapter 9, “Application Programming Considerations” on page 71
- Chapter 12, “IOCTL Subroutine” on page 89

If you do use the IBM Recoverable Virtual Shared Disk subsystem, also see:

- Chapter 10, “What You Need to Know about How the IBM Recoverable Virtual Shared Disk Component Works” on page 77
- Chapter 11, “Recovery Scenarios” on page 83

It should now be clear that you need to understand your starting point, your goal, and plan what to do to reach your goal. There are many possible scenarios and many actions and commands. Actions are explained in detail in the online help. Interfaces are summarized in Appendix A, “Interface Cross-Reference” on page 93. Usage of some commands is explained in Appendix B, “Single-Node Command and SMIT Interfaces” on page 101.

Chapter 4. Creating and Activating Virtual Shared Disks

Before you can do anything described in this or other chapters, you need to be authorized. See "Establishing Authorization" on page 27.

This chapter tells you how to do the following tasks to establish your virtual shared disk environment:

- Designate nodes as virtual shared disk nodes.
- Define global volume groups.

  The *create* actions and commands do this step for you. You only need to explicitly do this if you intend to use the *define* actions or commands instead.
- Create or define virtual shared disks and hashed shared disks.
- Configure the virtual shared disks and hashed shared disks.
- Activate the virtual shared disks.

If you have just completed software installation or migration, you might be interested in the following subjects in this chapter:

- How to verify that you can write to a virtual shared disk.
- What to do after adding the IBM Recoverable Virtual Shared Disk component to a system where the IBM Virtual Shared Disk component has already been in use.

Designating Nodes as IBM VSD Nodes

When you do this, you are actually entering information about your nodes into the SDR. To enter, display, or change the setup information that is required for each node that is to either have or use virtual shared disks:

- Start the IBM Virtual Shared Disk Perspective by issuing **spvsd** or by double-clicking on the IBM VSD Perspective icon on the SP Perspectives Launch Pad
- Click on one or more icons in the Nodes pane to select them
- Click on **Actions→Designate as an IBM VSD Node...**

That opens a dialog window where you can enter pertinent information. The following information must be entered before you can create virtual shared disks or hashed shared disks:

- Node number

  Each node is identified by a *node_number* that represents its position in the SP rack.
- Adapter name

  The name of the adapter to be used for virtual shared disk communications with this node. All nodes must use the same adapter.

  The virtual shared disk component uses this name and the related node number to get the IP address associated with each node from the SDR

Adapter object. IBM recommends you use the SP Switch (defined as **css0**) as the virtual shared disk adapter for best performance.

The set of nodes and addresses in the virtual shared disk configuration must define a fully-connected network, that is, all nodes can send messages directly to each other. For best performance, however, all the nodes should be directly connected to each other over the switch network.

- Initial cache buffer count

  The virtual shared disk device driver implements an optional write-through cache of pinned kernel memory with a block size of 4KB. When the first cached virtual shared disk is configured on a node, the cache is created, containing the specified number of blocks. The minimum value is 1. The recommended value is 256, which results in a 1MB (1024KB) cache.

- Maximum cache buffer count

  The number of buffers in the cache can be increased (using **ctlvsd**) up to this value. You cannot decrease the number of cache blocks. You can start over by unconfiguring all the virtual shared disks, changing the value of this number in the SDR, and then, when you configure the virtual shared disks, the initial number of virtual shared disk buffer cache blocks will be created. The recommended value is 256, resulting in a 1MB cache.

- Request block count

  Specifies the maximum number of outstanding virtual shared disk requests originating on the node. The virtual shared disk device driver allocates this number of request blocks in pinned kernel memory the first time a virtual shared disk is configured. If the number is too small, local requests will queue up waiting for a request block to become available. The recommended value is 256. The size of the block is approximately 76 bytes.

- Number of outstanding logical volume read and write requests

  Specifies the maximum number of outstanding requests the virtual shared disk device driver will allow for each underlying logical volume. Each such request uses a pbuf. A pbuf is a little bigger than a buf structure, which is approximately 104 bytes. The virtual shared disk device driver allocates this number of pbufs in pinned kernel memory whenever a virtual shared disk that has this node as its primary is configured. The minimum value is 1. The recommended value for a primary or secondary node is 48. If the node is not going to be used as a primary or secondary, this value is irrelevant.

- Minimum buddy buffer size

  The smallest buddy buffer a server will use to satisfy a request to a virtual shared disk. This value must be a power of 2 and greater than or equal to 4096. The recommended value is 4096 (4KB). For a 512-byte request, 4KB is clearly overkill but a buddy buffer is used only for the short period of time a remote request is being processed at the server node.

  For more information on buddy buffers, see “Buddy Buffers” on page 65.

- Maximum buddy buffer size

  The largest buddy buffer a server will use to satisfy a request. This value must be a power of 2 and greater than or equal to the *min_buddy_buffer_size*. The maximum value, and the recommended value, is 262144 (256KB). This value must be the same on all nodes within a system partition.

- Number of buddy buffers

  The buddy buffer is pinned kernel memory allocated when the virtual shared disk device driver is loaded. Loading occurs the first time a virtual shared disk is configured. Buddy buffers are freed when the last virtual shared disk is unconfigured. If you do not use the switch as your IP adapter, the size of the buddy buffer affects the number of remote requests the virtual shared disk server node can handle at one time. Remote requests can queue while waiting for a buddy buffer. The **statvsd** command and the Statistics notebook page of the IBM Virtual Shared Disk Perspective report this queuing as buddy buffer shortages. See “Buddy Buffers” on page 65 for more information about buddy buffers and how to determine how many to define. The recommended value when the switch is used as the IP adapter is 2, which results in a 512KB buddy buffer size in combination with the recommended maximum buddy buffer size of 256KB. Tuning may be required on the switch; see “SP Switch Considerations” on page 63.

- Maximum IP message size

  The maximum IP message size for virtual shared disks, in bytes. The default value is 24,576 (24KB). If you are using the switch as your virtual shared disk adapter, use a value of 61,440 (60KB).

You can display and change some virtual shared disk node attributes which are kept in the SDR from the Node Attributes page of the Properties notebook at any time. You must reconfigure the virtual shared disk on the node before the changes take effect, however.

For more information about these options, see the full description of the action **Designate as an IBM VSD Node...** in the online help or see the **vsdnode** command in the book *PSSP: Command and Technical Reference*.

Defining Global Volume Groups

Volume groups used for virtual shared disks must be given a global name that is unique across system partitions.

This task is always done but you do not always have to perform it. The **Create...** actions and the comparable **createvsd** and **createhsd** commands do this for you. You only have to do this explicitly if you need to use the **Define...** action or **defvsd** command to create your virtual shared disks because you already have logical volumes.

You can use the **Run Command...** action and run the **vsdvg** command to define global volume groups.

Creating or Defining Virtual Shared Disks or Hashed Shared Disks

Remember, your procedure is based on whether or not you already have logical volumes. The *create* actions and commands take care of logical volumes and global volume groups for you. If you already have them, you must do the *define* steps instead.

If you are using the *create* actions or commands, it's a good idea to check for old rollback files.

Checking for Old Rollback Files

If you issue a virtual shared disk action or command that operates on multiple nodes, such as **createvsd** or **createhsd**, and the command fails, a rollback file will be created so that a second invocation of that command can start at the last successful operation. (A command that is issued against multiple nodes fails if any of the nodes cannot execute it.) If you later change your virtual shared disk configuration or run a different command, **createvsd** attempts to complete processing using the rollback file and will fail. Be sure there are no rollback files from failed invocations of commands on your system. If there are any, they can be found in

/usr/lpp/csd/sysctl/new_rollback and **/usr/lpp/csd/vsdfiles/vsd_rollback**. You can delete them with the **rm** command. Old rollback files can interfere with the processing of new invocations of these commands.

Any time a virtual shared disk or hashed shared disk command that operates on multiple nodes fails, check for old rollback files.

Creating Virtual Shared Disks

You can create several virtual shared disks with a single graphical user interface action or a line command (on both primary and secondary nodes if you have the IBM Recoverable Virtual Shared Disk component running). You must first have used the IBM Virtual Shared Disk Perspective or the **vsdnode** command to set up information in the SDR about each node involved in this virtual shared disk configuration.

Do not perform this task if you have already used the Logical Volume Manager of AIX to establish logical volumes. The create process generates them for you. Instead, see “Defining Virtual Shared Disks and Hashed Shared Disks” on page 39. Also, if you want data striping see “Creating Hashed Shared Disks” on page 36 instead.

Creating Virtual Shared Disks with the IBM Virtual Shared Disk Perspective

To create virtual shared disks using the IBM Virtual Shared Disk Perspective graphical user interface, do the following:

- Click on the IBM VSDs pane
- Click on **Actions→Create...**

That opens the Create IBM VSDs dialog where you can :

- Enter all the pertinent information
  - Number of IBM VSDs per node
  - IBM VSD name prefix
  - Logical volume name prefix
  - Volume group Name
  - IBM VSD size (MB)
  - Mirroring count
  - Physical partition size (MB)
- Select a cache option (cache or no cache)

- Select from a list of nodes that have been designated as virtual shared disk nodes, the primary node and the backup node
- Select from a list, the physical disks that the virtual shared disk is to span
- Use the arrow button to move the node and disk combinations to the *Create on Nodes* list

The equivalent command is **createvsd**. See the book *PSSP: Command and Technical Reference* for syntax.

Information about virtual shared disk definitions is stored in the SDR **VSD_Global_Volume_Group** object and the **VSD_Table** object. You can view the information using the IBM Virtual Shared Disk Perspective graphical user interface.

Examples of Creating Virtual Shared Disks with createvsd

The following examples range from simple to fairly complex.

- To create identical virtual shared disk definitions on each of three nodes in a system partition, type:

  ```
  createvsd -n 1,3,5 -s 4 -g SYS2VG -v SYS2VSD
  ```

  This creates the following virtual shared disk definitions:

  - SYS2VSD1n1 on node 1. The local volume group name on node 1 is SYS2VG. The global volume group name is SYS2VGn1. The logical volume is lvSYS2VSD1n1.
  - SYS2VSD2n3 on node 3. The local volume group name on node 3 is SYS2VG. The global volume group name is SYS2VGn3. The logical volume is lvSYS2VSD2n3.
  - SYS2VSD3n5 on node 5. The local volume group name on node 5 is SYS2VG. The global volume group name is SYS2VGn5. The logical volume is lvSYS2VSD3n5.

  No secondary nodes are defined. The space allocated to a virtual shared disk is spread across all the physical disks (hdisks) within its local volume group on each node (1,3, and 5).

- To assign each disk in the previous example a secondary node (with the IBM Recoverable Virtual Shared Disk component running), type:

  ```
  createvsd -n 1/2/,3/4/,5/6/ -s 4 -g SYS2VG -v SYS2VSD
  ```

  This creates the following virtual shared disk definitions:

  - SYS2VSD1n1 on node 1 with a twin-tailed connection to node 2. The local volume group name on node 1 is SYS2VG. The global volume group name is SYS2VGn1. The logical volume is lvSYS2VSD1n1.
  - SYS2VSD2n3 on node 3 with a twin-tailed connection to node 4. The local volume group name on node 3 is SYS2VG. The global volume group name is SYS2VGn3. The logical volume is lvSYS2VSD2n3.
  - SYS2VSD3n5 on node 5 with a twin-tailed connection to node 6. The local volume group name on node 5 is SYS2VG. The global volume group name is SYS2VGn5. The logical volume is lvSYS2VSD3n5.

  The volume groups in this example are imported to the secondary node.

- To create three virtual shared disk definitions on primary and secondary nodes (with the IBM Recoverable Virtual Shared Disk running), where the logical volume created on nodes 3 and 4 spans two disks and the volume group spans three disks, type:

  ```
  createvsd -n 3/4:hdisk1,hdisk2+hdisk3/,5/6/,7/8/ -s 12 -g datavg\
  -v USER -x
  ```

  This command creates the following virtual shared disk definitions:

  - USER1n3, with logical volume lvUSER1n3 defined on a volume group with the global volume group name datavgn3 on node 3, imported to node 4. The volume group datavgn3 spans hdisk1, hdisk2, and hdisk3. The logical volume lvUSER2n3 spans hdisk1 and hdisk2.
  - USER2n5, with logical volume lvUSER2n5 defined on a volume group with the global volume group name datavgn5 on node 5, imported to node 6.
  - USER3n7, with logical volume lvUSER3n7 defined on a volume group with the global volume group name datavgn7 on node 7, also imported to node 8.

    The volume groups datavgn5 and datavgn7 are created with one 4MB partition from a single physical disk. The volume group datavgn3 is created with one 4MB partition from the three physical disks hdisk1, hdisk2, and hdisk3.

  The local volume group name on each node is datavg. The volume groups are **not** imported to the secondary node because the -x flag was used in the command.

Creating Hashed Shared Disks

You can create an entire configuration of hashed shared disks and underlying virtual shared disks with a single graphical user interface action or line command (on both primary and secondary nodes if you have the IBM Recoverable Virtual Shared Disk component running).

Do not perform this task if you have already used the Logical Volume Manager of AIX to establish logical volumes. The create process generates them for you. Instead, see “Defining Virtual Shared Disks and Hashed Shared Disks” on page 39.

Creating Hashed Shared Disks with IBM Virtual Shared Disk Perspective

To create hashed shared disks using the IBM Virtual Shared Disk Perspective graphical user interface, do the following:

- Click on the IBM HSDs pane
- Click on **Actions→Create...**

That opens the Create IBM HSDs dialog where you can :

- Enter all the pertinent information
  - Number of IBM VSDs per node
  - IBM HSD name prefix
  - Logical volume name prefix
  - Volume group Name
  - IBM HSD size (MB)
  - Mirroring count

  - Physical partition size (MB)
  - Stripe size (KB)
- Select a cache option (cache or no cache)
- Select a protect LVCB option (protect, do not protect)
- Select from a list of nodes that have been designated as virtual shared disk nodes, the primary node and the backup node
- Select from a list, the physical disks that the virtual shared disk is to span
- Use the arrow button to move the node and disk combinations to the *Create on Nodes* list

The equivalent command is **createhsd**.

Examples of Creating Hashed Shared Disks with createhsd

For the syntax of the **createhsd** command see the book *PSSP: Command and Technical Reference*. The following are examples wich range from simple to fairly complex.

- To create a hashed shared disk that stripes data across three identical virtual shared disk definitions on each of three disks in a system partition, type:

  ```
  createhsd -n 1,3,5 -s 12 -g SYS2VG -t 4 -d SYS2HSD
  ```

  This creates the hashed shared disk definition SYS2HSD and its underlying virtual shared disk definitions:

  - SYS2HSD1n1 on node 1. The local volume group name on node 1 is SYS2VG. The global volume group name is SYS2VGn1. The logical volume is lvSYS2HSD1n1.
  - SYS2HSD2n3 on node 3. The local volume group name on node 3 is SYS2VG. The global volume group name is SYS2VGn3. The logical volume is lvSYS2HSD2n3.
  - SYS2HSD3n5 on node 5. The local volume group name on node 5 is SYS2VG. The global volume group name is SYS2VGn5. The logical volume is lvSYS2HSD3n5.

    The usable hashed shared disk size is 12MB. The stripe size is 4KB. The first stripe on each disk is skipped to allow space for the LVCB.

  No secondary node is defined. The space allocated to the hashed shared disk is spread across all the physical disks (hdisks) connected to each node (1, 3, and 5).

- With the IBM Recoverable Virtual Shared Disk component running, to create a hashed shared diskdefinition whose underlying virtual shared disks are twin-tailed and have secondary direct client nodes assigned, type:

  ```
  createhsd -n 1/2/,3/4/,5/6/ -s 48 -g SYS2VG -t 8 -d SYS2HSD
  ```

  This adds the backup node number to the naming convention and creates hashed shared disk SYS2HSD with the following virtual shared disk definitions:

  - SYS2HSD1n1 on node 1 with a twin-tailed connection to node 2. The local volume group name on node 1 is SYS2VG. The global volume group name is SYS2VGn1. The logical volume is lvSYS2HSD1n1.
  - SYS2HSD2n3 on node 3 with a twin-tailed connection to a second direct client on node 4. The local volume group name on node 3 is SYS2VG. The

global volume group name is SYS2VGn3. The logical volume is lvSYS2HSD2n3.

- SYS2HSD3n5 on node 5 with a twin-tailed connection to a second direct client on node 6. The local volume group name on node 5 is SYS2VG. The global volume group name is SYS2VGn5. The logical volume is lvSYS2HSD3n5.

  The usable hashed shared disk size is 48MB. The stripe size is 8KB. The first stripe on each disk is skipped to allow space for the LVCB.

- With the IBM Recoverable Virtual Shared Disk subsystem running, to create a hashed shared disk with three underlying virtual shared disk definitions on primary and secondary nodes, where the logical volumes and volume groups span two physical disks, type:

  ```
  createhsd -n 3/4:hdisk1,hdisk2/,5/6:hdisk1,hdisk2/,7/8:hdisk1,hdisk2/
            -s 48 -t 12 -S -g datavg -d USER
  ```

  This command creates the hashed shared disk USER, with the following underlying virtual shared disk definitions:

  - USER1n3, with logical volume lvUSER1n3 defined on a volume group with the global volume group name datavgn3 on node 3, imported to node 4. The volume group datavgn3 and the logical volume lvUSER2n3 span hdisk1 and hdisk2.
  - USER2n5, with logical volume lvUSER2n5 defined on a volume group with the global volume group name datavgn5 on node 5, imported to node 6. The volume group datavgn5 and the logical volume lvUSER2n5 span hdisk1 and hdisk2.
  - USER3n7, with logical volume lvUSER3n7 defined on a volume group with the global volume group name datavgn7 on node 7, imported to node 8. The volume group datavgn7 and the logical volume lvUSER2n7 span hdisk1 and hdisk2.

    The usable size of the hashed shared disk is 48MB. The stripe size is 12KB. The first stripe on each disk is **not** skipped to allow space for the LVCB.

Creating Hashed Shared Disks in a Node-Pair Configuration

If your system partition is configured in a node-pair arrangement like that shown in Figure 15 on page 39, where sets of physical disks are twin-tailed on nodes 1 and 2, nodes 3 and 4, nodes 5 and 6, and so forth, you will need to use the following process to create a single hashed shared disk for the system partition. In this configuration, odd-numbered nodes are the backups for virtual shared disks defined on even-numbered nodes and even-numbered nodes are the backups for virtual shared disks defined on odd-numbered nodes.

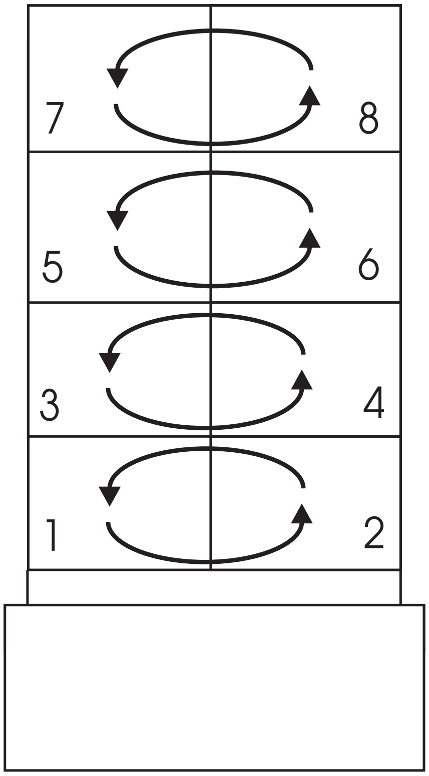


Figure 15. A Node-Pair Configuration

To set up the hashed shared disk for this configuration, use **createvsd** to define one or more virtual shared disks on the odd-numbered nodes, and then to define one or more virtual shared disks on the even-numbered nodes. Virtual shared disks on odd-numbered nodes must be defined separately from those on even-numbered nodes in this configuration if they are to be part of a single Hashed Shared Disk. You then use **defhsd** to define the Hashed Shared Disk.

For example, to define a Hashed Shared Disk for the configuration in Figure 15, you could use the following sequence of commands:

```
createvsd 1/2/,3/4/,5/6/,7/8/ -s 8192 -g s21ovg -v ovsd

createvsd 2/1/,4/3/,6/5/,8/7/ -s 8192 -g s21evg -v evsd

defhsd protect_lvcb s21hsd 8192 ovsdn1 ovsdn3 ovsdn5 ovsdn7
evsdn2 evsdn4 evsdn6 evsdn8
```

Defining Virtual Shared Disks and Hashed Shared Disks

When you already have global volume groups and logical volumes you cannot use the create actions or commands, you must perform a define process instead. First you must define all the virtual shared disks then, if you want data striping, define the hashed shared disks.

To define all the virtual shared disks:

- Start the IBM Virtual Shared Disk Perspective by issuing **spvsd** or by double-clicking on the IBM VSD Perspective icon on the SP Perspectives Launch Pad
- Click on the IBM VSDs pane

- Click on **Actions→Define IBM VSDs...**

That opens a dialog window where you can enter the pertinent information:

- Logical volume name
- Global volume group name
- IBM VSD name
- Cache option (cache or nocache)

Alternatively, you can use the **defvsd** command.

To define all the hashed shared disks:

- Start the IBM Virtual Shared Disk Perspective by issuing **spvsd** or by double-clicking on the IBM VSD Perspective icon on the SP Perspectives Launch Pad
- Click on the IBM HSDs pane
- Click on **Actions→Define IBM HSDs...**

That opens a dialog window where you can enter the pertinent information:

- IBM VSD name
- Protect LVCB option (protect or do not protect)
- Stripe size (KB)
- Underlying IBM VSD names

Alternatively, you can use the **defhsd** command.

Configuring Virtual Shared Disks or Hashed Shared Disks

After you've created your virtual shared disks or hashed shared disks, you must configure them on all the nodes that need to read from and write to them.

If you want recoverability you should have installed the IBM Recoverable Virtual Shared Disk software on each virtual shared disk node. In that case, you can use the action from the Nodes pane **Control IBM RVSD subsystem...**, which will automatically configure and activate all the virtual shared disks as soon as quorum is met and activate recoverability on all the virtual shared disk nodes after you set the state to **Initial Reset**. If you prefer to use the command **ha_vsd reset**, you must run it on each virtual shared disk node.

To configure all the virtual shared disks or hashed shared disks:

- Start the IBM Virtual Shared Disk Perspective by issuing **spvsd** or by double-clicking on the virtual shared disk icon on the SP Perspectives Launch Pad
- Select the applicable icons in the Nodes pane representative of all the virtual shared disk server and client nodes
- Click on **Actions →Configure IBM HSDs...** or **Configure IBM VSDs...**

Alternatively, from the IBM HSDs pane or from the IBM VSDs pane you can select the action **Configure...**

You can also use the **cfgvsd** or **cfghsdvsd** commands or SMIT.

Activating Virtual Shared Disks

If you have the IBM Recoverable Virtual Shared Disk subsystem running on each virtual shared disk node, the configuration task accomplishes this step for you as soon as quorum is met.

If you do not have recoverable virtual shared disks, use the IBM Virtual Shared Disk Perspective to start your virtual shared disks and hashed shared disks. See "Changing the States of Virtual Shared Disks" on page 47.

Verifying You Can Write to a Virtual Shared Disk

CAUTION:
The verification procedure writes data to the virtual shared disks. Do not perform this task after you have put real data on a virtual shared disk. This is only appropriate after a software install and creation of a new virtual shared disk.

To test that you have successfully installed the IBM Virtual Shared Disk software and that you can successfully define, activate, read from, and write to a virtual shared disk after creating your virtual shared disk using the **createvsd** command, from each virtual shared disk client node, run the **vsdvts** command. You can perform this task using the **Run Command...** dialog of the IBM Virtual Shared Disk Perspective interface.

Adding IBM Recoverable Virtual Shared Disk After Virtual Shared Disks are Already in Use

This section describes how to prepare to use the IBM Recoverable Virtual Shared Disk component on a system where the IBM Virtual Shared Disk component has already been in use, whether or not you are adding new disk hardware.

Follow these general steps:

1. If you have been using the IBM Virtual Shared Disk component without the IBM Recoverable Virtual Shared Disk component, disable any scripts that issue commands that change the states of virtual shared disks. The IBM Recoverable Virtual Shared Disk subsystem changes the states of virtual shared disks automatically. Your scripts could disrupt recovery processes.
2. To start the IBM Recoverable Virtual Shared Disk subsystem and your virtual shared disks, do one of the following:

   - Reboot all the virtual shared disk nodes.

   or

   - Use the IBM Virtual Shared Disk Perspective action **Control IBM RVSD subsystem** and select **Initial Reset**, or issue the **ha_vsd reset** command on all virtual shared disk nodes.

Chapter 5. Displaying and Modifying Virtual Shared Disk Information

This chapter tells you how to do the following:

- Display and modify information about virtual shared disk nodes
- Display information about global volume groups
- Display and modify information about virtual shared disks
- Display and reset virtual shared disk statistics
- Display and reset virtual shared disk device driver statistics
- Display disk resource information
- Display information about hashed shared disks

You can use the Properties notebook pages in the IBM Virtual Shared Disk Perspective to display a variety of information from the SDR related to your virtual shared disks. You can view Properties notebook pages by clicking on the tool bar icon or on **Actions→View and Modify Properties...**.

Another way to view information about more than one node or virtual shared disk at once is with table view. You can click on the tool bar icon or on **View→Show objects in table view** and select the attributes that you want to see.

Displaying and Modifying Information about Virtual Shared Disk Nodes

To display information about virtual shared disk nodes, whether server nodes or clients, do the following:

1. Click on the node in the Nodes pane
2. Click on the Properties notebook icon in the tool bar or click on **Actions→View and Modify Properties...**

The IBM VSD Node notebook shows the options set in the SDR for this node. The notebook contains pages with the following categories of information:

- Node Status

  This page shows:

  - Power LED
  - Host responds
  - Controller responds
  - Switch responds
  - CPU's online
  - Key switch position (normal, secure, service)
  - Environment LED
  - TTY open
  - Node failure
  - 3 Digit Display

  It also has the following action buttons:

  - Power On... or Power Off...

  - Fence... or Unfence...
  - Open TTY...
  - Network Boot...

- Configuration

  This page has information about the node, its frame, host names, IP address, system partition, PSSP level, processor, slot, and switch.

- IBM VSD Status

  This has counts of active, suspended and stopped virtual shared disks. It also has the IBM RVSD subsystem status. The values in this page will be unknown (question marks) until at least one virtual shared disk is configured on the node.

- IBM VSD Node Attributes

  This page shows and lets you change attributes related to the virtual shared disks on the node.

- Configured IBM VSDs and HSDs

  This page has two panes to show what is configured on the node identified in the notebook title bar. One shows virtual shared disk names, states, the primary server node, and other information. The other shows the hashed shared disk names.

- IBM VSD Node Statistics

  This page shows the statistics returned by the commands **statvsd** and **vsdsklst**.

- IBM HSD Node Statistics

  This page shows the statistics returned by the commands **ctlhsd** and **hsdatalst**.

- All Dynamic Resource Variables

  This page lists the variable names and values.

- Monitored Conditions

  If you set up conditions to be monitored, this page shows what is currently being monitored and the state of each condition.

See the online help for detailed descriptions.

You can use the **vsdatalst**, **statvsd**, **lsvsd**, or **ctlvsd** commands or SMIT to get some of the same information.

Displaying Information about Global Volume Groups

You can use the **Run Command...** action and run the **vsdatalst -g** command to display global volume group information.

Displaying and Modifying Information about Virtual Shared Disks

To view attributes for virtual shared disks, select the icon you want to view from the IBM VSDs pane and click on the Properties notebook icon in the tool bar. The notebook has pages for the following information:

- IBM VSD Attributes

  This page shows the name, primary and secondary servers, minor number, physical disks spanned, logical volume group, global volume group, logical volume, and cache option setting.

- IBM VSD Client Nodes

  This page lists the node number and state of the virtual shared disk for each client node on which the virtual shared disk is configured.

You can also use the **vsdatalst**, **ctlvsd**, and **lsvsd** commands with no operands, or SMIT, to see this information.

Certain virtual shared disk, attributes can be modified such as the cache or nocache option. To modify information about a virtual shared disk, use the **Run Command...** action and run the **updatevsdtab** command.

Displaying and Resetting Virtual Shared Disk Statistics

To display virtual shared disk statistics, use the **Run Command...** action and run the **lsvsd -s** command.

Virtual shared disk and hashed shared disk statistics are cumulative. If you want to use them to see the effects of specific parameter changes, you must reset the statistics, make the change, and then read them after a suitable time period.

Resetting statistics is not supported by the IBM Virtual Shared Disk Perspective directly, but you can use the **Run Command...** dialog. To reset the statistics, use the **Run Command...** action and run the command **ctlvsd [-v** vsd_name **|** **-V]**.

You can use the **ctlvsd** or **ctlhsd** commands to reset the statistics.

Displaying and Resetting Virtual Shared Disk Device Driver Statistics

To display the device driver statistics, do the following:

1. Click on a virtual shared disk node from the Nodes pane.
2. Click on the Properties notebook icon in the tool bar or click on **Actions→View and Modify Properties...**
3. Click on the **IBM VSD Node Statistics** or **IBM HSD Node Statistics** page.

You can also use the **statvsd** command.

To reset these statistics, use the **Run Command...** action and run the **ctlvsd -C** command.

Displaying Disk Resource Information

To display disk resource information, use the **Run Command...** action and run the **vsdsklst** command.

Displaying Information about Hashed Shared Disks

To view attributes for hashed shared disks, do the following:

1. Click on the icon in the IBM HSDs pane
2. Click on the Properties notebook icon in the tool bar

The notebook has pages for the following information:

- IBM HSD Attributes

  This page shows the name of the hashed shared disk, stripe size, and protect LVCB option setting.

- IBM HSD Clients

  This page lists the client node names.

- IBM VSD List

  This page lists the virtual shared disks that make up the hashed shared disk.

You can also use the **hsdatalst**, **ctlhsd**, **lshsd**, and **lsvsd** commands with no operands, or SMIT, to see this information.

Chapter 6. Managing and Monitoring Virtual Shared Disks

This chapter tells you how to do the following tasks:

- Control the IBM Recoverable Virtual Shared Disk subsystem.
- Change the states of virtual shared disks (do only when you do not have the IBM Recoverable Virtual Shared Disk subsystem operating.)
- Monitor virtual shared disks.
- Dynamically refresh the IBM Recoverable Virtual Shared Disk subsystem after changing an existing virtual shared disk configuration.
- Make changes to twin-tailed volume groups.
- Set up new physical disks.
- Recable without adding new physical disks.
- Run virtual shared disk diagnostics.
- Collect information for problem determination.

Controlling the IBM Recoverable Virtual Shared Disk Subsystem

To control the activity of the IBM Recoverable Virtual Shared Disk subsystem, you can do the following:

1. Click on a node in the Nodes pane
2. Click on **Actions→Control the IBM RVSD Subsystem...**

To query detailed status about the IBM Recoverable Virtual Shared Disk subsystem, you can do the following steps:

1. Click on a node in the Nodes pane
2. Click on **Actions→Run Command...** and run the **ha.vsd query** command

Changing the States of Virtual Shared Disks

Changes to your configuration or problems in your system, application, or network involve activities that move your virtual shared disks from one state (such as stopped, suspended, or active) to another. If you are not using the IBM Recoverable Virtual Shared Disk component, you might have to perform an activity that results in such a change. (The IBM Recoverable Virtual Shared Disk subsystem changes the states of your virtual shared disks for you, automatically.)

Several activities result in a virtual shared disk state change.

Starting a Virtual Shared Disk

Starting puts a stopped virtual shared disk in the active (and available) state. (This is equivalent to preparing and resuming a virtual shared disk.) Note that for a virtual shared disk to be usable, it must be in the active state on both the server and client nodes.

Preparing a Virtual Shared Disk

Preparing puts a stopped virtual shared disk in the suspended state. In the suspended state, open and close requests are honored. Read and write requests are held until the virtual shared disk is brought to the active state.

Resuming a Virtual Shared Disk

Resuming a virtual shared disk puts a suspended virtual shared disk in the active state. The virtual shared disk remains available and read and write requests that were held are resumed.

Suspending a Virtual Shared Disk

Suspending puts an active virtual shared disk in the suspended state. The virtual shared disk remains available. Read and write requests that were active are suspended and held. All read and write requests subsequent to those that were active are also held.

Stopping a Virtual Shared Disk

Stopping puts a suspended virtual shared disk in the stopped state. The virtual shared disk becomes unavailable. All applications that have outstanding requests to a stopped virtual shared disk terminate in error.

Unconfiguring a stopped virtual shared disk makes it inaccessible. It does not, however, undefine or change the definition information for the virtual shared disk in the SDR.

Changing a Virtual Shared Disk State Using Actions

Note: If you use IBM Recoverable Virtual Shared Disk, do not directly change the state of the virtual shared disks. IBM Recoverable Virtual Shared Disk handles state changes for you. Making such changes could cause the recovery process that IBM Recoverable Virtual Shared Disk manages to fail.

To change the state of a virtual shared disk using the IBM Virtual Shared Disk Perspective graphical user interface, you can do the following:

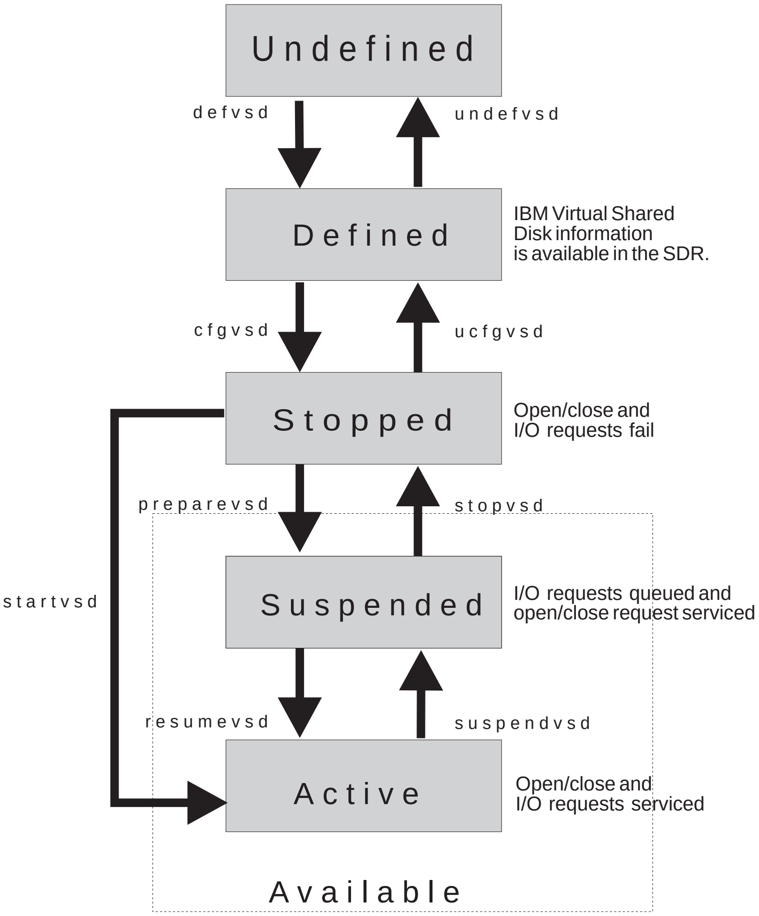
1. Click on the virtual shared disk node icon to select it
2. Click on **Actions→Change IBM VSDs State...**

A dialog window is opened. It contains two selection boxes. The one on the right has a list from which you can choose virtual shared disks. The one on the left collects your choices. You can set them to active, suspended, or stopped.

Changing a Virtual Shared Disk State Using Commands

Note: If you use IBM Recoverable Virtual Shared Disk, do not issue any commands that change the state of the virtual shared disks (**cfgvsd**, **cfghsdvsdstartvsd**, **preparevsd**, **resumevsd**, **suspendvsd**, **stopvsd**, **ucfgvsd**, or **ucfghsdvsd**). IBM Recoverable Virtual Shared Disk handles state changes for you. Issuing any of these commands could cause the recovery process that IBM Recoverable Virtual Shared Disk manages to fail.

Figure 16 on page 49 shows the commands that move virtual shared disks from one state to another.



Undefined
defvsd
undefvsd
Defined
IBM Virtual Shared Disk information is available in the SDR.
cfgvsd
ucfgvsd
Stopped
Open/close and I/O requests fail
preparevsd
stopvsd
startvsd
Suspended
I/O requests queued and open/close request serviced
resumevsd
suspendvsd
Active
Open/close and I/O requests serviced
Available

Note: Methods on arrows cause transitions

Figure 16. Virtual Shared Disk States and Associated Commands

Monitoring Virtual Shared Disks

The objects to be monitored are virtual shared disks on nodes. There are subjects of interest to monitor which are always available to virtual shared disk nodes as conditions and you can just start monitoring them on a per node basis. There are other subjects which you can choose to monitor but you must first prepare them for monitoring. You must make them available for monitoring on a node each time you monitor them. You must also create conditions for them the first time you decide to monitor them.

The subjects are called resource variables. Resource variables are based on the same information that is displayed by the **lsvsd -l**, **lsvsd -s**, and **statvsd** commands. Some of the information from these commands is available as static attributes which you can only view. Resource variables are dynamic attributes that might change over time due to events and are suitable for monitoring.

Conditions are based on resource variables. The resource variables that are always available for all virtual shared disk nodes are:

IBM.PSSP.VSDdrv.num_suspended
IBM.PSSP.VSDdrv.num_active

IBM.PSSP.VSDdrv.num_not_active
IBM.PSSP.VSDdrv.num_stopped
IBM.PSSP.VSDdrv.RVSD_status

There are many other virtual shared disk resource variables which you might want to prepare for monitoring. You can see a complete list when you are using the Event Perspective to create a condition (see "Preparing to Monitor Virtual Shared Disk Statistics" on page 51).

The conditions that are provided by default in the IBM Virtual Shared Disk Perspective graphical user interface are:

hasInactiveIBMVSDs
rvsdInRecovery

This rest of this section tells you how to:

- View virtual shared disk dynamic attributes using the IBM Virtual Shared Disk Perspective table view.
- Prepare to monitor virtual shared disk and device driver statistics that are not always available.
- Monitor virtual shared disk conditions on nodes.

Viewing Dynamic Attributes

Though using the IBM Virtual Shared Disk Perspective table view is technically not monitoring, some virtual shared disk attributes can change dynamically as a result of events and can be interesting to watch. By having the Nodes pane in table view, you can keep an eye on dynamic attributes while you continue with other virtual shared disk activities.

To view counts of active, suspended, or stopped virtual shared disks on each node in table view, do the following:

1. Click on the Nodes pane
2. Click on **View→Show Objects in Table View** or click on the table icon in the tool bar
3. Click on the **IBM VSD Node** tab in the Set Table Attributes for Nodes dialog
4. Click on the attributes you want to view while pressing the <Ctrl> key
5. Click on OK

For example, Figure 12 on page 14 shows a pane in table view with the following attributes:

Host responds
Switch responds
Active IBM VSDs count
Suspended IBM VSDs count
Stopped IBM VSDs count
IBM RVSD subsystem

After you have the Nodes pane in table view, you can toggle between icon and table view by clicking on the table icon in the tool bar. If you decide you want to see other attributes in your table view, click on **View→Set Table Attributes...** and change your selections.

However, the number of dynamically changing attributes available for table view is limited. For more in-depth monitoring, see “Preparing to Monitor Virtual Shared Disk Statistics” on page 51.

Preparing to Monitor Virtual Shared Disk Statistics

More in-depth monitoring is available after using the PSSP Event Perspective to create the additional conditions you want to monitor.

To prepare for monitoring virtual shared disk and device driver statistics that are not always available, you must:

1. Enable monitoring of virtual shared disks on a node by the PSSP Event Management services.

   Do this by running the **monitorvsd** command on each node you want to monitor. More specifically, the command makes available the statistics returned from the **lsvsd -s** command. You can enable monitoring of up to 300 virtual shared disks on one node.

2. Use the Event Perspective graphical user interface to create the virtual shared disk conditions to be monitored.

To create a monitoring condition, do the following:

1. Start the Event Perspective using the **spevent** command
2. Click in the Event Definitions pane
3. Click on **Actions→Create...**
4. Click the Definition tab on the right to make that page current
5. Specify an event definition name
6. In the Condition box, click on **Create Condition...**
7. Specify a name and description
8. Under Resource variable classes, click on IBM.PSSP.VSD
9. Under Resource variable names, click on a resource variable on which to base the condition to monitor (Show Details has a description and an example)
10. Specify an Event expression
11. Click on **Create**
12. In the Condition box, select the name of the condition you just created
13. Select appropriate items under Resource ID
14. To begin monitoring, be sure **Register the event definition** is checked
15. Click on the Notifications tab and the Actions tab to specify, on each page respectively, the notifications and actions that you want to take place when the condition is triggered
16. Click on **Create**

You can create conditions for virtual shared disk statistics and others for virtual shared disk device driver statistics by varying what you select under Resource variable names. See the online help in the Event Perspective for details on creating event definitions and conditions. You can also view the default conditions to see examples of how conditions are defined.

After event definitions have been created and registered, each time the event or the rearm occurs, it is posted in the Global View of Event Notification Log in the Event Perspective. You might find it useful to analyze this log as a history of where and when events occurred.

After the conditions have been created, they are available to be used for monitoring in the IBM Virtual Shared Disk Perspective.

Monitoring Conditions

In the IBM Virtual Shared Disk Perspective, monitoring means that selected objects are continually watched for changes in state. After a change in state occurs, the appearance of the object in the pane changes. For example, the object icon is shown with green when a condition being monitored has not triggered while a red X on the icon means a condition has triggered.

To monitor conditions that are available by default or that you have created, use the IBM Virtual Shared Disk Perspective graphical user interface to do the following:

1. Click on the Nodes pane
2. Click on **View→Set Monitoring...**
3. Click on the IBM VSD Node tab
4. Select the conditions you want to monitor (to select more than one, hold down the <Ctrl> key)
5. Click on **OK**

The icons in the Nodes pane will change color based on the aggregate state of all of the conditions being monitored. To see details about the state of each condition for a particular node, do the following:

1. Click on the node's icon in the Nodes pane
2. Click on **Actions→View and Modify Properties...**
3. Look at the Monitored Conditions page

After you have the monitoring results, stop monitoring the condition. If you used the monitorvsd command to enable monitoring, use it again to disable monitoring.

Refreshing the IBM Recoverable Virtual Shared Disk Subsystem

The IBM Recoverable Virtual Shared Disk daemon can dynamically refresh the IBM Recoverable Virtual Shared Disk subsystem. This means that nodes and virtual shared disks can be dynamically added or removed from an active virtual shared disk configuration without having to stop all applications and unconfigure the virtual shared disks.

In general, the process for changing an existing virtual shared disk configuration is as follows:

1. Follow the steps for whichever change to your configuration is necessary (adding or removing virtual shared disks and nodes).
2. Make the new information known to the other nodes in the virtual shared disk configuration.

- Click on the icon of a node that already has the IBM Recoverable Virtual Shared Disk subsystem running
- Click on **Actions→Control IBM RVSD Subsystem...**
- Click on **Refresh→OK**

The refresh support of the IBM Recoverable Virtual Shared Disk subsystem automatically configures all the virtual shared disks on all the virtual shared disk nodes and starts them.

Making Changes to Twin-Tailed Volume Groups

This section describes how to make a logical volume configuration change to your virtual shared disks, such as extending a volume group, changing the size of your virtual shared disks, or adding a new physical disk, and coordinate the timestamps so the IBM Recoverable Virtual Shared Disk subsystem does not incur unnecessary overhead during recovery processing. If you make a change to your virtual shared disks on twin-tailed volume groups that are managed by the IBM Recoverable Virtual Shared Disk subsystem without resetting the timestamps, any later recovery processing will cause the volume groups to be exported and then imported to the secondary nodes. This will occur even if you have made the updates to the virtual shared disks at both nodes.

Timestamps are maintained on the primary node, the secondary node, and the virtual shared disk itself.

Making a Change to Only One Side of a Twin-Tailed Volume Group

This procedure describes how to make a change to one side (side A) of a twin-tailed volume group **without** varying off the volume group on that side.

1. Make the change.
2. Issue

   ```
   vsdvgts volume_group_name
   ```

   on side A. This reads the volume group timestamp from the disk and saves the value. The next time the IBM Recoverable Virtual Shared Disk recovery scripts vary on this volume group on this node, the timestamps will be the same and the overhead of importing the volume group will be avoided. If a failover occurs, the timestamps will be different on side B, so the volume group will be exported and then imported.
3. If virtual shared disks have been added, use the virtual shared disk Perspective to configure and start the virtual shared disks on all the nodes that need to be aware of this virtual shared disk, or issue the appropriate virtual shared disk commands.
4. Start IBM Recoverable Virtual Shared Disk if it is not running, using the **ha_vsd reset** command.

Making a Change to Both Sides of a Twin-Tailed Volume Group

This procedure describes how to make a change to both sides (sides A and B) of a twin-tailed volume group **without** varying off the volume group on side A.

1. Make the change, issuing all the appropriate AIX commands on side B to reflect the changes made on side A (the volume group on side A is varied online).
2. Issue

   ```
   vsdvgts -a volume_group_name
   ```

   on side A. This reads the volume group timestamp from the disk and saves the value for both sides A and B, since the same changes have been made on both. The next time the IBM Recoverable Virtual Shared Disk recovery scripts vary on this volume group on either node, the timestamps will be the same and the overhead of importing the volume group will be avoided.
3. If virtual shared disks have been added, use the virtual shared disk Perspective to configure and start the virtual shared disks on all the nodes that need to be aware of this virtual shared disk, or issue the appropriate virtual shared disk commands.
4. Start IBM Recoverable Virtual Shared Disk if it is not running, using the **ha_vsd reset** command.

Making a Change to Both Sides of a Twin-Tailed Volume Group with the Volume Group Varied Off

This procedure describes how to make a change to both sides (sides A and B) of a twin-tailed volume group where both sides are varied off and you intend to explicitly export and import the volume group.

1. Make the change on side A. On side B, export the volume group and import it (issue **exportvg** and **importvg**).
2. Issue

   ```
   vsdvgts -a volume_group_name
   ```

   on either side. This reads the volume group timestamp from the disk and saves the value for both sides A and B, since the changes made on A have been exported to B. The next time the IBM Recoverable Virtual Shared Disk recovery scripts vary on this volume group on either node, the timestamps will be the same and the overhead of importing the volume group will be avoided.
3. If virtual shared disks have been added, use the virtual shared disk Perspective to configure and start the virtual shared disks on all the nodes that need to be aware of this virtual shared disk, or issue the appropriate virtual shared disk commands.
4. Start IBM Recoverable Virtual Shared Disk if it is not running, using the **ha_vsd reset** command.

Setting up Twin-Tailed Physical Disks

Follow these steps to migrate virtual shared disk data to a twin-tailed recoverable virtual shared disk.

1. Bring down any application that uses the old virtual shared disks for the duration of this migration.
2. Suspend, stop, and unconfigure all virtual shared disks involved.
3. Add the new physical disks to your hardware configuration, using the instructions that came with them.
4. Create the new volume groups, logical volumes, and virtual shared disks.
5. Perform test reads and writes to make sure you've successfully created your new virtual shared disks.
6. Copy your data from your existing (old) virtual shared disks to the newly-defined virtual shared disks by copying the data directly from one logical volume to the other, using the new volume group name. For example:

   ```
   cplv [-v new_volume_group]-y new_logical_volume \
   old_logical_volume
   ```

7. After making sure that the data is copied to the new logical volume, remove the old virtual shared disk.
8. Issue a reset to make all the virtual shared disks in the configuration active and start the IBM Recoverable Virtual Shared Disk subsystem.

Note: If you are using data striping, the process of creating the new hashed shared disks also creates the underlying virtual shared disks.

Recabling without Adding New Physical Disks

Follow these steps to use the IBM Recoverable Virtual Shared Disk subsystem when you are recabling to add the connection to a secondary node but do not need to add new physical disks.

- Stop any applications that use the affected volume groups.
- Suspend, stop, and unconfigure all affected virtual shared disks.
- Add a physical connection from each of the physical disks in the IBM Recoverable Virtual Shared Disk configuration to a secondary node.
- Import the volume groups from the primary node to the secondary, using **smit importvg**.
- If the secondary nodes have not previously been designated as virtual shared disk nodes, designate the nodes as virtual shared disk nodes, using the IBM Virtual Shared Disk Perspective graphical user interface or the **vsdnode** command.
- Redefine the virtual shared disks with the secondary nodes identified.
- Issue a reset to make all the virtual shared disks in the configuration active and start the IBM Recoverable Virtual Shared Disk subsystem.

Running Virtual Shared Disk Diagnostics

If you have problems with your virtual shared disk or hashed shared disk configuration, your IBM service representative might ask you to look at the virtual shared disk diagnostics.

To run and display diagnostics, do the following:

- Click on the Nodes pane
- Click on **Actions**→**Display IBM VSD Diagnostics**

The diagnostics compare the settings of some virtual shared disk parameters across all the nodes in a virtual shared disk configuration and look for discrepancies that could affect the operation of your virtual shared disks.

You can also use the **vsddiag** command for this purpose.

Collecting Information for Problem Determination

To collect information needed by IBM service for problem diagnosis, at the time of the problem use the **Run Command...** action and run the **vsd.snap** command.

Chapter 7. Unconfiguring and Removing Virtual Shared Disks and Nodes

This chapter tells you how to eliminate virtual shared disks, hashed shared disks, or nodes from an existing virtual shared disk configuration. There are several steps that might be used individually or in sequence. These steps are:

1. Stop activity on virtual shared disks or hashed shared disks that are to be removed.
2. Unconfigure the virtual shared disks or hashed shared disks being removed.
3. Remove the virtual shared disks or hashed shared disks.
4. Remove a node from the configuration

Depending on your circumstances, you might not complete all the steps. For instance, to remove a virtual shared disk, complete the first three steps. To remove a virtual shared disk server node from the configuration, complete all the steps. To remove a client node, you need to complete steps 1, 2, and 4 on the node to be removed.

Stopping Virtual Shared Disk Activity

To stop activity, do the following:

1. Shut down any applications that might use any virtual shared disks or hashed shared disks that are to be removed.
2. If you are removing the virtual shared disks or hashed shared disks and you need the data, back it up.
3. Stop virtual shared disk activity on the server node and any node that serves as a backup for the volume groups being served by this node.
    - Click on the virtual shared disk server node in the Nodes pane
    - If the IBM Recoverable Virtual Shared Disk subsystem is running, click on **Actions→Control IBM RVSD Subsystem...→Stop→OK**

      Alternatively, you can use the command **ha.vsd stop**.
    - If the IBM Recoverable Virtual Shared Disk subsystem is not running click on **Actions→Change IBM VSDs State...**. Then select the virtual shared disks to stop and click on **Stopped→OK**

      Alternatively, you can use the **suspendvsd** and **stopvsd** commands.

See the book *PSSP: Command and Technical Reference* for syntax of the commands.

Unconfiguring Virtual Shared Disks and Hashed Shared Disks

There are primarily two occasions when you need to unconfigure virtual shared disks after you place them in the *stopped* state:

- When you are preparing to remove them or the node on which they are configured.

- When you want to unload the IBM Virtual Shared Disk device driver from the kernel so that you can load a new one.

To unconfigure all virtual shared disks defined on a node, you can do the following:

1. Click on the node to be removed in the Nodes pane
2. Click on **Actions→Unconfigure IBM VSDs...** or **Unconfigure IBM HSDs...**
3. Select the virtual shared disks or hashed shared disks to unconfigure and click on **OK**

To unconfigure a virtual shared disk, you can do the following:

1. Click on the icon in the IBM VSDs pane to select it
2. Click on **Actions→Unconfigure...**
3. Select the nodes from which to unconfigure the virtual shared disks

To unconfigure a hashed shared disk and the underlying virtual shared disks, you can do the following:

1. Click on the icon in the IBM HSDs pane to select it
2. Click on **Actions→Unconfigure...**
3. Select the nodes from which to unconfigure the virtual shared disks

Alternatively, you can use the **ucfghsdvsd** command which also unconfigures the underlying virtual shared disks. To unconfigure just a hashed shared disk, use the **ucfghsd** command. To unconfigure a virtual shared disk, use the **ucfgvsd** command. See the book *PSSP: Command and Technical Reference* for syntax of the commands.

Removing Virtual Shared Disks and Hashed Shared Disks

CAUTION:
Removing a virtual shared disk or hashed shared disk also removes the logical volumes associated with that virtual shared disk and causes you to lose access to all the data on it. If you need the data on the virtual shared disk that is to be removed, first shutdown any applications that might use it and then back it up. Then, stop all activity to the virtual shared disk and unconfigure it (see "Stopping Virtual Shared Disk Activity" on page 57 and "Unconfiguring Virtual Shared Disks and Hashed Shared Disks" on page 57.)

To remove a virtual shared disk, you can do the following:

- Click on the icon in the IBM VSDs pane to select it
- Click on **Actions→Remove**

Alternatively, you can use the **removevsd** command.

To remove a hashed shared disk and the underlying virtual shared disks, you can do the following:

- Click on the icon in the IBM HSDs pane to select it
- Click on **Actions→Remove**

Alternatively, you can use the **removehsd** command. The **removehsd** command also removes the underlying virtual shared disks. See the book *PSSP: Command and Technical Reference* for syntax of the commands.

Removing a Node from a Virtual Shared Disk Configuration

To remove a node from an existing virtual shared disk configuration, do the following:

1. Complete the steps in “Stopping Virtual Shared Disk Activity” on page 57
2. Complete the steps in “Unconfiguring Virtual Shared Disks and Hashed Shared Disks” on page 57 for the node to be removed
3. If this is a virtual shared disk server node, do the following:
    a. Complete the steps in “Removing Virtual Shared Disks and Hashed Shared Disks” on page 58
    b. Remove global volume group information
        - Click on the node to be removed in the Nodes pane
        - Click on **Actions→Run Command...**
        - Issue "vsdelvg [-f] *global group name* ..."
4. Remove the virtual shared disk node designation
    - Click on the node to be removed in the Nodes pane
    - Click on **Remove IBM VSD Node Designation**
5. If you stopped the IBM Recoverable Virtual Shared Disk subsystem, continue with “Refreshing the IBM Recoverable Virtual Shared Disk Subsystem” on page 52.

See the book *PSSP: Command and Technical Reference* for syntax of the commands.

Chapter 8. Performance and Tuning Considerations for Virtual Shared Disks and Hashed Shared Disks

Those of you who are system administrators or application programmers need to be aware of the performance characteristics of virtual shared disks and hashed shared disks and of how you can affect those characteristics.

As you read this section, remember to take the performance characteristics of your database subsystem and applications into account.

Tuning Virtual Shared Disk Performance

The IBM Virtual Shared Disk device driver passes all its requests to the underlying Logical Volume Manager subsystem. Before you tune the virtual shared disk, check that the I/O subsystem is not the bottleneck. See *AIX Performance and Tuning Guide* for information on I/O subsystem performance and tuning. If an overloaded I/O subsystem is degrading your system's performance, tuning the virtual shared disk will not help. In the case of I/O subsystem overload, consider spreading the I/O load over more disks or nodes.

For best performance, do the following:

1. Use the defaults when defining virtual shared disks (refer to “Designating Nodes as IBM VSD Nodes” on page 31)
2. Turn IBM Virtual Shared Disk caching off if you are not using your system for online transaction processing.
3. Do a performance run to collect statistics on the virtual shared disks, your I/O subsystem, and the CPU on all nodes (or use Performance Monitor to collect information during normal system operation.) Issue **statvsd** several times during the performance run and compare the values for the various statistics. Use **iostat** to check your disk utilization. If you notice increasing numbers of queued requests, do the following:

   - If the system is I/O bound (meaning your disks are more than 50% utilized), add disks.
   - If the system is CPU bound, add nodes or spread the workload on the virtual shared disk server nodes. You can use the Hashed Shared Disk data striping subsystem to spread the workload.
   - If nodes are doing excessive swapping due to insufficient pinned memory, which is used by pbufs, buddy buffers, cache, and the switch pool, reduce cache size.
   - If requests are queuing because of a shortage of buddy buffers or pbufs, you might have disk bottlenecks. Spread the data or add disks to the server nodes.
   - If your application issues requests that are larger than 64KB, set your maximum buddy buffer size to 256KB.
   - If you see too many retries, check for disk bottlenecks. If that is not the problem, consider increasing the switch pool size (see “mbufs and the Switch Pool” on page 64).

4. Reset the statistics counter by running the **ctlvsd** command (you can use the **Run Command...** action of the IBM Virtual Shared Disk Perspective graphical user interface).
5. Do another performance run.

You should generally operate with IBM Virtual Shared Disk caching off. Memory is better allocated to the operating system itself, for paging, and to the cache belonging to the application using the virtual shared disk. To turn IBM Virtual Shared Disk caching off, do the following:

1. Shut down your applications that use virtual shared disks and stop activity (see “Stopping Virtual Shared Disk Activity” on page 57)
2. If you do not use the IBM Recoverable Virtual Shared Disk subsystem, unconfigure the virtual shared disks (see “Unconfiguring Virtual Shared Disks and Hashed Shared Disks” on page 57)
3. Select one or more nodes
4. Use the **Run Command...** action and run the **updatevsdtab** command to change the *cache/nocache* option to nocache
5. If you do not use the IBM Recoverable Virtual Shared Disk subsystem, configure the virtual shared disks (see “Configuring Virtual Shared Disks or Hashed Shared Disks” on page 40)
6. If you do use the IBM Recoverable Virtual Shared Disk subsystem, refresh the virtual shared disk configuration (see “Refreshing the IBM Recoverable Virtual Shared Disk Subsystem” on page 52)
7. Restart your applications.

If you do use caching, remember that the IBM Virtual Shared Disk component only caches 4KB requests aligned on 4KB boundaries.

See the *PSSP: Command and Technical Reference* for command options and syntax.

Tunable Parameters Related to Virtual Shared Disks

The main tunable parameters are:

- Logical Volume Manager (striping and other characteristics)
- IP communications adapter (usually the switch)
- The IBM Virtual Shared Disk cache buffer
- Buddy buffer
- Maximum I/O request size
- Request blocks
- pbufs
- mbufs

These are discussed with relevant tuning considerations in the following sections. You should also consider the tunable characteristics of the applications that use virtual shared disks, especially the use of buffers.

Logical Volume Manager (LVM) Tuning Considerations

There is always an associated logical volume for every virtual shared disk defined and configured in a system. Every virtual shared disk I/O request eventually becomes an I/O request to the associated logical volume (unless you get a cache hit at the server). This mapping of virtual shared disk I/O requests to the associated logical volume I/O requests is handled transparently by the IBM Virtual Shared Disk subsystem. All the performance tuning considerations that apply to a logical volume also apply to a virtual shared disk. Refer to *AIX System Management Guide: Operating Systems and Devices* and *AIX Performance and Tuning Guide* for information on the performance tuning of logical volumes.

SP Switch Considerations

If you configure the virtual shared disk nodes to use the SP Switch (**css0**), set the maximum IP message size (utilized by the virtual shared disk driver) set to 61440 (60KB). Ensure the maximum buddy buffer size is 256KB, since the value you assign to *maximum_buddy_buffer_size* in the SDR also limits the maximum size of the request that the IBM Virtual Shared Disk subsystem sends across the nodes. For example, if you have:

- A request from a client to write 256KB of data to a remote virtual shared disk
- A maximum buddy buffer size of 64KB
- A maximum IP message size of 60KB

the following transmission sequence occurs:

1. The IBM Virtual Shared Disk subsystem divides the 256KB of data into four 64KB requests in four buddy buffers
2. Each 64KB block of data becomes one 60KB packet and one 4KB packet for transmission to the server via IP
3. At the server, the eight packets are reassembled into four 64KB blocks of data, each in a 64KB buddy buffer
4. The server then has to perform four 64KB write operations and return four acknowledgements to the client.

A better scenario for the same write operation would use the maximum buddy buffer size:

- The same 256KB client request to the remote virtual shared disk
- The maximum buddy buffer size of **256KB**
- The maximum IP message size of 60KB

Producing the following transmission sequence:

1. The 256KB request becomes four 60 KB packets and one 16KB packet for transmission to the server via IP
2. At the server, the five packets are reassembled into one 256KB block of data in a single buddy buffer
3. The server then performs one 256 KB write operation and returns an acknowledgement to the client.

The second scenario is preferable to the first because the I/O operations at the server are minimized. A perfect scenario would be one where the IBM Virtual Shared Disk component does not use buddy buffers at all — when the client request is less than or equal to the maximum IP message size. For example:

- A request from a client to write 60KB of data to a remote virtual shared disk server
- A maximum IP message size of 60KB

When you use the switch, send pool clusters are used instead of buddy buffers as long as the request size is less than the *ip_message_size*, as in the example just cited. Buddy buffers are used only when a shortage in the switch buffer pool occurs or when the size of the request is greater than the *ip_message_size*. If you see buddy buffer shortages, instead of increasing your buddy buffers, you need to increase your switch send pool size. See “mbufs and the Switch Pool.”

mbufs and the Switch Pool

mbufs are used for data transfer between the client and the server nodes by the IBM Virtual Shared Disk subsystem's own UDP-like internet protocol. If you are using the switch (**css0**) as your communications adapter, the IBM Virtual Shared Disk component uses mbuf clusters to do I/O directly from the switch's send and receive pools.

If you notice that the indirect I/O statistic (from the IBM Virtual Shared Disk Perspectives Statistics notebook page or from the output of the **vsdstat** command) is incremented consistently, run **errpt** to check the error log. If you see the line:

```
IFIOCTL_MGET(): send pool shortage
```

you should consider increasing the size of the send and receive pools.

To check the current sizes of the send and receive pools, type:

```
lsattr -l css0 -E
```

The default size for each pool is 524288 bytes (512KB).

To change the sizes of the pools to 4MB, type:

```
/usr/lpp/ssp/css/chgcss -l css0 -a spoolsize=4194304
/usr/lpp/ssp/css/chgcss -l css0 -a rpoolsize=4194304
```

This command increases the send and receive pool size to 4MB.

Note: You must reboot the node for the new sizes to take effect.

IBM suggests you allow 16MB for mbufs and clusters. You can set this value by issuing:

```
no -o thewall=16777216
```

To see what your current system mbuf setting is, type:

```
no -a | grep thewall
```

System performance considerations regarding mbufs and mbuf clusters also apply to virtual shared disk environments. See *AIX Performance and Tuning Guide* for more information.

Buddy Buffers

The virtual shared disk server node uses the buddy buffer to temporarily store data for I/O operations originating at a client node and to handle requests that are greater than the *ip_message_size*. In contrast to the data in the cache buffer, the data in a buddy buffer is purged immediately after the I/O operation completes.

Note: Buddy buffers are used only when a shortage in the switch buffer pool occurs or on certain networks (for example, the Ethernet).

The values associated with the buddy buffer are:

- Minimum buddy buffer size allocated to a single request
- Maximum buddy buffer size allocated to a single request
- Total size of the buddy buffer

These values can be set using the IBM Virtual Shared Disk Perspective graphical user interface or the **vsdnode** command.

Buddy buffer space is allocated in powers of two. If an I/O request size is not a power of two, the smallest power of two that is larger than the request is allocated. For example, for a request size of 24KB, 32KB are allocated on the server.

If you are using the switch as your adapter for virtual shared disks, we recommend setting 4096 (4KB) and 262144 (256KB), respectively, for minimum and maximum buddy buffer size allocated to a single request.

To define the total size of the buddy buffer, consider the remote I/O throughput for the server and specify the number of maximum-sized buddy buffers in the buffer. For example, if you expect the server to serve 10MB per second on behalf of remote clients and a request spends an average of 60 milliseconds on the server, multiply 10MB per second X 0.06 second and, for safety, double or triple the result for a total buddy buffer size of 1.8MB (eight 256KB maximum buddy buffers).

If the virtual shared disk statistics consistently show requests queued waiting for buddy buffers, do not add more buddy buffers. Instead, increase the size of the switch send pool (see “mbufs and the Switch Pool” on page 64) or spread the data over disks attached to other nodes, to prevent a bottleneck.

Note: If your application uses the fastpath option of asynchronous I/O, the maximum buddy buffer size must be greater than or equal to 128KB. Otherwise, you will get EMSGSIZE "Message to long" errors.

Buffer Allocation

Your application should make all new allocated buffers on the page boundary. If your I/O buffer is not aligned on a page boundary, the IBM Virtual Shared Disk device drivers will not parallelize I/O requests to underlying virtual shared disks and performance will be degraded.

The Cache Buffer

Each IBM Virtual Shared Disk device driver, that is, each node, has a single cache buffer, shared by all cacheable virtual shared disks configured on and served by the node. The cache buffer is used to store the most recently accessed data from the cached virtual shared disks (associated logical volumes) on the server node. The objective is to minimize physical disk I/O activity. If the requested data is found in the cache, it is read from the cache, rather than the corresponding logical volume.

Data in the cache is stored in 4KB blocks. The content of the cache is a replica of the corresponding data blocks on the physical disks. Write-through cache semantics apply; that is, the write operation is not complete until the data is on the disk.

When you create virtual shared disks with the IBM Virtual Shared Disk Perspective graphical user interface or the **createvsd** command, you can specify the *cache* option or the *nocache* option. IBM suggests that you specify **nocache** (or make the cache buffer small) in most instances (especially in the cases of read-only or other than 4KB applications) for the following reasons:

- Requests that are not exactly 4KB and not aligned on a 4KB boundary will bypass the cache buffer, but will incur the overhead of searching the cache blocks for overlapping pages.
- Every 4KB I/O operation incurs the overhead of copying into or out of the cache buffer, as well as the overhead of moving program data from the processor cache due to the copy.
- There is overhead for maintaining an index on the blocks cached.

If you are running an application that involves heavy writing followed immediately by reading, it might be advantageous to turn the cache buffer on for some virtual shared disks on a particular node. Choose the appropriate size for the cache based on the expected throughput and the expected time lag between writes and reads. For example, if the expected throughput is 100 4 KB-aligned I/O operations per second and reads lag writes by 0.5 seconds, calculate the cache buffer size by multiplying 100 X 0.5 and, as a safety factor, double it for a total of 100 cache blocks.

The **lsvsd -s** command gives detailed statistics on virtual shared disk cache hits and I/O activities. This will tell you which virtual shared disks are heavily used. See “Monitoring Virtual Shared Disks” on page 49 for information on how to see statistics using the IBM Virtual Shared Disk Perspective and the Event Perspective graphical user interfaces.

Maximum I/O Request Size

The following factors limit the block size that the IBM Virtual Shared Disk subsystem uses to process each I/O request:

- The largest block size the IBM Virtual Shared Disk subsystem will use is the smaller of *max_buddy_buffer_size* or 256KB.
- If the virtual shared disk uses the switch as its adapter, the *max_IP_msg_size* that could be sent is 65024 bytes (63.5KB). IBM suggests the value 61440 (60KB) for the virtual shared disk device driver when **css0** is defined as the virtual shared disk adapter in the SDR. The **ctlvsd -M** command can override the default. The *max_IP_msg_size* should be set to a value that is a multiple of 512 bytes and is less than or equal to 63.5KB (when the switch is used) and less than or equal to 24KB (when the switch is not used). The **statvsd** command displays the current value.

  Note: Setting the *max_IP_msg_size* to more than 24KB when using communication adapters with small MTU (maximum transmission unit) could overflow the adapter driver's internal buffers, causing the IP layer to drop packets. This forces the virtual shared disk device driver to retry, sometimes without success, resulting in a timeout.

Every virtual shared disk I/O request is subject to both limits. For example, with a *max_buddy_buffer_size* of 262144 (256KB) and a *max_ip_msg_size* of 61440 (60KB), if an application requests a single 64KB read to a virtual shared disk served by the local node, the request is passed down to the local logical volume as one 60KB request and one 4KB request.

The atomicity of an I/O operation is gated by the size of the virtual shared disk request, rather than the size of the application request (if the virtual shared disk request is smaller than the application request the application request would be split down to the size of the virtual shared disk request).

Request Blocks

The number of request blocks is the total number of physical I/O operations that have been issued by all processes on a node, but have not completed. The number includes requests to both local and remote devices. Because large requests may be broken up into smaller subrequests, the request block number may be several times greater than the actual number of pending read/write requests.

For example, if I/O requests are:

- issued at a rate of 1000 I/O operations per second
- broken on the average into three pieces
- responded to in 50 milliseconds

Then the algorithm for calculating the number of request blocks is 1000 X 3 X 0.05 for a total of 150 request blocks. You may want to increase the number somewhat as a safety precaution to account for the possibility of workload surges. Specifying an inordinately large number of request blocks could have a negative performance impact. A large number of request blocks could flood the network, causing servers to run out of mbufs and causing unnecessary retransmissions. What constitutes a large number of request depends on how large the request size is and how many nodes are in the system.

Although the **statvsd** command reports the number of times there is no request block available, queueing for request blocks does not necessarily imply a performance bottleneck. If you increased the number or request blocks infinitely, queueing would occur elsewhere in the operating system.

The number of request blocks can be set and changed with the IBM Virtual Shared Disk Perspective graphical user interface or the **vsdnode** and **updatevsdnode** commands respectively.

pbufs

Buffers called pbufs are used for actual physical I/O requests that are submitted to the disks. A pbuf shortage affects overall performance by degrading performance. However, you must also be careful to not exceed your environment limitations.

pbufs are specified as a way of controlling the number of pending device requests for a specific virtual shared disk on its server node. pbufs are specified on a per device basis.

You can set the number of pbufs to be allocated per virtual shared disk by the **Designate as an IBM VSD node...** action, the **rw_request_count** parameter of the **vsdnode** command, or the SMIT **vsdnode_dialog** fast path.

Each virtual shared disk, regardless of its activity and whether the node is a client or server, is allocated these pbufs. Each pbuf is 128 bytes long. You can calculate how much of the kernel heap you need for pbufs using the following formula:

heap_allocated = *nvsd* * *nreq* * 128

where:

nvsd is the number of virtual shared disks
nreq is the number of pbufs you are requesting

Note: Make sure that *heap_allocated* never exceeds the available kernel heap. For example, given that an eighth of the heap is available for pbufs (about 32,000,000 bytes), and 1300 virtual shared disks are configured, the value of *nreq* cannot exceed 192.

To check on the interactions among request blocks, pbufs, and cache blocks using the IBM Virtual Shared Disk Perspective graphical user interface, do the following:

1. With a node selected, click on **Actions→View and Modify Properties...**
2. Select the IBM VSD Node Statistics page
3. View the statistics and decide which parameters to tune

The **statvsd** command also gives detailed statistics on request shortages, pbuf shortages and cache block shortages. You can run this command together with **lsvsd -s** before and after an application execution to determine how to tune these parameters to best fit a particular application workload.

Tuning Hashed Shared Disk Performance

This section presents information for tuning hashed shared disk performance.

Note: For more information on tuning for performance, access the RS/6000 home page at **http://www.rs6000.ibm.com**. Once you get to the home page, look under **Resources** for information about virtual shared disks and hashed shared disks.

Since, all I/O requests are passed to the underlying virtual shared disks, the parameters you select when configuring virtual shared disks affect the performance of hashed shared disks. Therefore, hashed shared disk tuning is discussed in relationship to these additional performance considerations:

- Database (“hot table”) access
- Disk throughput
- CPU overhead
- I/O completion speeds

The parameters that can be tuned to affect performance are:

- Request size
- Stripe size

To take advantage of the parallelism potential, the stripe size should be smaller than the application's request size.

Request latency for a hashed shared disk or virtual shared disk is primarily gated by disk I/O. Disk I/O can be separated into three distinct operations:

- Seek delay
- Rotational delay
- Transfer delay

The smaller the stripe size, the smaller the transfer delay. Hashed Shared Disk data striping allows the system to handle virtual shared disk requests in parallel.

If your system is CPU bound, using the Hashed Shared Disk subsystem (with a stripe size greater than the buddy buffer size) can help only with eliminating I/O bottlenecks. Eliminating I/O bottlenecks might improve overall transaction throughput. You should consider spreading the workload by adding more nodes or disks.

If your system is I/O bound, using the Hashed Shared Disk subsystem might help. However, consider that:

- Processing more information from disks requires more CPU cycles. Striping uses some of these CPU cycles. It is important to stay below the point of diminishing returns when you set the stripe size.
- You should set a stripe size large enough to minimize the chances that the same virtual shared disk could be hit more than once by a parallelized application request.
- Disks should be on separate adapters because adapters (such as SCSI) limit the number of concurrent requests they can send to disks that are sharing the same bus.

The only way to fully optimize a hashed shared disk to your environment is to try out different configurations. The main obstacle in tuning a hashed shared disk is that it is not simple to add or remove virtual shared disks and change the stripe size after the hashed shared disk is loaded with data. The only way to do this is to back up the hashed shared disk (using **dd**), recreate it, and load it with the new parameters. Therefore, give careful consideration to planning how you will use a hashed shared disk. If tuned properly, a hashed shared disk can significantly improve virtual shared disk throughput performance.

Chapter 9. Application Programming Considerations

This chapter tells you how to create or modify your application programs to take advantage of functions in the components of PSSP; IBM Virtual Shared Disk, Hashed Shared Disk, and IBM Recoverable Virtual Shared Disk.

Application Programming Considerations for Virtual Shared Disks

Application programmers who work with virtual shared disks need to be aware of data integrity considerations, the IBM Virtual Shared Disk transmission protocol, and how to get information about the configured virtual shared disks by using C language interfaces.

Data Integrity

There are data integrity considerations involved in accessing data, synchronizing I/O, and checksum processing.

Accessing Data on a Virtual Shared Disk

Use the raw device (also called device special file) to access a virtual shared disk, because the block device can give you stale data from the operating system cache.

Synchronizing I/O

A virtual shared disk is a device. On a single node, you must provide and use your own synchronization mechanisms for your reads and writes to insure data integrity. Since IBM Virtual Shared Disk allows multiple nodes to have simultaneous access to devices, you must provide your own distributed synchronization mechanisms. Neither IBM Virtual Shared Disk nor the SP system software provides a distributed synchronization mechanism.

Checksum Processing on Unreliable Networks

The switch is a reliable communication medium. The switch adapter performs checksum processing in hardware to provide a reliable network. Some networks do not provide checksum processing (for example, the Ethernet). IBM Virtual Shared Disk does provide a checksum processing option for you to use on unreliable networks. You can use the **cksumvsd** command to turn checksum processing on and off in the IBM Virtual Shared Disk device driver. Because the switch provides checksum processing in hardware, do not use **cksumvsd** if you use **css0** as your virtual shared disk communication adapter. For more information refer to *PSSP: Command and Technical Reference*.

Note: Checksum processing must be set the same, either on or off, on all virtual shared disk nodes in a system partition.

IBM Virtual Shared Disk Transmission Protocol

IBM Virtual Shared Disk implements its own UDP-like internet protocol. The requesting node implements an exponential back-off retransmission strategy. If the remote node does not service the request after about 15 minutes of retransmission, IBM Virtual Shared Disk fails the application's I/O request. Suspending and resuming a virtual shared disk (done by IBM Recoverable Virtual Shared Disk) gives all requests a fresh start with the full 15 minutes of retransmission time. The time needed for recovery with IBM Recoverable Virtual Shared Disk does not cause application requests to fail.

Sequence Numbers

Each node keeps an expected and outgoing sequence number for all other nodes. Each time a node sends a request message to another node, it increments its outgoing sequence number for that node. If a node receives a message, and its sequence number is not within the valid range, based on the expected number for that node, the message is discarded. The valid range is defined as the current value of the expected number for the node, plus a fairly large window size. When a node receives a valid request message from another node, the receiving node resets its expected sequence number for that node to the sequence number in the message.

IBM Recoverable Virtual Shared Disk maintains and resets sequence numbers for its users.

Sequence numbers become significant if a node in the virtual shared disk cluster is rebooted after it has performed virtual shared disk I/O. When the node comes back up, all its sequence numbers will be reset to 0, but all other nodes in the system partition will expect numbers greater than the last sequence number they received from the node before it rebooted, that is, nonzero. Therefore, none of the other existing nodes that received messages from the node before it crashed will now accept requests from the rebooted node because the sequence number is below the expected value. The rebooted node continues to retransmit, increasing the outgoing sequence number each time, until it eventually reaches the expected range on all the other nodes. Some requests, however, may exhaust their retry limits and fail. The rebooted node will probably accept requests from the existing nodes as their sequence number will be within the range of the expected count (0) and the (large) window size.

When one node's sequence numbers are not synchronized with the rest of the virtual shared disk nodes, it will look like all virtual shared disk I/O to remote servers from the out-of-synch node is hung. After about 15 minutes of retransmissions, the I/O will start to fail if the system administrator does not intervene and reset the sequence numbers.

Checking Sequence Numbers: To check sequence numbers from the IBM Virtual Shared Disk Perspective:

- Click on the applicable node in the Nodes pane
- Click on the Properties notebook icon in the tool bar
- Look at the IBM VSDs Node Statistics page

Alternatively, you can run the **statvsd** command on client and server nodes. (See the **statvsd** command reference pages for more details.)

Resetting Sequence Numbers: To reset sequence numbers from the IBM Virtual Shared Disk Perspective:

- Select the applicable client and server nodes in the Nodes pane
- Click on **Actions→Run Command...**
- Run the **ctlvsd** command as described below.

To reset sequence numbers with line commands, use the **-R** and **-r** options of the **ctlvsd** to reset the sequence number of all or some of the nodes to 0 (both the expected and the outgoing). Before issuing **ctlvsd**, put the virtual shared disk in suspended state. (See the **ctlvsd** command reference pages for more details.)

Getting Virtual Shared Disk Information with C Interfaces

Note: The header file **/usr/include/vsd_ioctl.h** is installed with the **vsd.vsdd** option. Structures defined in **vsd_ioctl.h** might change from release to release. If ioctl fails, you might have to recompile.

The IBM Virtual Shared Disk base code has a sample directory: **/usr/lpp/csd/samples**.

This directory contains the following files:

vsdinfo.c When compiled, a binary is generated that shows information for the **vsd_name** given from the command line.

vsdsinfo.c When compiled, a binary is generated that lists information for all configured virtual shared disks.

Application Programming Considerations for Hashed Shared Disks

Read this section for information on application programming performance with Hashed Shared Disk and on how to use Hashed Shared Disks through C language interfaces.

Allocating I/O Buffers

The address of your I/O buffer must be aligned on a 4KB boundary. If it is not, the Hashed Shared Disk read and write routines will **not** parallelize I/O requests to underlying virtual shared disks. Performance will be less than optimal.

Set the environment variable MALLOCTYPE to **3.1** to make all new allocated buffers (MALLOC) at the page boundary.

Getting Hashed Shared Disk Information with C Interfaces

The Hashed Shared Disk base code has a sample directory: **/usr/lpp/csd/samples**.

This directory contains the following files:

hsdinfo.c When compiled, this file generates a binary that shows hashed shared disk information for the **hsd_name** given from the command line.

hsdsinfo.c When compiled, this file generates a binary that lists information for all configured hashed shared disks.

Note: The header file **/usr/include/hsd_ioctl.h** is installed with the **ssp.csd.hsd** image. Structures defined in **hsd_ioctl.h** may change from release to release. This may require that you recompile your applications that utilize C interfaces to the HSD to run on new releases. If you see IOCTL failures in your application after migrating to a new release, try recompiling the application.

Making Your Application Recoverable

You can code your program to use the **hc** daemon of IBM Recoverable Virtual Shared Disk to aid in application recovery. A data management subsystem is the type of program that might use **hc**'s services.

Your program should include the following **hc.h** header file:

```
#include <hc.h>
```

Your program connects to a socket where **hc** is listening. If a node fails or the client application on a node fails, **hc** sends a reconfiguration message according to the protocol defined in the **hc.h** header file. Your program should check for messages from that socket and react accordingly.

The hc subsystem can support multiple applications by running multiple instances of hc. Each hc is invoked by a different **hc.vsd** script.

To create a new instance of the hc subsystem, make a copy of the **hc.vsd** script (from **usr/lpp/csd/bin**) and rename it; for example, *hc.myappname*. Then edit the newly-created *hc.myappname* and change the *INSTANCE* variable to *myappname*. You should also export the path to the well-known socket for communication with your application, inserting the following line into the script:

```
export CLIENT_PATH=myappath/
```

where PATH is an absolute pathname. The hc subsystem then creates the socket **myappath/myappname** to communicate with your application.

As a convenience, the **hc** program runs an optional script:**/usr/lpp/csd/bin/hc.activate** once when it initializes (note that **/usr/lpp/csd/bin/hc.deactivate** automatically runs first). The **hc.activate** script is passed two parameters; the local node number and the path name of the UNIX domain stream socket that it serves. Your recovery program is started according to the value you gave **INSTANCE**. If `INSTANCE=myappname`, for example, **hc** calls **myappname.activate**.

If the script doesn't complete, **hc** won't communicate with the application. An application that checks connectivity through **hc** running on the other clients can't perform that check until the **hc.activate** script completes. If you need **hc** to continue processing, fork a background process from the **hc.activate** script. Then exit the **hc.activate** script with an exit code of 1, which indicates that communication with the application does not have to be complete for **hc** to continue processing. **hc.activate** is executed only once.

The **hc** program also runs the **/usr/lpp/csd/bin/hc.deactivate** script upon detecting failure of its client. The same parameters are passed as are passed to **/usr/lpp/csd/bin/hc.activate**, letting you restart the client if you desire. The client may fail either by exiting or by failure to respond to a **ping** message within 10 minutes.

Note: Ten minutes is the default for the **PING_DELAY**. That is, if a client application fails to respond to a ping sent from the **hc** program within 10 minutes, **hc** will consider the application to have failed and will invoke **hc.deactivate.**

After **hc.deactivate** runs, it sends a **node down** message to the other nodes.

Some application programs running under some system loads may require a longer **PING_DELAY** timeframe. To change the **PING_DELAY**, edit the **hc.vsd** script. For example, if you wish to increase the **PING_DELAY** to 15 minutes, change the line in the **hc.vsd** script that begins **export SCRIPT_PATH** to:

```
export
SCRIPT_PATH=...PING_DELAY=900
```

If **hc** fails, the application will receive a zero-length message over the socket and should shut itself down.

Applications are responsible for cleaning up system resources when they complete or fail; **hc.deactivate** should be used for this purpose.

Preserving Data Integrity During Application Recovery

Rather than using **hc**, you can use fencing. Application recovery can be independent.

If a recoverable database application is running in a system and one of the nodes in that system fails, any data that had been locked by the failed application instance is returned to a consistent state. If the node that failed is still capable of issuing I/O requests to virtual shared disks, those requests must not be carried out. An application's recovery script can use the **fencevsd** command to prevent I/O operations from failed nodes to virtual shared disks in the system.

When the node that failed is active again, the application's recovery script can issue the **unfencevsd** command to permit it to issue I/O requests to virtual shared disks again.

You can check the fenced status of all the nodes in the system with the **lsfencevsd** command, which displays a map of all fenced virtual shared disks and the nodes that are fenced from them.

Chapter 10. What You Need to Know about How the IBM Recoverable Virtual Shared Disk Component Works

This chapter explains how the IBM Recoverable Virtual Shared Disk subsystem works. The subject is addressed in terms of the shared disk commands. Keep in mind that the IBM Virtual Shared Disk Perspective essentially executes these commands. So this discussion applies when using the IBM Virtual Shared Disk Perspective as well.

The Recovery Subsystems

The IBM Recoverable Virtual Shared Disk recovery subsystems, rvsd and hc, respond to changes in the status of the system by running recovery and notifying client applications. The subsystems operate as daemons named **rvsdd** and **hcd**. They use the utilities of the Group Services subsystem. For more information about Group Services, see *RS/6000 Cluster Technology: Group Services Programming Guide and Reference*.

The following sections describe the functions of the rvsd and hc subsystems.

The rvsd Subsystem

The rvsd subsystem controls recovery for the IBM Recoverable Virtual Shared Disk component of PSSP. It invokes the recovery scripts whenever there is a change in the group membership. The **ha.vsd** command controls the rvsd subsystem. When a node goes down or a disk adapter or cable fails, the rvsd subsystem notifies all surviving processes in the remaining virtual shared disk nodes, so they can begin recovery. If a node fails, recovery involves switching the ownership of a twin-tailed disk to the secondary node. If a disk adapter or cable fails, recovery involves switching the server node for a volume group to the secondary node. When the failed component comes back up, recovery involves switching disk or volume group ownership back to the primary node.

Communication adapter (**css** or **en**) failures are treated in the same manner as node failures. Recovery for twin-tailed volume groups consists of switching to the secondary server.

The *primary node* is a node that is physically connected to a set of virtual shared disks and will always manage them if it is active. The *secondary node* is a node that is physically connected to a set of virtual shared disks and will manage them only if the primary node becomes inactive. Primary and secondary nodes are defined with the **createvsd** and **createhsd** commands.

A *client* is a node that has access to virtual shared disks but is not physically connected to them. Some nodes in a system partition may not have any access to virtual shared disks defined on that system and are neither servers nor clients.

The rvsd subsystem uses the notion of *quorum*, the majority of the virtual shared disk nodes, to cope with communication failures. If the nodes in a system partition are divided by a network failure, so that the nodes in one group cannot communicate with the nodes in the other group, the rvsd subsystem uses the quorum to decide which group continues operating and which group is deactivated.

In previous releases of the SP system, quorum was defined as a majority of all the nodes in a system partition. As of IBM Recoverable Virtual Shared Disk Version 1 Release 2, quorum is based on nodes that have been defined as virtual shared disk nodes. You can check the current quorum value with the **ha.vsd** query command on a node by node basis. You can override the system default value for the quorum with the **ha.vsd quorum** command.

Table 4 shows how the daemons in the nodes in a system partition react as inactive nodes come back up and as active nodes fail. The table shows the changes that affect three of the nodes in a system that has more than three nodes.

Table 4 (Page 1 of 2). Recovery Actions When Nodes Fail

Nodes and Clients	**Node Is Active**	**Recovery Scenario (Node Is Inactive)**
Node1, primary	• Daemons running on the other nodes in the system partition accept node1 into the group. • All virtual shared disks on node1 become active. • All clients designate node1 as the manager for the node1 virtual shared disks and designate those virtual shared disks as active.	• All virtual shared disks defined on node1 are put into suspended state on all clients. • Daemons running on the other nodes in the system partition remove node1 from the group. • Node2, the secondary node, takes over the management of node1's virtual shared disks. • All clients designate node2 as the server for node1's virtual shared disks and put those virtual shared disks into active state.
Node2, secondary to node1	• Daemons running on the other nodes in the system partition accept node2 into the group. • If node1 is active, there is no change to the status of the node1 virtual shared disks. • If node1 is inactive, node2 takes over the management of the node1 virtual shared disks. The node1 virtual shared disks are designated as active and managed by node2 on all clients.	• If node1 is active, there is no change to the status of its virtual shared disks. • If node1 is inactive, the node1 virtual shared disks are put into stopped state on all clients. They remain in stopped state until node1 or node2 comes back up. • Daemons running on the other nodes in the system partition remove node2 from the group.

Table 4 (Page 2 of 2). Recovery Actions When Nodes Fail

Nodes and Clients	Node Is Active	Recovery Scenario (Node Is Inactive)
Node3, client	• Daemons running on the other nodes in the system partition accept node3 into the group. • All virtual shared disks on active nodes for which node3 is a client are put into active state from node3's point of view.	• All virtual shared disks defined on node3 are put into stopped state from node3's point of view. • Daemons running on the other nodes in the system partition remove node3 from the group.

Disk Cable and Disk Adapter Failures

Hardware interruptions at the disk are known as EIO errors. IBM Recoverable Virtual Shared Disk can recover by volume group from some kinds of EIO errors, for example, disk cable and disk adapter failures. These failures can affect some of the volume groups defined on a node without affecting other volume groups. IBM Recoverable Virtual Shared Disk switches the server function from the primary node to the secondary node for the failed volume groups on the node, without changing the server for those volume groups that have not failed.

When the cause of the failure has been located and repaired, use the **vsdchgserver** command to restore the server function to the primary setting. This does not happen automatically. To enable or disable EIO recovery, for example, use the following:

```
vsdchgserver -g -p -b -o 1|0
```

Communication Adapter Failures

Communication adapter failure is supported for SP switch and Ethernet adapters. When communication adapter recovery is enabled, an adapter failure is promoted to an IBM Recoverable Virtual Shared Disk node failure so that twin-tailed volume groups can fail over to the secondary server.

Only twin-tailed volume groups on nodes connected by SP switch and Ethernet adapters will recover from adapter failure. Other communication adapters such as FDDI and Token Ring can be used, but volume groups connected by these will not switch to the secondary server in the event of adapter failure.

Communication adapter recovery is enabled by default but can be disabled by issuing the **ha.vsd adapter_recovery off** command. Use this command when you have supported communication adapters and do not want their failures promoted to node failures.

Issue **ha.vsd query** to determine whether adapter recovery is enabled or disabled. The output will be similar to the following example, where *adapter_recovery* can be `on` or `off`, and *adapter_status* can be `up`, `down`, or `unknown`.

```
Subsystem         Group            PID     Status
rvsd             rvsd             13660   active
rvsd(vsd): quorum= 8, active=0, state=idle, isolation=member,
              NoNodes=2, lastProtocol=nodes_joining,
              adapter_recovery=on, adapter_status=down.
```

The hc Subsystem

The hc subsystem is also called the Connection Manager. It supports the development of recoverable applications. Chapter 9, “Application Programming Considerations” on page 71 provides more information on how to write recoverable applications. The hc subsystem maintains a membership list of the nodes that are currently running hc processes and an incarnation number that is changed every time the membership list changes. The hc subsystem shadows the rvsd subsystem; recording the same changes in state and management of virtual shared disks that the rvsd subsystem records. The difference is that the hc subsystem only records these changes after the rvsd subsystem processes them, to assure that the IBM Recoverable Virtual Shared Disk recovery activities begin and complete before the recovery of the hc subsystem client applications takes place. This serialization helps ensure data integrity. It also explains why the hc subsystem cannot run on a node where the rvsd subsystem is not running.

You can use the **fencevsd** command to implement application recovery procedures that are independent of IBM Recoverable Virtual Shared Disk recovery. The application can request IBM Recoverable Virtual Shared Disk to fence nodes with failing application instances from nodes where the application is running correctly. See “Preserving Data Integrity During Application Recovery” on page 75 for more information.

Another important characteristic of the hc subsystem is that it waits for its client application to connect before joining the hc group. If the client application loses its connection, the hc subsystem runs the **hc.deactivate** script and leaves the group. This means that the hc group's membership list corresponds to the list of nodes in which the client application is currently running.

Startup Time

When you start up your SP, the IBM Recoverable Virtual Shared Disk subsystem does the following:

- Configures **all** defined virtual shared disks at system startup using **cfgvsd -a**. All the virtual shared disks on the nodes are in stopped state.
- When quorum is reached, the IBM Recoverable Virtual Shared Disk subsystem activates all virtual shared disks with servers in the active group.
  - All virtual shared disks on the active nodes that also have a server in the active group will be in the active state.
  - If an virtual shared disk's primary server node is in the active group, the primary will be the server for that virtual shared disk.
  - If an virtual shared disk's primary server is not in the active group but the secondary server is, the secondary will be the server for that virtual shared disk.
  - All virtual shared disks without a server in the active group will be in the stopped state.
- If quorum is lost, all the virtual shared disks are put into the stopped state. When quorum is active again, the virtual shared disks are put into appropriate states based on the list above.

The IBM Recoverable Virtual Shared Disk subsystem actively manages all configured virtual shared disks. The recovery scripts invoke the **preparevsd**, **resumevsd**, **suspendvsd**, **stopvsd**, **varyonvg**, and **varyoffvg** commands asynchronously and automatically in the background as virtual shared disk server nodes become active or fail.

If you have any script that issues any virtual shared disk command, you should disable it so it does not interfere with IBM Recoverable Virtual Shared Disk recovery. Do not issue any commands that change the state of virtual shared disks.

The commands you must not issue when the rvsd subsystem is running are:

cfgvsd
cfghsdvsd
preparevsd
resumevsd
startvsd
stopvsd
suspendvsd

Notes:

1. In order for the IBM Recoverable Virtual Shared Disk subsystem to be activated, there must be a quorum.
2. After you install your system, you can start IBM Virtual Shared Disk and IBM Recoverable Virtual Shared Disk with the **ha_vsd reset** command. See the **ha_vsd** reference page in *PSSP: Command and Technical Reference*, for the command syntax.

IBM Hashed Shared Disk Recovery Considerations

When a virtual shared disk's primary server node fails and the secondary node takes over, the takeover is transparent to the Hashed Shared Disk subsystem. The Hashed Shared Disk subsystem works with the virtual shared disks and does not know where the virtual shared disk servers are.

If the hashed shared disk device driver is working with the IBM Recoverable Virtual Shared Disk subsystem, the hashed shared disks are recovered too.

Chapter 11. Recovery Scenarios

The recovery scenarios describe what an administrator or operator might see when the IBM Recoverable Virtual Shared Disk subsystem goes into action to recover from system problems.

Recognizing Recovery

You know that the IBM Recoverable Virtual Shared Disk subsystem is performing recovery processing when you see virtual shared disks that are in the active or in the suspended state and you did not put them there. Use the IBM Virtual Shared Disk Perspective graphical user interface to display the states of your virtual shared disk nodes. See “Monitoring Virtual Shared Disks” on page 49.

If you recognize that recovery is not taking place normally, check if the IBM Recoverable Virtual Shared Disk subsystem is active on all nodes. Use the IBM Virtual Shared Disk Perspective graphical user interface or the **ha.vsd query** and **hc.vsd query** commands to see if the respective subsystems are active. If **active=0** or **state=idle** for an extended period of time, then recovery is not taking place normally. The **ha.vsd query** command returns output in the following format:

```
Subsystem         Group            PID     Status
rvsd              rvsd             18320   active
rvsd(vsd): quorum= 7, active=0, state=idle, isolation=member,
           NoNodes=5, lastProtocol=nodes_failing,
           adapter_recovery=on, adapter_status=up,
           RefreshProtocal has never been issued from this node,
           Running function level 3.1.0.0.
```

The **hc.vsd query** command output looks like this:

```
Subsystem         Group            PID     Status
hc.hc              rvsd             20440   active
 hc(hc): active=0, state=waiting for client to connect
 PING_DELAY=600
 CLIENT_PATH=/tmp/serv.
 SCRIPT_PATH=/usr/lpp/csd/bin
```

Planning for Recovery

You should have disabled or removed all user-provided scripts that issue change-of-state commands to virtual shared disks.

Monitor the activity of the IBM Recoverable Virtual Shared Disk subsystem using the IBM Virtual Shared Disk Perspective graphical user interface to become aware of potential problems earlier. You can begin a monitoring session, leave it in a window on your workstation and go about doing other work while the monitoring activity continues.

Each of the recovery scenarios is organized as follows:

1. Symptoms
2. Detection
3. Affected components
4. Recovery steps

5. Restart

Virtual Shared Disk Node Failure

- **Symptoms**

  A node has either hung or has failed.

- **Detection**

  Node failure was detected by the Group Services program and it has notified the IBM Recoverable Virtual Shared Disk subsystem. An operator may have seen the change in state displayed by the IBM Virtual Shared Disk Perspective graphical user interface.

- **Affected Components**

  The affected components can be everything accessing the virtual shared disks, including: the virtual shared disks themselves, software applications, and the IBM Recoverable Virtual Shared Disk and related subsystems running on the failed nodes.

- **Recovery Steps**

  The recovery steps are:

  1. The recovery services of surviving nodes suspend virtual shared disks served by the failing node. The virtual shared disk recovery process puts the suspended virtual shared disks into the active state on the secondary node.
  2. Optional subscribers to **hc** on surviving nodes are informed of the membership change.

- **Restarting the Failed Node**

  Some of the following actions must be done manually; others are done automatically by the IBM Recoverable Virtual Shared Disk subsystem. You should have procedures in place to instruct operators when and how to do the manual operations.

  1. An operator reboots the failed node.
  2. An operator may need to issue the **Estart** or **Eunfence** command to enable the rebooted node to access the switch. Nodes can be set up to reboot automatically, using the **Estart -M** command, which starts the monitor function.
  3. The IBM Recoverable Virtual Shared Disk subsystem is automatically brought up on reboot, once the communications adapter is available.
  4. The **hc.activate** script, if present, is invoked.
  5. The rebooted node rejoins the active group. If it was a primary node, it takes over again as a primary node.
  6. Optional subscribers to **hc** on surviving nodes are informed of the membership change.

Switch Failure Scenarios

The sequence of events triggered by switch failure depends on whether adapter recovery is enabled.

When Adapter Recovery is Enabled

- **Symptoms**

  The IBM Recoverable Virtual Shared Disk subsystem initiates recovery on failing nodes. Errors might be generated by applications running on these nodes. Error reports might be generated as well.

- **Detection**

  Node failure was detected by the IBM Recoverable Virtual Shared Disk subsystem. An operator may have seen the change in state displayed by the IBM Virtual Shared Disk Perspective graphical user interface.

- **Affected Components**

  The affected components can be everything accessing the virtual shared disks, including: the virtual shared disks themselves, software applications, and the IBM Recoverable Virtual Shared Disk and related subsystems running on the failed nodes.

- **Recovery Steps and Restart**

  Follow problem determination procedures for switch failure. See PSSP: Messages Reference.

When Adapter Recovery is Disabled

- **Symptoms**

  Remote virtual shared disk I/O requests hang and then fail after about 15 minutes. The virtual shared disk clients see a time out error.

- **Detection**

  The IBM Recoverable Virtual Shared Disk subsystem neither detects nor handles switch failure when adapter recovery is disabled or when a non-supported adapter is used. An operator using the SP or IBM Virtual Shared Disk Perspective graphical user interface might recognize that **switch_responds** for the node is off.

- **Affected Components**

  Applications using virtual shared disks will hang and then fail after about 15 minutes.

- **Recovery Steps and Restart**

  An operator issues the **Estart** or **Eunfence** command to restart the switch.

  If **Estart** or **Eunfence** fails, use standard diagnostic methods for handling switch problems (see the switch error logs in *PSSP: Messages Reference*).

  The Problem Management interface could be used to run a script that would stop the IBM Recoverable Virtual Shared Disk subsystem on switch failures and restart it when the switch is active again.

Topology Services or Recovery Service Daemon Failure

- **Symptoms**
  - At the node where the failure occurred, all the virtual shared disks stop and restart, causing I/O errors to the application.
  - At other nodes, the virtual shared disks served by the problem node are switched to their secondary servers briefly, then returned to the primary server.
- **Detection**

  There is no automatic method to determine that the daemons are failing.
- **Recovery Steps**
  1. In the unlikely event that the recovery service daemons hang, no recovery is performed.
  2. At the node where the failure occurred, the virtual shared disks remain in the active state, but subsequent node failures or reboots can cause I/O requests to remote virtual shared disks to hang and fail after 15 minutes. Issue the **ps** command at that node to check for the **rvsd** and **hc** daemons.
  3. At the other nodes, some virtual shared disks remain indefinitely in the suspended state. All the **rvsd** daemons are present and their presence can be verified by using the **ps** command.
  4. When the daemons are available again, you can manually issue the **ha_vsd reset** command on the problem node. If this is insufficient, reboot the problem node.

  For more information, see *RS/6000 Cluster Technology: Group Services Programming Guide and Reference*

Hardware Failures

- **Symptoms**

  I/O errors occur on some or all of the virtual shared disks served from a node.
- **Detection**

  There is no automatic method of determining that volume group failures are occurring. A hardware error return code (EIO) is returned to the IBM Virtual Shared Disk device driver.
- **Affected Components**

  The IBM Virtual Shared Disk subsystem cannot access the data on the failed volume groups.
- **Recovery Steps**
  1. The volume group that contains the failed virtual shared disks is automatically put into the suspended state by the IBM Recoverable Virtual Shared Disk subsystem.
  2. If the secondary node is available, the volume group failure recovery procedure takes place and the secondary node begins handling requests for the virtual shared disks that share the failed volume group.

3. If the secondary node is inactive or if the secondary node was already acting as the server for this volume group, the virtual shared disks that share the failed volume group are put into the stopped state. The IBM Recoverable Virtual Shared Disk subsystem is not able to recover in this situation.
4. Correct the condition that caused the error. You can now issue the **vsdchgserver** command to change the server function back to the primary node.

Note: If you mirror data from each virtual shared disk to a virtual shared disk on another adapter, you will never need to recover from this type of error.

Chapter 12. IOCTL Subroutine

fence subroutine

Purpose

fence – Allows you to request and change the virtual shared disk fence map. The fence map controls whether virtual shared disks can send or satisfy requests from virtual shared disks at remote nodes.

Syntax

```
#include <vsd_ioctl.h>
int ioctl(FileDescriptor, Command, Argument)
int FileDescriptor, Command;
void *Argument;
```

Parameters

FileDescriptor
Specifies the open file descriptor for which the control operation is to be performed.

Command
Specifies the control function to be performed. The value of this parameter is always **GIOCFENCE**.

Argument Specifies a pointer to a **vsd_fence_map** structure.

The flags field of the **vsd_fence_map** structure determines the type of operation that is performed. The flags could be set with one or more options using the OR operator. These options are as follows:

VSD_FENCE_COMPARE
Indicates that all the bits in the **node_FenceMask** bitmap represent the current status for a given VSD minor, and the **node_FenceMask** bitmap is the new status to be applied by this request. The request will succeed only if the current status depicted in the **node_FenceMask** is valid.

VSD_FENCE_FORCE
If this option is specified, a node can unfence itself.

VSD_FENCE_GET
Denotes a query request.

VSD_FENCE_SET
Denotes a fence request.

VSD_FENCE_SWAP
Indicates that the **node_FenceMask** bitmap denotes which bits in the **node_FenceMap** the request applies to. Only nodes whose bit is turned on in the **node_FenceMask** bitmap may change status.

Description

The fence-map is an array with the following definition:

```
typedef struct
unsigned int   nvsds; /* number of elements in the vsd_Fence
                         array */
ulong          flags; /* specifies the type of request */
vsd_Fence_t    *Map;  /* array of vsd_Fence structures */
} vsd_FenceMap_t;
```

nvsds represents the number of elements in the **vsd_Fence** array. The flags specify the type of request. The map is an array of the following structure:

```
typedef struct
{
    unsigned int      vsd_minor;       /* The bitmap is associated
                                          with this minor */
    vsd_Fence_Bitmap  node_FenceMap;   /* Nodes status - fenced or
                                          unfenced */
    vsd_Fence_Bitmap  node_FenceMask;  /* Determine how node_FenceMap
                                          is used */
} vsd_Fence_t;
```

The *vsd_minor* represents the virtual shared disk device's minor number. The **node_FenceMap** bitmap denotes which nodes are fenced (bit turned on) or unfenced (bit turned off) from the virtual shared disk. You could use the **MAP_SET** and **MAP_CLR** macros defined in **vsd_ioctl.h** to turn a given bit in the map on or off. The **node_FenceMask** is used to interpret the **node_FenceMap** bitmap, depending on the flags field.

You can submit a fence map that changes the current virtual shared disk map (maintained by the IBM Virtual Shared Disk subsystem) according to the flags set in the **vsd_fence_map** structure. The request is considered a delta to the current virtual shared disk map. You can also utilize this ioctl to query the current virtual shared disk map.

Return Values

If the request succeeds, the ioctl returns 0. In the case of an error, a value of -1 is returned with the global variable **errno** set to identify the error.

Error Values

The **fence** ioctl subroutine can return the following error codes:

EACCES Indicates that an unfence was requested from a fenced node without the **VSD_FENCE_FORCE** option.

EINVAL Indicates an invalid request (ambiguous flags or unidentified virtual shared disks).

ENOCONNECT
Indicates that either the primary or the secondary node for a virtual shared disk to be fenced is not a member of the virtual shared disk group, or the virtual shared disk in question is in the **stopped** state.

ENOMEM Indicates, in the case of a query request, that the current virtual shared disk fence map is larger than the one specified in the query request. The query fails and the *nvsds* field in the request is changed to reflect the current size of the fence map.

ENOTREADY Indicates that the group is not active or IBM Recoverable Virtual Shared Disk is not available.

ENXIO Indicates that the IBM Virtual Shared Disk driver is being unloaded.

Examples

The following is an example of how to fence a virtual shared disk with a minor number of 7 from node 4 and 5, and to unfence a virtual shared disk with a minor number of 5 from node 1:

```
int fd;
vsd_FenceMap_t FenceMap;

/* two VSDs to work with */
FenceMap.Map=(vsd_Fence *) malloc (2*sizeof(vsd_Fence));
FenceMap.nvsds = 2;

/* Clear the Map and Mask*/
MAP_ZERO(FenceMap.Map[0].node_FenceMap);
MAP_ZERO(FenceMap.Map[0].node_FenceMask);
MAP_ZERO(FenceMap.Map[1].node_FenceMap);
MAP_ZERO(FenceMap.Map[1].node_FenceMask);
FenceMap.flags = VSD_FENCE_SET | VSD_FENCE_FORCE | VSD_FENCE_SWAP;

/* fence nodes 4,5 from minor 7 */
FenceMap.Map[0].vsd_minor = 7;
MAP_SET(4, FenceMap.Map[0]node_FenceMap);
MAP_SET(4, FenceMap.Map[0].node_FenceMask);
MAP_SET(5, FenceMap.Map[0].node_FenceMap);
MAP_SET(5, FenceMap.Map[0].node_FenceMask)

/* Unfence node 1 from minor 5*/
FenceMap.Map[1].vsd_minor = 5;
MAP_SET(1, FenceMap.Map[1].node_FenceMask);

/* Issue the fence request */
ioctl(fd,GIOCFENCE,&FenceMap);
```

Appendix A. Interface Cross-Reference

This appendix contains a cross-reference chart including basic tasks and how to perform them using each of the interfaces. Your interface options include the IBM Virtual Shared Disk Perspective graphical user interface, SMIT, and the commands. Most commands complete a single step and operate on a single node while others, like the **createvsd** and **createhsd** commands, can operate on multiple nodes and perform multiple steps.

This appendix also includes a summary of the virtual shared disk commands, showing the command name and its purpose. The full path for all commands involved with managing virtual shared disks is **/usr/lpp/csd/bin/** and the path for other PSSP commands is **/usr/lpp/ssp/bin/**. The syntax and complete descriptions of all commands in PSSP are in the book *PSSP: Command and Technical Reference*.

Virtual Shared Disk Tasks

Table 5 (Page 1 of 4). Interface Cross-Reference for Virtual Shared Disk Tasks

Task	IBM Virtual Shared Disk Perspective	SMIT	Command
Designate nodes as virtual shared disk nodes	• Click on icons in the Nodes pane • Click on **Actions→Designate as an IBM VSD Node...**	• **smit vsd_data** • Click on **VSD Node Information**	**vsdnode**
Define global volume groups (do only if you intend to ***define***, not ***create*** virtual shared disks)	Use **Actions→Run Command...**	• **smit vsd_data** • Click on **VSD Global Volume Group Information**	**vsdvg**
Create virtual shared disks	• Click on the IBM VSDs pane • Click on **Actions→Create...**	• **smit vsd_data** • Click on **Create a VSD**	**createvsd**
Define virtual shared disks (do only if you intend to ***define***, not ***create*** virtual shared disks)	Click on the IBM VSDs pane Click on **Actions→Define...**	• **smit vsd_data** • Click on **Define a VSD**	**defvsd**
Configure virtual shared disks (not when using the rvsd subsystem)	• Click on virtual shared disk node icons in the Nodes pane • Click on **Actions→Configure IBM VSDs...** or • Click on icons in the IBM VSDs pane • Click on **Actions→Configure...**	• **smit vsd_mgmt** • Click on **Configure a VSD**	**cfgvsd**
Activate virtual shared disks and hashed shared disks (not when using the rvsd subsystem)	• Click on virtual shared disk node icons in the Nodes pane • Click on **Actions→Change IBM VSDs State...→Active** or • Click on icons in the IBM VSDs pane • Click on **Change State→Active**	• **smit vsd_mgmt** • Click on **Start a VSD**	**startvsd** **preparevsd** **resumevsd**

Table 5 (Page 2 of 4). Interface Cross-Reference for Virtual Shared Disk Tasks

Task	IBM Virtual Shared Disk Perspective	SMIT	Command
Change the state of virtual shared disks (not when using the rvsd subsystem)	• Click on virtual shared disk node icons in the Nodes pane • Click on **Actions→Change IBM VSDs State...→Active**, **Stopped**, or **Suspended** or • Click on icons in the IBM VSDs pane • Click on **Change State→Active**, **Stopped**, or **Suspended**	• **smit vsd_mgmt** • Click on one of: **Start a VSD** **Prepare a VSD** **Resume a VSD** **Suspend a VSD** **Stop a VSD**	**startvsd** **preparevsd** **resumevsd** **suspendvsd** **stopvsd**
Display node information	Double-click on an icon in the Nodes pane or • Click on an icon in the Nodes pane • Do one of: – Click on Properties notebook icon in the tool bar – Click on **Actions→View and Modify Properties...** Look at the IBM VSD Node Attributes page. Look at the Configured IBM VSDs and HSDs page.	• **smit list_vsd** • Click on **List VSD Node Information**	**vsdatalst -n**
Display global volume group information	Use **Actions→Run Command...**	• **smit list_vsd** • Click on **List VSD Global Volume Group Information**	**vsdatalst -g**
Display virtual shared disk information	Double-click an icon in the IBM VSDs pane or • Click on an icon in the IBM VSDs pane • Do one of: – Click on Properties notebook icon in the tool bar – Click on **Actions→View and Modify Properties...** Look at the IBM VSD Attributes page.	• **smit list_vsd** • Click on **List Defined Virtual Shared Disks** or • **smit vsd_mgmt** • Click on **Show All Managed VSD Characteristics**	**vsdatalst -v** **lsvsd** **ctlvsd**
Display virtual shared disk statistics	Use **Actions→Run Command...**	• **smit vsd_mgmt** • Click on **Show All Managed VSD Statistics**	**lsvsd -s** **ctlvsd**
Display virtual shared disk device driver statistics	Double-click a virtual shared disk node icon in the Nodes pane or • Click on a virtual shared disk node icon in the Nodes pane • Do one of: – Click on Properties notebook icon in the tool bar – Click on **Actions→View and Modify Properties...** Look at the IBM VSD Node Statistics page.	• **smit vsd_mgmt** • Click on **Show VSD Device Driver Statistics**	**statvsd**
Reset virtual shared disk statistics	Use **Actions→Run Command...**	• **smit list_vsd** • Click on **Set/Show VSD Device Driver Operational Parameters**	**ctlvsd**

Table 5 (Page 3 of 4). Interface Cross-Reference for Virtual Shared Disk Tasks

Task	IBM Virtual Shared Disk Perspective	SMIT	Command
Modify node information	Double-click an icon in the Nodes pane or • Click on an icon in the Nodes pane • Do one of: – Click on Properties notebook icon in the tool bar – Click on **Actions→View and Modify Properties...** Click on the **IBM VSD Node Attributes** tab and make modifications.	• **smit list_vsd** • Click on **Set/Show VSD Device Driver Operational Parameters**	**updatevsdnode**
Modify virtual shared disk information	Use **Actions→Run Command...**	• **smit list_vsd** • Click on **Set/Show VSD Device Driver Operational Parameters**	**updatevsdtab**
Modify virtual shared disk owner and group	• Click on icons in the IBM VSDs pane • Click on **Actions→Change Owner and Group...**	Not supported	AIX commands: **chown** **chgrp**
Verify you can write to a virtual shared disk	Use **Actions→Run Command...**	Not supported	**vsdvts**
Note: Use ***extreme caution*** with the vsdts command because all data on the virtual shared disk and the underlying logical volume is destroyed.			
Run virtual shared disk diagnostics	• Click on the Nodes pane • Click on one of: – The Run Diagnostics icon in the tool bar – **Actions→Display IBM VSD Diagnostics...**	• **smit vsd_mgmt** • Click on **VSD Level One Diagnostics**	**vsddiag**
Display virtual shared disk resource information	Double-click a virtual shared disk node icon in the Nodes pane or • Click on a virtual shared disk node icon in the Nodes pane • Do one of: – Click on Properties notebook icon in the tool bar – Click on **Actions→View and Modify Properties...** Look at the IBM VSD Node Statistics page of the notebook.	• **smit vsd_mgmt** • Click on **Show Hard Disk and VSD Logical Volume Information**	**vsdsklst**
Enable, disable, or list virtual shared disks to be monitored	Use **Actions→Run Command...**	Not supported	**monitorvsd**
Unconfigure virtual shared disks (not while the rvsd subsystem is active)	• Click on virtual shared disk node icons in the Nodes pane • Click on **Actions→Unconfigure IBM VSDs...** or • Click on icons in the IBM VSDs pane • Click on **Actions→Unconfigure...**	• **smit vsd_mgmt** • Click on **Unconfigure a VSD**	**ucfgvsd**
Remove virtual shared disks (not while the rvsd subsystem is active)	• Click on icons in the IBM VSDs pane • Click on **Actions→Remove...**	• **smit delete_vsd** • Click on **Remove a VSD**	**removevsd**

Table 5 (Page 4 of 4). Interface Cross-Reference for Virtual Shared Disk Tasks

Task	IBM Virtual Shared Disk Perspective	SMIT	Command
Undefine virtual shared disks (not while the rvsd subsystem is active)	• Click on icons in the IBM VSDs pane • Click on **Actions→Undefine...**	• **smit delete_vsd** • Click on **Undefine a VSD**	**undefvsd**
Delete global volume groups	Use **Actions→Run Command...**	• **smit delete_vsd** • Click on **Delete VSD Global Volume Group Information**	**vsdelvg**

Hashed Shared Disk Tasks

Table 6 (Page 1 of 2). Interface Cross-Reference for Hashed Shared Disk Tasks

Task	Hashed Shared Disk Perspective	SMIT	Command
Create hashed shared disks	• Click on the IBM HSDs pane • Click on **Actions→Create...**	• **smit vsd_data** • Click on **Create an HSD**	**createhsd**
Define hashed shared disks (do only if you intend to ***define***, not ***create*** hashed shared disks)	Click on the IBM HSDs pane Click on **Actions→Define...**	• **smit vsd_data** • Click on **Define an HSD**	**defhsd**
Configure hashed shared disks (not while the rvsd subsystem is active)	• Click on virtual shared disk node icons in the Nodes pane • Click on **Actions→Configure IBM HSDs...** or • Click on icons in the IBM HSDs pane • Click on **Actions→Configure...**	• **smit hsd_mgmt** • Click on **Configure an HSD**	**cfghsd**
Display hashed shared disk information	Double-click an icon in the IBM HSDs pane or • Click on an icon in the IBM HSDs pane • Do one of: – Click on Properties notebook icon in the tool bar – Click on **Actions→View and Modify Properties...** Look at the IBM HSD Attributes page.	• **smit list_vsd** • Click on **List Defined Hashed Shared Disks**	**lshsd -l**
Display hashed shared disk statistics	Double-click a virtual shared disk node icon in the Nodes pane or • Click on a virtual shared disk node icon in the Nodes pane • Do one of: – Click on Properties notebook icon in the tool bar – Click on **Actions→View and Modify Properties...** Look at the IBM HSD Node Statistics page.	• **smit vsd_mgmt** • Click on **Show All Managed VSD Statistics**	**ctlhsd** **hsdatalst** **lshsd -s**

Table 6 (Page 2 of 2). Interface Cross-Reference for Hashed Shared Disk Tasks

Task	Hashed Shared Disk Perspective	SMIT	Command
Unconfigure HSDs (not while the rvsd subsystem is active)	• Click on virtual shared disk node icons in the Nodes pane • Click on **Actions→Unconfigure IBM HSDs...** or • Click on icons in the IBM HSDs pane • Click on **Actions→Unconfigure...**	• **smit hsd_mgmt** • Click on **Unconfigure an HSD**	**ucfghsd** **ucfghsdvsd**
Remove HSDs (not while the rvsd subsystem is active)	• Click on icons in the IBM HSDs pane • Click on **Actions→Remove...**	• **smit delete_vsd** • Click on **Remove an HSD**	**removehsd**
Undefine HSDs (not while the rvsd subsystem is active)	• Click on icons in the IBM HSDs pane • Click on **Actions→Undefine...**	• **smit delete_vsd** • Click on **Undefine an HSD**	**undefhsd**

Summary of Commands

Though IBM encourages you to use the IBM Virtual Shared Disk Perspective graphical user interface to relieve you of the more error prone command syntax, long command strings, and making sure to run them on all the right nodes, there are still times when you might have to use commands. For instance, you might have scripts which contain commands for managing your virtual shared disks. Also, there are some commands not directly available as an action. You can run any command from within the graphical user interface, using the **Run Command** action from the Nodes pane.

The syntax and complete descriptions of all commands in PSSP are in the book *PSSP: Command and Technical Reference*. The following summarizes the commands that are associated with managing shared disks:

Command	Purpose
cfghsd	Configures a hashed shared disk.
cfghsdvsd	Configures a hashed shared disk and the underlying virtual shared disks that comprise it and starts the virtual shared disks.
cfgvsd	Configures a virtual shared disk.
cksumvsd	Views and manipulates the IBM Virtual Shared Disk component's checksum parameters.
createhsd	Creates one hashed shared disk that encompasses two or more virtual shared disks.
createvsd	Creates a set of virtual shared disks, with their associated logical volumes, and puts information about them into the SDR.
ctlhsd	Sets the operational parameters for the Hashed Shared Disk subsystem on a node.
ctlvsd	Sets the operational parameters for the IBM Virtual Shared Disk subsystem on a node.
defhsd	Designates a node as either having or using a hashed shared disk.

defvsd Designates a node as either having or using a virtual shared disk.

fencevsd Prevents an application running on a node or group of nodes from accessing a virtual shared disk or group of virtual shared disks.

ha_vsd Starts and restarts the IBM Recoverable Virtual Shared Disk subsystem. This includes configuring virtual shared disks and hashed shared disks as well as activating the recoverability subsystem.

ha.vsd Queries and controls the activity of the rvsd subsystem of the IBM Recoverable Virtual Shared Disk component.

hc.vsd Queries and controls the hc subsystem of the IBM Recoverable Virtual Shared Disk component.

hsdatalst Displays hashed shared disk information for the virtual shared disk from the SDR.

hsdvts Verifies that a hashed shared disk for a virtual shared disk has been correctly configured and it works.

lshsd Displays configured hashed shared disk for a virtual shared disk and the characteristics.

lsvsd Display configured virtual shared disks and the characteristics.

monitorvsd Enable, disable, or list the virtual shared disks that will be monitored.

preparevsd Makes a virtual shared disk available.

removehsd Removes one or more hashed shared disks, the virtual shared disks associated with them, and the SDR information for virtual shared disks on the associated nodes.

removevsd Removes a set of virtual shared disks that are not part of any hashed shared disk.

resumevsd Activates an available virtual shared disk.

rvsdrestrict Displays and sets which level of the IBM Recoverable Virtual Shared Disk software is to run when you have a system partition with mixed levels of the PSSP or IBM Recoverable Virtual Shared Disk software.

spvsd Directly launches the IBM Virtual Shared Disk Perspective graphical user interface.

startvsd Makes a virtual shared disk available and activates it.

stopvsd Makes a virtual shared disk unavailable.

suspendvsd Deactivates an available virtual shared disk.

ucfghsd Makes a hashed shared disk unavailable.

ucfghsdvsd Stops the virtual shared disks that comprise a hashed shared disk and makes the hashed shared disk and the virtual shared disks unavailable.

ucfgvsd Makes a virtual shared disk unavailable.

undefhsd Undefines a hashed shared disk.

undefvsd Undefines a virtual shared disk.

unfencevsd Gives applications running on a node or group of nodes access to a virtual shared disk or group of virtual shared disks that were previously fenced from applications running on those nodes.

updatehsd Lets you change the option in the SDR that prevents overwriting the Logical Volume Control Block (LVCB) for specified hashed shared disks.

updatevsdnode Changes the IBM Virtual Shared Disk subsystem options in the SDR.

updatevsdtab Changes the IBM Virtual Shared Disk subsystem option to set cache or nocache in the SDR.

verparvsd Verifies IBM Virtual Shared Disk system partitioning.

vsdatalst Displays IBM Virtual Shared Disk system definition data from the SDR.

vsdchgserver Switches the server function for one or more virtual shared disks from the node that is currently acting as the server node to the other.

vsddiag Displays information about the status of virtual shared disks.

vsdelnode Removes IBM Virtual Shared Disk information for a node or series of nodes from the SDR.

vsdelvg Removes IBM Virtual Shared Disk global volume group information from the SDR.

vsdnode Enters IBM Virtual Shared Disk information for a node or a series of nodes into the SDR.

vsdsklst Produces output that shows you the disk resources used by the IBM Virtual Shared Disk subsystem across a system or system partition.

vsdvg Defines a virtual shared disk global volume group.

vsdvgts Reads the timestamp from the volume group descriptor area (VGDA) of the physical disks and sets the value in the SDR.

vsdvts Verifies that the IBM Virtual Shared Disk component works.

Note: The full path for all commands involved with managing virtual shared disks is **/usr/lpp/csd/bin/** and the path for other PSSP commands is **/usr/lpp/ssp/bin/**. Appendix B, “Single-Node Command and SMIT Interfaces” on page 101 shows tasks that use some of these commands in a strategy of acting on one node at a time.

Appendix B. Single-Node Command and SMIT Interfaces

Since PSSP 2.2, commands that operate on multiple nodes (such as **createvsd**, **createhsd**) and the IBM Virtual Shared Disk Perspective graphical user interface have been introduced to improve the usability of the shared disk management components. However, you might still have scripts that use single-node commands, such as **defvsd** and **defhsd**. This appendix contains the earlier version of the information on defining and managing shared disks one node at a time.

Command and SMIT Interfaces for Virtual Shared Disks

Entering Node Information in the SDR for One Node

You can use SMIT or the **vsdnode** command to enter virtual shared disk node information into the SDR.

To enter virtual shared disk node information using SMIT:

SELECT VSD Node Information

> The VSD Node Information dialog appears.

See the book *PSSP: Command and Technical Reference* for the command syntax.

Entering Global Volume Group Information in the SDR for One Node

Global volume group information is stored in the System Data Repository (SDR) **VSD_Global_Volume_Group** object. For each volume group on which virtual shared disks will be defined, the following data is required:

1. Local volume group name

   The LVM volume group's name on the serving node. The length of the name must be less than or equal to 15 characters.

2. Primary server node for the volume group

   The node number of the server node. The primary server node may be identified in four ways. See the **vsdvg** command reference page for specific information.

   If you do not have the IBM Recoverable Virtual Shared Disk software installed, all server nodes are primary nodes.

3. Secondary server node for the volume group

   The secondary server node is only intended for use with the IBM Recoverable Virtual Shared Disk software.

   The node number of the serving node. The secondary server node may be identified in four ways. See the **vsdvg** command reference page for specific information.

4. Global volume group name (must be unique across a system partition)

   The global volume group name is the globally unique name for this virtual shared disk volume group. The global volume group name is usually identical to the volume group name. The length of the name must be less than or equal to 31 characters. We recommend that the global volume group names be unique across the physical SP system. It *must* be unique within a system partition.

To enter virtual shared disk global volume group information using SMIT:

SELECT VSD Global Volume Group Information

> The VSD Global Volume Group Database Information dialog appears.

You can also enter virtual shared disk volume group information using the **vsdvg** command:

```
vsdvg [-g global_group_name ] local_group_name
primary_server_node [secondary_server_node]
```

Entering Virtual Shared Disk Information in the SDR for One Node

A virtual shared disk is defined by creating a **VSD_Table** object in the SDR. For each virtual shared disk, the following data is required:

1. virtual shared disk name (this name must be unique across the system partition)

   The globally unique name of the virtual shared disk. The name must be less than or equal to 31 characters. We recommend that the name be unique across the physical SP system.

   If you choose a *vsd_name* that is already the name of another device, the **cfgvsd** command will reject the name to insure that the special device files created for the name do not overlay and destroy files of the same name representing another device type (such as a logical volume).

2. Logical volume name

   The name of the logical volume on the serving node. The length of the name must be less than or equal to 15 characters.

3. Global volume group name

   The globally unique name of the volume group on which the virtual shared disk resides. This field is used to access the **VSD_Global_Volume_Group** to determine the node and the volume group that contains the underlying logical volume. The length of the name must be less than or equal to 31 characters.

4. Cache or nocache option

   Use the cache option only if your application does I/O in 4K blocks aligned on 4K disk boundaries, and issues a read immediately following a write.

   Nocache is the default for new virtual shared disks created with the **defvsd** command.

5. Minor number

   This number is automatically allocated.

To enter virtual shared disk information using SMIT:

SELECT Define a Virtual Shared Disk

> The Define A Virtual Shared Disk dialog appears.

You can also enter virtual shared disk information using the **defvsd** command:

```
defvsd logical_volume_name global_group_name  vsd_name [nocache
| cache ]
```

An Example of Defining a Virtual Shared Disk on One Node at a Time

Figure 17 is a simple example of a virtual shared disk. There are three virtual shared disks, one on each node. We have two physical hard disks per node and we defined a logical volume on one disk per node.

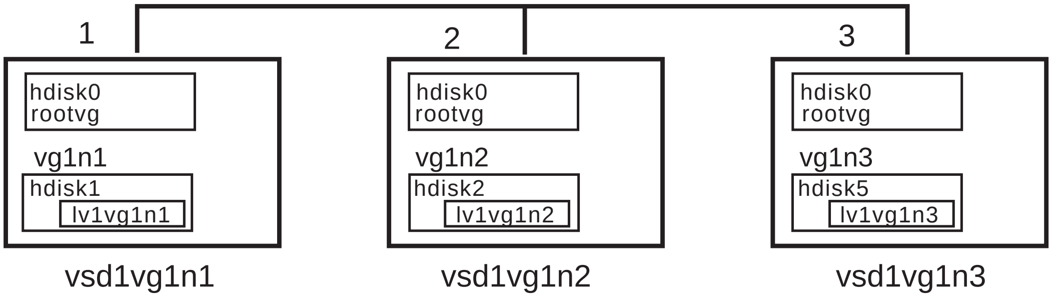


Figure 17. An Example of a Virtual Shared Disk Configuration

Here's how we did it. First, we decided that three virtual shared disks were needed, one on each node on a separate volume group. We created a table of the information for our configuration.

Table 7. Virtual Shared Disk Information

Node Number	hdisks	Volume Groups	Logical Volumes	vsd Name
1	hdisk0			
	hdisk1	vg1n1	lv1vg1n1	vsd1vg1n1
2	hdisk0			
	hdisk2	vg1n2	lv1vg1n2	vsd1vg1n2
3	hdisk0			
	hdisk5	vg1n3	lv1vg1n3	vsdvg1n3

Create Volume Groups and Logical Volumes on One Node

You must create the volume groups and logical volumes before defining the virtual shared disk information. We created the volume groups (**vg1n1**, **vg1n2**, **vg1n3**) on each node for each virtual shared disk using SMIT:

```
smit mkvg
```

and entered the volume group information requested. We took the defaults. See *AIX System Management Guide: Operating Systems and Devices*, for more information.

Next we created the logical volume groups (**lv1vg1n1**, **lv1vg1n2**, **lv1vg1n3**) on each node using SMIT:

```
smit mklv
```

and entered the global volume group information. See *AIX System Management Guide: Operating Systems and Devices*, for more information on tuning logical volumes.

Next we used SMIT to check the volume group and logical volume information we created:

```
smit lsvg
```

smit lslv

Define Virtual Shared Disks

Now we start to define our virtual shared disks. We entered the virtual shared disk node information using SMIT. This information was entered separately for node **1**, **2**, and **3**. See "Designating Nodes as IBM VSD Nodes" on page 31 for more information.

```
smit vsd_data
```

and selected the VSD Node Information option.

In the table *italics* show the information we entered. A `constant width` font shows the default data. We took most of the defaults except where noted.

Table 8. Virtual Shared Disk Node Information

Node Number	Adapter	Initial Cache	Max Cache	Request Blocks	pbufs	Min Buddy Buffer	Max Buddy Buffer	# Buddy Buffers
1	*css0*	`64`	`256`	`256`	`48`	`4096`	`24576`	`4`
2	*css0*	`64`	`256`	`256`	`48`	`4096`	`24576`	`4`
3	*css0*	`64`	`256`	`256`	`48`	`4096`	`24576`	`4`

Next we used SMIT to enter the virtual shared disk global volume group information for **vg1n1**, **vg1n2** and **vg1n3**. See "Entering Global Volume Group Information in the SDR for One Node" on page 101 for more information.

```
smit vsd_data
```

and selected the VSD Global Volume Information option.

Table 9. Virtual Shared Disk Volume Group Information

Global Group Name	Local VG Name	Primary Server	Secondary Server
vg1n1	*vg1n1*	*1*	-
vg1n2	*vg1n2*	*2*	-
vg1n3	*vg1n3*	*3*	-

We did not enter information about secondary servers because that is only for users of the IBM Recoverable Virtual Shared Disk software.

Next we entered the virtual shared disk definition information for **vsd1vg1n1**, **vsd1vg1n2** and **vsd1vg1n3**. See "Entering Virtual Shared Disk Information in the SDR for One Node" on page 102 for more information.

Again we used the SMIT command:

```
smit vsd_data
```

and selected the Define a Virtual Shared Disk option.

Table 10. Virtual Shared Disk Definition Information

LV Name	Global Group	Name	Option
lv1vg1n1	*vg1n1*	*vsd1vg1n1*	`nocache`
lv1vg1n2	*vg1n2*	*vsd1vg1n2*	`nocache`
lv1vg1n3	*vg1n3*	*vsd1vg1n3*	`nocache`

Now we have defined our virtual shared disk in the SDR.

- We checked all the data base information using SMIT:

  ```
  smit
  list_vsd
  ```

- We did the following steps next to get our system running. They are described in "Managing Virtual Shared Disks" on page 107.

  1. Configuring on "Configuring a Virtual Shared Disk" on page 109.
  2. Starting on "Starting a Virtual Shared Disk" on page 109.

Displaying Virtual Shared Disk Information Stored in the SDR

Once you have defined virtual shared disk information in the SDR, you can display this information.

To list virtual shared disk definition information using SMIT (from the SP Configuration Database Management menu):

SELECT List Database Information

> The List Database Information menu appears.

SELECT List VSD Database Information

> The List VSD Database Information menu appears.

Or you can use the fastpath invocation for this menu:

```
smit list_vsd
```

At this point, you can select options for listing node, global volume group, and defined virtual shared disk information.

You can also list virtual shared disk information using the **vsdsklst** command with the appropriate flags to indicate what kind of information you want to display. The following example shows the format of the information returned by **vsdsklst** for one node from the `vsdsklst -dv -a` command.

```
k7n12.ppd.pok.ibm.com
Node Number:12; Node Name:k7n12.ppd.pok.ibm.com
    Volume group:rootvg; Partition Size:4; Total:537; Free:315
        Physical Disk:hdisk0; Total:537; Free:315
    Volume group:vsdvg; Partition Size:4; Total:537; Free:533
        Physical Disk:hdisk1; Total:537; Free:533
        VSD Name:1HsD8n12{lv1HsD8n12}; Size:2
        VSD Name:1HsD20n12{lv1HsD20n12}; Size:2
```

Listing Node Information

To view virtual shared disk information using SMIT:

SELECT List VSD Node Information

> A listing of VSD node information appears.

You can also display virtual shared disk node information using the **vsdatalst** command:

```
vsdatalst -n
```

Listing Global Volume Group Information

To view virtual shared disk global volume group information using SMIT:

SELECT List VSD Global Volume Group Information

> A listing of VSD global volume group information appears.

You can also display virtual shared disk node information using the **vsdatalst** command:

```
vsdatalst -g
```

Listing Defined Virtual Shared Disks

To view defined virtual shared disk information using SMIT:

SELECT List Defined Virtual Shared Disks

> A listing of defined virtual shared disks appears.

You can also display defined virtual shared disk information using the **vsdatalst** command:

```
vsdatalst -v
```

Deleting Virtual Shared Disk Information from the SDR

You can delete the virtual shared disk information you have stored in the SDR. Once you have defined virtual shared disk information in the SDR, you may want to delete or change (delete and add) information. To delete virtual shared disk definition information using SMIT (from the SP Configuration Database Management menu):

SELECT Delete Database Information

> The Delete Database Information menu appears.

SELECT Delete VSD Database Information

> The Delete VSD Database Information menu appears.

The fastpath invocation for this menu is:

```
smit delete_vsd
```

At this point, you can select options for deleting node and global volume group information and for undefining a virtual shared disk.

You can also use the **removevsd** or **undefvsd** commands to remove a virtual shared disk, a list of virtual shared disks, or all the virtual shared disks in a system or system partition. See the command reference pages for **removevsd** and **undefvsd** for more information.

Managing Virtual Shared Disks

Once virtual shared disks have been defined in the SDR, they can be configured to the system and managed to various states by using SMIT menus or commands.

Running any of the following commands affects the virtual shared disk states only on the node on which the command is run:

- **varyonvg**
- **cfgvsd**
- **startvsd**
- **preparevsd**
- **resumevsd**
- **suspendvsd**
- **stopvsd**
- **varyoffvg**
- **ucfgvsd**

The SMIT panels use **dsh** to run the command on any number of nodes you select. The SMIT menus may be invoked from the control workstation. See Figure 18 on page 108 for an overview of virtual shared disk states and associated commands.

In order for an application to be able to read or write **vsd.***nnn*, **vsd.***nnn* must be in the active state on both the client and server. Both **cfgvsd** and **startvsd** must be run on **vsd.***nnn*. Once **vsd.***nnn* is in the active state, the application may open **/dev/rvsd.***nnn*, providing the permissions on the file permit access. For example, to allow an application on node **1** to access **/dev/rvsd1vg1n1**, **cfgvsd** and **startvsd** must have been run on node **1** for **vsd1vg1n1**. To allow access to **vsd1vg1n1** from node **2**, **cfgvsd** and **startvsd** must also be run on node **2**. You may use **cfgvsdhsd** on all nodes instead of **cfgvsd** on each node.

Figure 18 on page 108 summarizes virtual shared disk states, how I/O requests are treated in each state, and the commands (sometimes called methods) used to change states. New users should study Figure 18 on page 108 to understand virtual shared disk operation.

If you use the IBM Recoverable Virtual Shared Disk software, do not issue any commands that change the state of the virtual shared disks (**cfgvsd**, **statvsd**, **stopvsd**, or **ucfgvsd**.) The IBM Recoverable Virtual Shared Disk software issues these commands for you.

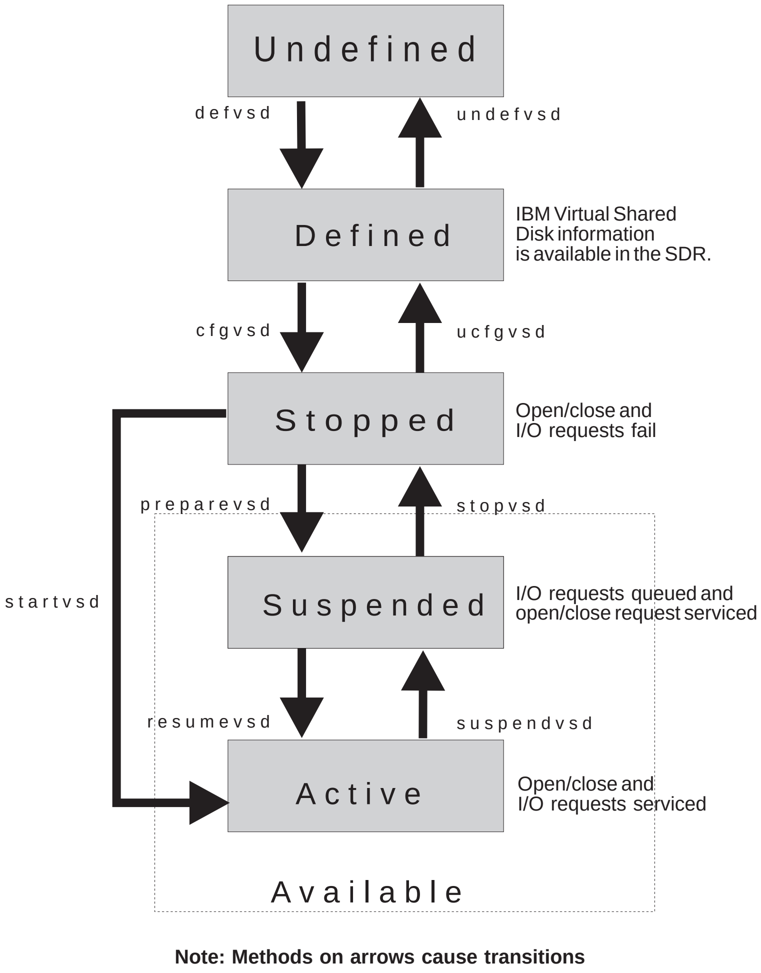

Undefined
defvsd
undefvsd
Defined
IBM Virtual Shared Disk information is available in the SDR.
cfgvsd
ucfgvsd
Stopped
Open/close and I/O requests fail
preparevsd
stopvsd
startvsd
Suspended
I/O requests queued and open/close request serviced
resumevsd
suspendvsd
Active
Open/close and I/O requests serviced
Available
Note: Methods on arrows cause transitions

Figure 18. Virtual Shared Disk States and Associated Commands

To manage virtual shared disks using SMIT:

TYPE **smit**

> The System Management menu appears

SELECT SP System Management

> The SP System Management menu appears.

SELECT SP Cluster Management

> The SP Cluster Management menu appears.

SELECT VSD Management

> The VSD Management menu appears.

The fastpath invocation for this menu is:

```
smit vsd_mgmt
```

Note: If you are not using the IBM Recoverable Virtual Shared Disk software and you want to bring up virtual shared disks automatically at boot time, all **cfgvsd** and **startvsd** (or **preparevsd** and **resumevsd**) commands can be run at boot time on all virtual shared disk nodes. Put these commands in a script that is invoked by either **/etc/init.rc** or **/etc/inittab**. This approach allows all virtual shared disk configuration to occur in parallel across the system partition as each machine boots up. You must ensure that each global volume group is varied on (using the **varyonvg** command) on the server node before running **startvsd** or **resumevsd** on the server node. If you subsequently install the IBM Recoverable Virtual Shared Disk software, make sure to undo these steps.

Configuring a Virtual Shared Disk

Configuring puts a defined virtual shared disk in the stopped state, but does not make it available. The virtual shared disk configuration method (**cfgvsd**) issues **/usr/lpp/csd/bin/readSDR** to extract information from the SDR and puts it into the flat files **VSD_Global_Volume_Group**, **VSD_Table**, **VSD_ipaddr** and **Node** in the **/usr/lpp/csd/vsdfiles** directory. Other virtual shared disk commands access these files.

Note: Do not alter these files. Each **cfgvsd** command invocation recreates the files from the SDR information and any changes you make to the files will be lost. To make changes, update the SDR information.

Use **cfgvsd** with **dsh** to configure virtual shared disks on more than one node at a time.

To configure a virtual shared disk using SMIT (from the VSD Management menu):

SELECT Configure a VSD

> The VSD and node selection lists appear.

You can also configure a virtual shared disk using the **cfgvsd** command.

To configure all virtual shared disks defined in the SDR to the node you are on, enter:

```
cfgvsd -a
```

To configure specific virtual shared disks defined in the SDR, enter:

```
cfgvsd vsd_name ...
```

Starting a Virtual Shared Disk

Starting a virtual shared disk puts a stopped one in the active (and available) state. (This is equivalent to preparing and resuming a virtual shared disk.) Note that for a virtual shared disk to be usable, it must be in the active state on both the server and client nodes.

To start a virtual shared disk using SMIT (from the VSD Management menu):

SELECT Start a VSD

> The VSD and node selection lists appear.

You can also start a virtual shared disk using the **startvsd** command.

To start all virtual shared disks defined in the SDR to the node you are on, enter:

```
startvsd -a
```

To start specific virtual shared disks defined in the SDR, enter:

```
startvsd vsd_name ...
```

Preparing a Virtual Shared Disk

Preparing a virtual shared disk puts a stopped one in the suspended state. In the suspend state, open and close requests are honored. Read and write requests are held until the virtual shared disk is brought to the active state.

To prepare a virtual shared disk using SMIT (from the VSD Management menu):

SELECT Prepare a VSD

> The VSD and node selection lists appear.

You can also prepare a virtual shared disk using the **preparevsd** command.

To prepare all virtual shared disks in the SDR, enter:

```
preparevsd -a
```

To prepare specific virtual shared disks defined in the SDR, enter:

```
preparevsd vsd_name ...
```

Resuming a Virtual Shared Disk

Resuming a virtual shared disk puts a suspended one in the active state. The virtual shared disk remains available and read and write requests that were held are resumed.

To resume a virtual shared disk using SMIT (from the VSD Management menu):

SELECT Resume a VSD

> The VSD and node selection lists appear.

You can also resume a virtual shared disk using the **resumevsd** command.

To resume all virtual shared disks defined in the SDR, enter:

```
resumevsd -a
```

To resume a specific virtual shared disk defined in the SDR, enter:

```
resumevsd vsd_name ...
```

Suspending a Virtual Shared Disk

Suspending a virtual shared disk puts an active one in the suspended state. The virtual shared disk remains available and read and write requests that were active are suspended and held. All read and write requests subsequent to those that were active are also held.

To suspend a virtual shared disk using SMIT (from the VSD Management menu):

SELECT Suspend a VSD

> The VSD and node selection lists appear.

You can also suspend a virtual shared disk using the **suspendvsd** command.

To suspend all virtual shared disks defined in the SDR, enter:

```
suspendvsd -a
```

To suspend specific virtual shared disks defined in the SDR, enter:

```
suspendvsd vsd_name ...
```

Stopping a Virtual Shared Disk

Stopping a virtual shared disk puts a suspended one in the stopped state. The virtual shared disk becomes unavailable. All applications that have outstanding requests to a stopped virtual shared disk terminate in error.

To stop a virtual shared disk using SMIT (from the VSD Management menu):

SELECT Stop a VSD

> The VSD and node selection lists appear.

You can also stop a virtual shared disk using the **stopvsd** command.

To stop all virtual shared disks defined in the SDR, enter:

```
stopvsd -a
```

To stop specific virtual shared disks defined in the SDR, enter:

```
stopvsd vsd_name ...
```

Unconfiguring a Virtual Shared Disk

Unconfiguring a stopped virtual shared disk makes it inaccessible. It does not, however, undefine or change the definition information for the virtual shared disk in the SDR.

To unconfigure a virtual shared disk using SMIT (from the VSD Management menu):

SELECT Unconfigure a VSD

> The VSD and node selection lists appear.

You can also unconfigure a virtual shared disk using the **ucfgvsd** command.

To unconfigure all virtual shared disks defined in the SDR, enter:

```
ucfgvsd
-a
```

To unconfigure specific virtual shared disks in the SDR, enter:

```
ucfgvsd vsd_name ...
```

Displaying Managed Virtual Shared Disk Characteristics

Configuration information for a virtual shared disk includes:

- Name
- Minor number
- State
- Current server node number
- *cache*|*nocache* option
- At the server, the major and minor number of the underlying logical volume. At other nodes, this information is not available.

To display managed virtual shared disk characteristics for all virtual shared disks using SMIT (from the VSD Management menu):

SELECT Show All Managed VSD Characteristics

> The VSD and node selection lists appear.

You can also display managed virtual shared disk characteristics using the **lsvsd** command. To display configuration information for specific virtual shared disks in your system, enter:

```
lsvsd -l vsd_name ...
```

Displaying Managed Virtual Shared Disk Statistics

Usage information includes:

- Number of local logical read and write operations
- Number of remote logical read and write operations
- Number of client logical read and write operations
- Number of physical reads and writes
- Number of cache hits for read
- Number of 512KB blocks read and written

To display managed virtual shared disk usage statistics for all virtual shared disks using SMIT (from the VSD Management menu):

SELECT Show All Managed VSD Statistics

> The VSD and node selection lists appear.

You can also display managed virtual shared disk usage statistics using the **lsvsd** command. To display usage information for virtual shared disks in your system, enter:

```
lsvsd -s vsd_name ...
```

Note: The number of blocks read and written is cumulative. To reset this count before you measure it, issue **ctlvsd -V** command .

Displaying Virtual Shared Disk Device Driver Statistics

You might want to display usage information about the virtual shared disk device driver. Usage information includes:

- Virtual shared disk parallelism value
- Nonzero sequence numbers and associated nodes
- Virtual shared disk device driver counters
- Node outcast status
- IP message size
- Timeouts
- Retries
- Cache shortages
- Buddy buffer shortages
- Indirect I/O
- Communications buffer pool shortage

To display usage information about the virtual shared disk device driver using SMIT (from the VSD Management menu):

SELECT Show VSD Device Driver Statistics

> A node selection list appears.

You can also display virtual shared disk device driver usage statistics using the **statvsd** command:

```
statvsd
```

See the book *PSSP: Command and Technical Reference* for a description of these statistics.

Setting and Displaying Virtual Shared Disk Device Driver Operational Parameters

You may wish to display and, in some cases, set virtual shared disk device driver operational parameters. Operational parameters include:

- Current cache buffer count
- Maximum cache buffer count
- Request block count
- pbuf count
- Level of virtual shared disk parallelism
- Buddy buffer configuration

To display and set virtual shared disk device driver operational parameters using SMIT (from the VSD Management menu):

SELECT Set/Show VSD Device Driver Operational Parameters

>Show VSD Device Driver Operational Parameters

>Reset Sequence Numbers

>Increase Cache Size

>Set Level of VSD Parallelism

>Reset VSD Device Driver Counters

>Reset a VSD's Statistics

>Reset All VSDs' Statistics

You can also display and set virtual shared disk device driver operational parameters using the **ctlvsd** command.

Testing to Verify a Virtual Shared Disk (the Verification Test Suite)

Warning: The **vsdvts** command writes data to the virtual shared disk. **Do not use the command after you have put real data on a virtual shared disk.** Refer to the **vsdvts** command reference page.

To test that you have successfully installed the IBM Virtual Shared Disk software and that you can successfully define and make a virtual shared disk active, and then read and write to it, follow these steps:

1. Make sure the switch or other network adapter is up and running.
2. Create a logical volume. One physical partition will be enough space for it.
3. Define the logical volume as a virtual shared disk in the SDR. Then, in the following commands, substitute the name of the virtual shared disk you defined for *vsd.name*. Commands are in **/usr/lpp/csd/bin**.
4. Configure the virtual shared disk to your system. Enter:

   ```
   cfgvsd vsd_name
   ```

5. Make the virtual shared disk available. Enter:

   preparevsd *vsd_name*

6. Make the virtual shared disk active. Enter:

   resumevsd *vsd_name*

7. Display configuration information about the virtual shared disk. Enter:

   lsvsd -l *vsd_name*

   The display should show that *vsd.LV* is in the active state.

8. Confirm that the virtual shared disk Verification Test is successful. Enter:

   vsdvts *vsd_name*

   The command should complete within 15 seconds and you should get the message:

   ```
   VSD verification test successful!
   ```

You can use the **vsdvts** command to verify all of your virtual shared disks. Use it to verify both local and remote virtual shared disks. Run the **cfgvsd** and **startvsd** commands on all client and server nodes. Use the SMIT panels to verify that the panels work.

Command and SMIT Interfaces for Hashed Shared Disks

Defining Hashed Shared Disk Device Driver Information in the SDR

The hashed shared disk must be defined before you can configure and use it. The data striping device information is stored in the System Data Repository (SDR). You can use either SMIT or a command to enter the information in the SDR.

Creating Hashed Shared Disks with SMIT

The SMIT panels refer to the data striping devices as “hsd”

To define the hashed shared disk information in the SDR using SMIT:

SELECT SP System Management

> The SP System Management menu appears.

SELECT SP Configuration Database Management

> The Configuration Database Management appears.

SELECT Enter Database information

> The Enter Database information menu appears.

SELECT virtual shared disk Database information

> The Virtual Shared Disk Database information menu appears

SELECT Define a Hashed Shared Disk

> The Define A Hashed Shared Disk dialog appears.

The fastpath invocation for this menu is:

smit vsd_data

An Example of Creating a Hashed Shared Disk

Example 1

We will define our hashed shared disk, called **hsd1**, with four virtual shared disks at the same node and a stripe size of 4096 bytes. The virtual shared disks are named vsd1vg1n1, vsd2vg1n1, vsd3vg1n1, and vsd4vg1n1.

The hashed shared disks were defined:

```
defhsd hsd1 4096 vsd1vg1n1 vsd2vg1n1 vsd3vg1n1 vsd4vg1n1
```

We have 16384 bytes (four 4KB blocks) of data to write. The first 4096 bytes are stored in vsd1vg1n1. The second 4096 bytes are stored in vsd2vg1n1, the third 4096 bytes are stored in vsd3vg1n1, and the last 4096 bytes are stored in vsd4vg1n1.

Example 2

Assume now we have two striped devices defined, **hsd1** and **hsd2**. Our stripe size is 4096.

We had identified the virtual shared disks to be included in the hashed shared disks. Assume there are four nodes in the system and two physical hard disks per node. We wanted to stripe across all the four nodes and all the disks. We defined a logical volume on the disks and nodes. Then we defined the virtual shared disks:

hsd name	**node**	**virtual shared disks in the hashed shared disk**
hsd1	node1	vsd1vg1n1
	node2	vsd1vg1n2
	node3	vsd1vg1n3
	node4	vsd1vg1n4
hsd2	node1	vsd2vg1n1
	node2	vsd2vg1n2
	node3	vsd2vg1n3
	node4	vsd2vg1n4

We defined the hashed shared disk with the following commands:

```
defhsd hsd1 4096 vsd1vg1n1 vsd1vg1n2 vsd1vg1n3 vsd1vg1n4
defhsd hsd2 4096 vsd2vg1n1 vsd2vg1n2 vsd2vg1n3 vsd2vg1n4
```

Displaying Data Striping Device Information in the SDR

To list hashed shared disk definition information in the SDR do one of the following:

Using SMIT

From the SP Configuration Database Management menu:

SELECT List Database Information

> The List Database Information menu appears.

SELECT List HSD Database Information

> The List Virtual Shared Disk Database Information menu appears.

SELECT List Defined Hashed Shared Disks

> A listing of defined Hashed Shared Disks appears.

You can use the fastpath invocation for this menu :

```
smit list_vsd
```

At this point, you can select options for listing defined hashed shared disk information.

To view defined hashed shared disk information using SMIT:

SELECT List Defined Hashed Shared Disks

> A listing of defined Hashed Shared Disks appears.

From the Command Line

You can also display defined hashed shared disk information using the **hsdatalst** command:

```
hsdatalst
```

Undefining Hashed Shared Disk Information in the SDR

To delete hashed shared disk definition information

Using SMIT

From the SP Configuration Database Management menu:

SELECT Delete Database Information

> The Delete Database Information menu appears.

SELECT Delete Virtual Shared Disk Database Information

> The Delete Virtual Shared Disk Database Information menu appears.

The fastpath invocation for this menu is:

```
smit delete_vsd
```

At this point, you can select options for undefining a hashed shared disk.

To undefine a hashed shared disk using SMIT:

SELECT Undefine a Hashed Shared Disk

> The Undefine a Hashed Shared Disk selection list appears.

You can either type in the *hsd_name* or select an *hsd_name* from the list function. The *hsd_name* must be in the unconfigured state.

From the Command Line

You can also undefine a hashed shared disk using the **removehsd** or **undefhsd** commands:

```
undefhsd -v hsd_names [-f]
```

```
removehsd -v hsd_names [-f]
```

If the *hsd_names* are configured, you can use the **-f** parameter of **removehsd** to force the system to unconfigure and remove them.

Managing Data Striping Devices

To manage hashed shared disks using SMIT:

TYPE **smit**

> The System Management menu appears

SELECT SP System Management

> The SP System Management menu appears.

SELECT SP Cluster Management

> The SP Cluster Management menu appears.

SELECT HSD Management

> The HSD Management menu appears.

The fastpath invocation for this menu is:

```
smit hsd_mgmt
```

Configuring a Hashed Shared Disk

Configuration makes a defined hashed shared disk available.

Using SMIT: From the HSD Management menu,

SELECT Configure an HSD

> The HSD and node selection lists appear.

From the Command Line

To configure specific hashed shared disks defined in the SDR on one node, enter:

```
cfghsd hsd_name ...
```

Unconfiguring a Hashed Shared Disk

Unconfiguring a hashed shared disk makes it unavailable. It does not, however, undefine or change the definition information for the hashed shared disk in the SDR. You must do this on each node.

Using SMIT: From the HSD Management menu,

SELECT Unconfigure an HSD

> The HSD and node selection lists appear.

From the Command Line

To unconfigure specific hashed shared disks in the SDR on one node, enter:

```
ucfghsd hsd_name ...
```

Displaying Hashed Shared Disk Information at a Node

You can display managed virtual shared disk and hashed shared disk characteristics for all configured hashed shared disks.

Using SMIT: From the HSD Management menu,

SELECT Show All Managed HSD Characteristics

> The HSD and node selection lists appear.

From the Command Line: You can also display managed hashed shared disk characteristics using the **lshsd** command. To display information for a specific hashed shared disk in your system, enter:

```
lshsd -l [ hsd_name ... ]
```

Configuration information displayed for a hashed shared disk includes:

Name
Minor number
Stripe size
protect_lvcb / **not_protect_lvcb** option
Underlying virtual shared disks in the hashed shared disk

You can display managed usage statistics for all configured hashed shared disks..

Using SMIT: From the HSD Management menu,

SELECT Show All Managed HSD Characteristics

> The HSD and node selection lists appear.

From the Command Line: You can also display managed hashed shared disk usage statistics using the **lshsd** command. To display usage information for a specific hashed shared disk in your system, enter:

```
lshsd -s [ hsd_name ... ]
```

Usage information displayed for a hashed shared disk includes:

- Number of reads on each underlying virtual shared disk
- Number of writes on each underlying virtual shared disk

Setting and Displaying Operational Parameters

You can display and, in some cases, set virtual shared disk parallelism level, as well as reset the device counters and statistics. Operational parameters include:

- Current level of hashed shared disk parallelism
- Number of read requests not at page boundary count
- Number of write requests not at page boundary count

Using SMIT: From the HSD Management menu,

SELECT Set/Show HSD Device Driver Operational Parameters

> Show HSD Device Driver Operational Parameters

> Set Level of HSD Parallelism

> Reset HSD Device Driver Counters

> Reset an HSD's Statistics

> Reset All HSDs' Statistics

You can also display and set hashed shared disk operational parameters using the **ctlhsd** command. Refer to the *PSSP: Command and Technical Reference* for more information.

Verifying Data Striping Device Installation

After you install the Hashed Shared Disk software and define and configure several hashed shared disks, you can use the verification test. Do this **before** you store data on these devices.

CAUTION:
The hsdvts command writes data to the configured *hsd_name*. Do not use the command after you have put real data on that *hsd_name* or the underlying virtual shared disks.

Refer to the **hsdvts** command reference page. Follow these steps:

1. Enter:

   ```
   hsdvts hsd.name
   ```

   You should get the message:

   ```
   HSD verification test successful!
   ```

You can use the **hsdvts** command to verify all of your hashed shared disks. Use it to verify both local and remote hashed shared disks. Issue the **cfghsd** command on all client and server nodes.

Management and Usage Notes

Configuring a Hashed Shared Disk

You can define, configure, unconfigure, and undefine hashed shared disks, using the **defhsd**, **cfghsd**, **ucfghsd**, and **undefhsd** commands.

Configuring a hashed shared disk creates the special device file **/dev/r***hsd_name*, which loads the hashed shared disk device driver into the kernel and makes the device available for use by user applications.

Use **cfghsdvsd** and **ucfghsdvsd** to configure and unconfigure hashed shared disk's and their underlying virtual shared disks on multiple nodes at once.

Unconfiguring a hashed shared disk makes the device unavailable to applications.

Defining a hashed shared disk creates a hashed shared disk object in the SDR.

Undefining a hashed shared disk deletes the hashed shared disk object from the SDR and removes the special device file **/dev/***hsd_name*.

The Hashed Shared Disk software provides interfaces to application programs through the **open**, **close**, **read**, **write**, and related system calls. A **read** or **write** first uses the offset in the data file to figure out which virtual shared disk the request is for. The Hashed Shared Disk software then passes the request to the IBM Virtual Shared Disk software to do the I/O. Usually the **ioctl** call is device-dependent. The **ioctl** call for the hashed shared disk provides minimum information about the device.

The hashed shared disk is a character (raw) device which must be opened as **/dev/r***hsd_name*; all the data addresses must be at 512 block boundaries.

Command and SMIT Interfaces for the IBM Recoverable Virtual Shared Disk Software

This section tells how to maintain twin-tailed Logical Volume Manager (LVM) components shared by two nodes in an SP system running the IBM Recoverable Virtual Shared Disk software. It includes procedures for volume groups, logical volumes, and physical volumes.

Defining Twin-tailed LVM Components in the Recoverable Virtual Shared Disk Environment

Some of the definitions you need to understand are:

twin-tailed volume group A volume group consisting of supported disks that are all twin-tailed to the same nodes.

twin-tailed physical volume
A disk in a twin-tailed volume group.

twin-tailed logical volume
A logical volume in a twin-tailed volume group.

Coordinating the Recoverable Virtual Shared Disk Environment

The twin-tailed components must have the same definition on both twin-tailed nodes. Any change to an LVM component **must** be reflected in the Object Data Manager (ODM) definitions on both nodes.

Follow the procedures in this chapter carefully in the order given to keep the LVM ODM definitions on both nodes synchronized.

Note: Do not reboot or issue **ha_vsd reset** until you have completed all the steps in a task.

Overview of Shared Volume Groups Administrative Tasks

The overall procedure for modifying a twin-tailed LVM component is the same for all tasks. In general, you export the volume group from a secondary node, make the change on the primary node, and then reimport the volume group on the secondary node. Specific operations, however, have unique steps. They are described in the following sections.

The table below summarizes the steps you must complete on the primary node and the secondary node to change a twin-tailed LVM component in a recoverable virtual shared disk system. Perform all the steps in the correct order so the data does not become corrupted.

Table 11. General Procedure for Changing a Twin-tailed LVM Component

Step Number	Primary Node	Secondary Node
Step 1	Complete prerequisite tasks	Complete prerequisite tasks
Step 2		Export a volume group
Step 3	Vary on a volume group	
Step 4	Make changes to the twin-tailed LVM component	
Step 5	Vary off the volume group	
Step 6		Import a volume group
Step 7		Change a volume group to remain dormant at start up
Step 8		Vary off the volume group
Step 9	Complete follow-up tasks	Complete follow-up tasks

Prerequisite Tasks

The prerequisite tasks, while not directly involved in modifying LVM components, must be completed before you begin to make the change.

Note: You must suspend, stop, and unconfigure all virtual shared disks on all nodes in the volume group to be changed before changing a twin-tailed LVM component.

The prerequisite tasks can vary for the different operations. The descriptions for each operation on the following pages have a list of specific prerequisite tasks.

Maintaining Twin-Tailed Volume Groups

Maintaining twin-tailed volume groups requires the following administrative tasks:

- "Creating a Twin-Tailed Volume Group"
- "Adding Physical Volumes To a Twin-Tailed Volume Group" on page 126
- "Adding or Removing Logical Volumes to or from a Twin-Tailed Volume Group" on page 130
- "Reducing a Twin-Tailed Volume Group" on page 132
- "Removing a Twin-Tailed Volume Group" on page 138

Creating a Twin-Tailed Volume Group

The table below summarizes the steps you must complete on the primary node and the secondary node to create a twin-tailed volume group. Perform all the steps in the correct order so the data does not become corrupted.

Note: Twin-tailed disks that have SCSI IDs must have a different ID on the secondary than the one on the primary.

Table 12. Steps to Create a Twin-tailed Volume Group

Step Number	Primary Node	Secondary Node
Step 1	Complete prerequisite tasks	Complete prerequisite tasks
Step 2	Create a twin-tailed volume group	
Step 3	Create logical volumes	
Step 4	Vary off the volume group	
Step 5		Import a volume group
Step 6		Change a volume group to remain dormant at startup
Step 7		Vary off the volume group
Step 8	Complete follow-up tasks	Complete follow-up tasks

Complete the following steps to create a twin-tailed volume group.

Step 1: Complete Prerequisite Tasks

1. The physical volumes (**hdisks**) should be installed, configured, named, and available.
2. Make sure the primary and the secondary nodes are active.

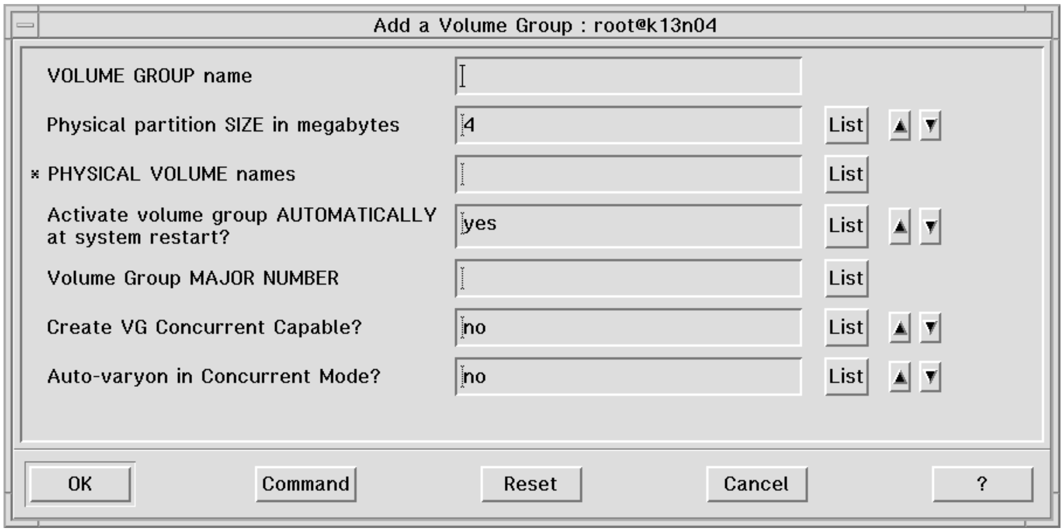
Step 2: Create a Twin-tailed Volume Group on the Primary Node

Use the **smit mkvg** fastpath to create a volume group.

1. As the root user at the primary node, enter:

   ```
   smit mkvg
   ```

   SMIT returns a screen similar to the following:



Add a Volume Group : root@k13n04

VOLUME GROUP name	
Physical partition SIZE in megabytes	4
* PHYSICAL VOLUME names	
Activate volume group AUTOMATICALLY at system restart?	yes
Volume Group MAJOR NUMBER	
Create VG Concurrent Capable?	no
Auto-varyon in Concurrent Mode?	no

2. Enter field values as follows:

   VOLUME GROUP name Enter the name of the twin-tailed volume group.

Physical partition size in megabytes
Accept the default value, unless your site has another specific partitioning requirement.

PHYSICAL VOLUME names
Enter the names of all the disks to be used in the volume group. You can click on List to display all available physical volumes. . Click on OK when you have selected all the entries you want. The physical volumes you selected are automatically entered in the PHYSICAL VOLUME names field.

Activate volume group AUTOMATICALLY at system restart?
Set the field to **no** so that the volume group can be activated as appropriate by the IBM Recoverable Virtual Shared Disk recovery scripts.

ACTIVATE volume group after it is created?
Set this field to **yes**.

Volume Group MAJOR NUMBER
Use the default, the next available number in the valid range.

Create VG Concurrent Capable?
Accept the default of **no**. Do not specify **yes**.

Auto-varyon in Concurrent Mode?
Accept the default of **no**. Do not specify **yes**.

3. Click on OK. The system asks if you are sure. Check and correct your entries, if needed.
4. Create logical volumes. Refer to the LVM reference pages.
5. After the command completes, press F12 to exit SMIT and return to the command line.

Step 3: Create Logical Volumes

Use the **smit mklv** fastpath to create logical volumes. Create the logical volumes you want for this virtual shared disk.

Step 4: Vary Off the Volume Group on the Primary Node

Use the **varyoffvg** command to quiesce the affected volume group.

To vary off the volume group so that it can be activated as appropriate by the IBM Recoverable Virtual Shared Disk recovery scripts, enter:

```
varyoffvg  volume_group_name
```

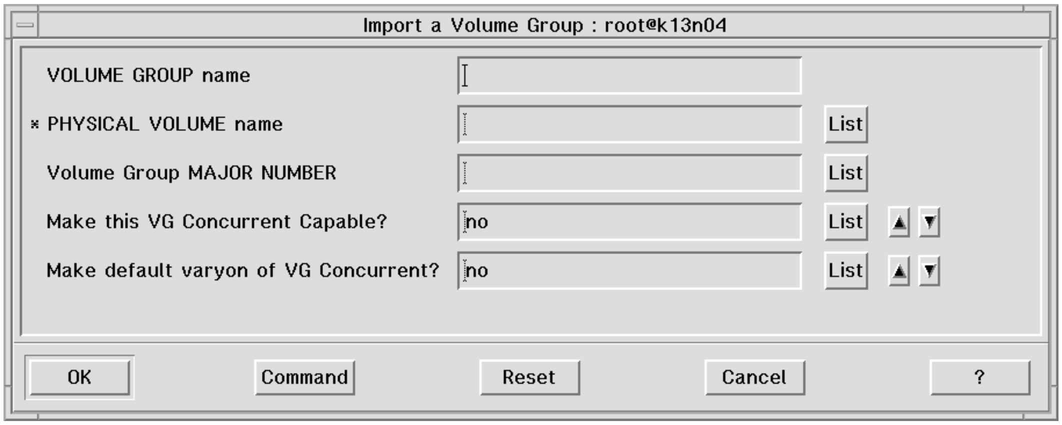
Step 5: Import Volume Group Information onto the Secondary Node

You now return to the secondary node. First, import the volume group onto the secondary node. Importing the volume group onto the secondary node synchronizes the ODM definition of the volume group on both nodes.

1. To use the **smit importvg** fastpath to import the volume group, enter:

   ```
   smit importvg
   ```

   SMIT returns a screen similar to the following:



VOLUME GROUP name

* PHYSICAL VOLUME name

Volume Group MAJOR NUMBER

Make this VG Concurrent Capable? no

Make default varyon of VG Concurrent? no

2. Enter field values as follows:

   VOLUME GROUP name Enter the name of the volume group that you are importing. Make sure the volume group name is the same name that you used on the primary node.

   PHYSICAL VOLUME name
   Enter the name of one of the physical volumes that resides in the volume group. Note that a disk can have a different physical name on different nodes. Make sure that you use the disk name as it is defined on the *secondary* node. Every **hdisk** has an ID. If you know that the name of a disk on the primary node is *hdisk01*, use the name that matches that ID on the secondary node. You can use the **lspv** command to determine the name.

   ACTIVATE volume group after it is imported?
   Set the field to **yes**.

   Volume Group MAJOR NUMBER
   Use the default, the next available number in the valid range.

   Make this VG Concurrent Capable?
   Accept the default of **no**. Do not specify **yes**.

   Make default varyon of VG Concurrent?
   Accept the default of **no**. Do not specify **yes**.

3. Click on OK.

4. Press F12 to exit SMIT and return to the command line.

Step 6: Change a Volume Group to Remain Dormant at Start Up

By default, a volume group that was just imported is configured to automatically become active at system restart. However, a recoverable virtual shared disk volume group should be varied on as appropriate by the IBM Recoverable Virtual Shared Disk recovery scripts. Therefore, after importing a volume group, use the SMIT Change a Volume Group screen to reconfigure the volume group to remain dormant at start up.

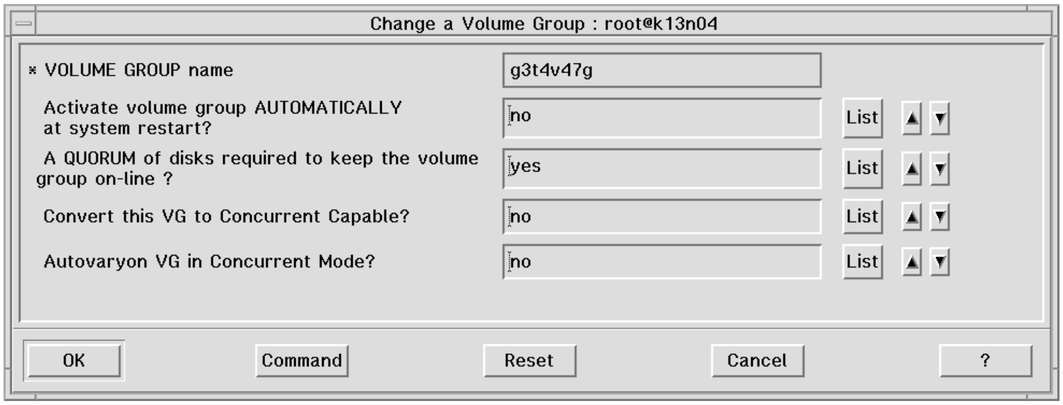
1. To use the **smit chvg** fastpath to change the characteristics of a volume group, enter:

```
smit chvg
```

SMIT prompts you to select the volume group.

2. Enter the name of the volume group you just imported, or click on List and select the name from the display. Click on OK.

   SMIT returns a screen similar to the following: The first field contains the volume group name you specified.



Change a Volume Group : root@k13n04

* VOLUME GROUP name	g3t4v47g
Activate volume group AUTOMATICALLY at system restart?	no
A QUORUM of disks required to keep the volume group on-line ?	yes
Convert this VG to Concurrent Capable?	no
Autovaryon VG in Concurrent Mode?	no

3. Enter remaining field values as follows:

   ACTIVATE volume group automatically at system restart?
   Set this field to **no**.

   A QUORUM of disks required to keep the volume group on-line?
   Accept the default, **yes**.

   Convert this VG to Concurrent Capable?
   Accept the default **no**. Do not specify **yes**.

   Autovaryon VG in Concurrent Mode?
   Accept the default **no**. Do not specify **yes**.

4. Click on OK.
5. Press F12 to exit SMIT and return to the command line.

Step 7: Vary Off the Volume Group on the Secondary Node

Use the **varyoffvg** command to quiesce the affected volume group after making the change.

To vary off the volume group so that it can be activated as appropriate by the IBM Recoverable Virtual Shared Disk recovery scripts, enter:

`varyoffvg` *`volume_group_name`*

Step 8: Complete Follow-up Tasks

Once you have created the volume group, do the following tasks:

- Make sure the primary and secondary nodes are active.
- Use the **vsdvg** command to define the twin-tailed volume groups.
- Use **defvsd** to define virtual shared disks on all the twin-tailed nodes.
- Use the **cfgvsd** command for all the virtual shared disks in this volume group at all the nodes.

- Use the **startvsd** command at all the virtual shared disk nodes to start these new virtual shared disks.

Adding Physical Volumes To a Twin-Tailed Volume Group

The following table summarizes the steps you do on both the primary and secondary nodes to extend (add one or more physical volumes to) a twin-tailed volume group. Perform all the steps in the correct order so the data does not become corrupted.

Table 13. Procedure for Extending a Twin-tailed LVM Component

Step Number	Primary Node	Secondary Node
Step 1	Complete prerequisite tasks	Complete prerequisite tasks
Step 2		Export a volume group
Step 3	Vary on a volume group	
Step 4	Extend a twin-tailed volume group	
Step 5	Vary off the volume group	
Step 6		Import a volume group
Step 7		Change a volume group's characteristics
Step 8		Vary off the volume group
Step 9	Complete follow-up tasks	Complete follow-up tasks

Step 1: Complete Prerequisite Tasks

1. The physical volumes (**hdisks**) should be installed, configured, named, and available.
2. Suspend, stop, and unconfigure all virtual shared disks in the volume group involved in the change on all nodes.

Step 2: Export Volume Group Information from the Secondary Node

Before making any changes to the LVM elements on the primary node, you must export the appropriate volume group from the secondary node. Exporting the volume group deletes the information about this volume group from the ODM.

To use the **exportvg** command to export the volume group containing the component you are going to change on the secondary node, enter:

```
exportvg volume_group_name
```

Step 3: Vary on a Volume Group on the Primary Node

To use the **varyonvg** command to activate the affected volume group after making the change, enter:

```
varyonvg  volume_group_name
```

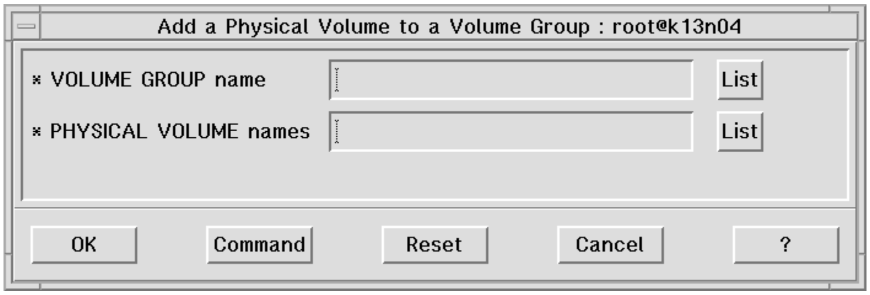
Step 4: Extend the Twin-Tailed Volume Group on the Primary Node

Use the **smit extendvg** fastpath to extend a volume group.

1. As the root user at the primary node, enter:

   ```
   smit extendvg
   ```

   SMIT returns a screen similar to the following:



Add a Physical Volume to a Volume Group : root@k13n04

* VOLUME GROUP name
* PHYSICAL VOLUME names

2. Enter field values as follows:

   VOLUME GROUP name Enter the name of the volume group that you are extending.

   PHYSICAL VOLUME names
   Enter the names of the physical volumes you are adding to the volume group.

3. Click on OK to extend the volume group.
4. After the command completes, press F12 to exit SMIT and return to the command line.
5. Create new logical volumes or extend existing ones.

Step 5: Vary Off the Volume Group on the Primary Node

Use the **varyoffvg** command to quiesce the affected volume group after making the change.

To vary off the volume group so that it can be activated as appropriate by the IBM Recoverable Virtual Shared Disk recovery scripts, enter:

```
varyoffvg volume_group_name
```

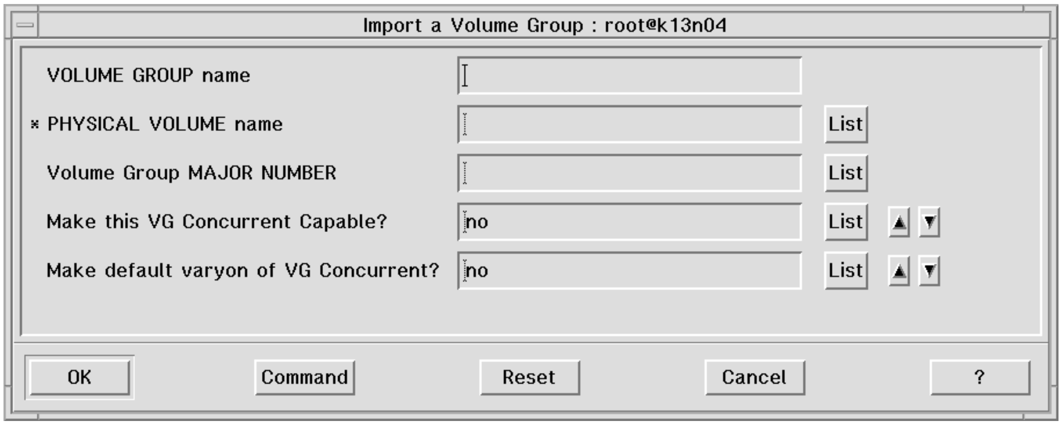
Step 6: Import Volume Group Information onto the Secondary Node

You now return to the secondary node. First, import the volume group onto the secondary node. Importing the volume group onto the secondary node synchronizes the ODM definition of the volume group with the primary node.

1. To use the **smit importvg** fastpath to import the volume group, enter:

   ```
   smit importvg
   ```

   SMIT returns a screen similar to the following:



Import a Volume Group : root@k13n04

VOLUME GROUP name

* PHYSICAL VOLUME name

Volume Group MAJOR NUMBER

Make this VG Concurrent Capable? no

Make default varyon of VG Concurrent? no

2. Enter field values as follows:

   VOLUME GROUP name Enter the name of the volume group that you are importing. Make sure the volume group name is the same name that you used on the primary node.

   PHYSICAL VOLUME name
   Enter the name of one of the physical volumes that resides in the volume group. Note that a disk can have a different physical name on different nodes. Make sure that you use the disk name as it is defined on the *secondary* node. Every **hdisk** has an ID. If you know that the name of a disk on the primary node is *hdisk01*, use the name that matches that ID on the secondary node. You can use the **lspv** command to determine the name.

   Volume Group MAJOR NUMBER
   Use the default, the next available number in the valid range.

   Make this VG Concurrent Capable?
   Accept the default of **no**. Do not specify **yes**.

   Make default varyon of VG Concurrent?
   Accept the default of **no**. Do not specify **yes**.

3. Click on OK.

4. Press F12 to exit SMIT and return to the command line.

Step 7: Change Volume Group Characteristics on the Secondary Node

By default, a volume group that was just imported is configured to automatically become active at system restart. However, a recoverable virtual shared disk volume group should be varied on as appropriate by the IBM Recoverable Virtual Shared Disk recovery scripts. Therefore, after importing a volume group, use the SMIT Change a Volume Group screen to reconfigure the volume group to remain dormant at start up.

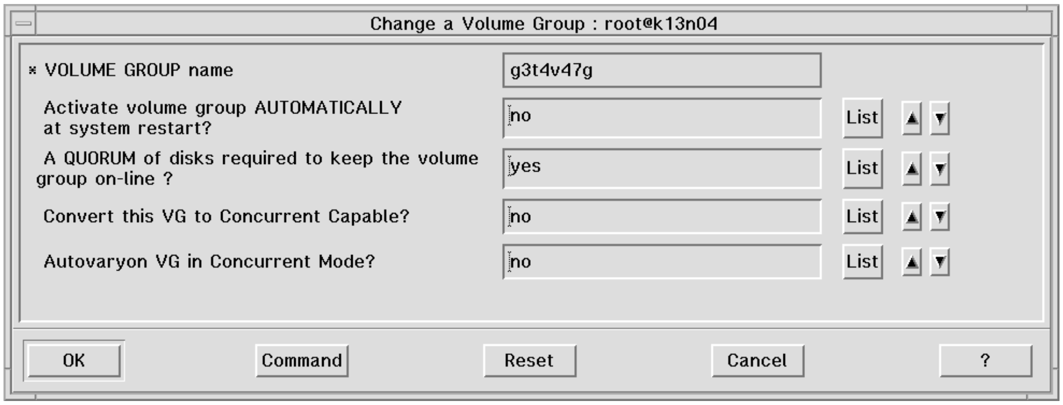
1. To use the **smit chvg** fastpath to change the characteristics of a volume group, enter:

   ```
   smit chvg
   ```

SMIT prompts you to select the volume group.

2. Enter the name of the volume group you just imported, or click on List and select the name from the display. Click on OK.

   SMIT returns a screen similar to the following. The first field contains the volume group name you specified.



Change a Volume Group : root@k13n04

* VOLUME GROUP name: g3t4v47g

Activate volume group AUTOMATICALLY at system restart?: no

A QUORUM of disks required to keep the volume group on-line ?: yes

Convert this VG to Concurrent Capable?: no

Autovaryon VG in Concurrent Mode?: no

3. Enter remaining field values as follows:

   ACTIVATE volume group automatically at system restart?
   Set this field to **no**.

   A QUORUM of disks required to keep the volume group on-line?
   Accept the default, **yes**.

   Convert this VG to Concurrent Capable?
   Accept the default **no**. Do not specify **yes**.

   Autovaryon VG in Concurrent Mode?
   Accept the default **no**. Do not specify **yes**.

4. Click on OK.

5. Press F12 to exit SMIT and return to the command line.

Step 8: Vary Off the Volume Group on the Secondary Node

Use the **varyoffvg** command to quiesce the affected volume group after making the change.

To vary off the volume group so that it can be activated as appropriate by the IBM Recoverable Virtual Shared Disk recovery scripts, enter:

```
varyoffvg volume_group_name
```

Step 9: Complete Follow-up Tasks

Once the you have extended the volume group, do the following tasks:

- Make sure the primary and secondary nodes are active.
- Use **defvsd** to define any new virtual shared disks.
- Use the **cfgvsd** command for all the virtual shared disks in this volume group at all the nodes.
- Use the **startvsd** command at all the virtual shared disk nodes to start these new virtual shared disks.

Adding or Removing Logical Volumes to or from a Twin-Tailed Volume Group

The following table summarizes the steps you do on both the primary and secondary nodes to add or remove logical volumes to or from a twin-tailed volume group. Perform all the steps in the correct order so the data does not become corrupted.

Table 14. Procedure for Extending a Twin-tailed LVM Component

Step Number	Primary Node	Secondary Node
Step 1	Complete prerequisite tasks	Complete prerequisite tasks
Step 2		Export a volume group
Step 3	Create or remove logical volumes	
Step 4	Vary off the volume group	
Step 5		Import a volume group
Step 6		Change a volume group's characteristics
Step 7		Vary off the volume group
Step 8	Vary on the volume group	
Step 9	Complete follow-up tasks	Complete follow-up tasks

Step 1: Complete Prerequisite Tasks

1. Shutdown your applications using the volume group to ensure data consistency and ODM integrity.
2. Suspend, stop, and unconfigure all virtual shared disks on all the virtual shared disk nodes.
3. Undefine virtual shared disks on any logical volumes being removed. Ensure all data is moved from the logical volumes before removing.

Step 2: Export Volume Group Information from the Secondary Node

Before making any changes to the LVM elements on the primary node, you must export the appropriate volume group from the secondary node. Exporting the volume group deletes the information about this volume group from the ODM.

To use the **exportvg** command to export the volume group containing the component you are going to change on the secondary node, enter:

```
exportvg volume_group_name
```

Step 3: Create or Remove Required Logical Volumes on the Primary Node

To create the logical volumes, use the fastpath invocation to the SMIT panel:

```
smit mklv
```

Alternatively, you can use the command line interface to create the logical volumes:

```
mklv
```

To remove the logical volumes, use the fastpath invocation to the SMIT panel:

```
smit rmlv
```

Alternatively, you can use the command line interface to remove the logical volumes:

```
rmlv
```

Step 4: Vary Off the Volume Group on the Primary Node

Use the **varyoffvg** command to quiesce the affected volume group after making the change.

To vary off the volume group so that it can be activated as appropriate by the IBM Recoverable Virtual Shared Disk recovery scripts, enter:

```
varyoffvg volume_group_name
```

Step 5: Import Volume Group Information onto the Secondary Node

You now return to the secondary node. First, import the volume group onto the secondary node. Importing the volume group onto the secondary node synchronizes the ODM definition of the volume group with the primary node.

1. To use the **smit importvg** fastpath to import the volume group, enter:

   ```
   smit importvg
   ```

Step 6: Change Volume Group Characteristics on the Secondary Node

By default, a volume group that was just imported is configured to automatically become active at system restart. However, a recoverable virtual shared disk volume group should be varied on as appropriate by the IBM Recoverable Virtual Shared Disk recovery scripts. Therefore, after importing a volume group, use the SMIT Change a Volume Group screen to reconfigure the volume group to remain dormant at start up.

To use the **smit chvg** fastpath to change the characteristics of a volume group, enter:

```
smit chvg
```

For more information, refer to "Step 7: Change Volume Group Characteristics on the Secondary Node" on page 128.

Alternatively, to use the command line interface to change the volume group to remain dormant at startup, enter:

```
chvg -a  n' -Q y' volume_group_name
```

Step 7: Vary Off the Volume Group on the Secondary Node

Use the **varyoffvg** command to quiesce the affected volume group after making the change.

To vary off the volume group so that it can be activated as appropriate by the IBM Recoverable Virtual Shared Disk recovery scripts, enter:

```
varyoffvg volume_group_name
```

Step 8: Vary On the Volume Group on the Primary Node

Use the **varyonvg** command to quiesce the affected volume group after making the change.

To vary on the volume group so that it can be activated as appropriate by the IBM Recoverable Virtual Shared Disk recovery scripts, enter:

```
varyonvg volume_group_name
```

Step 9: Complete Follow-up Tasks

Once the you have extended the volume group, do the following tasks:

- Make sure the primary and secondary nodes are active.
- Use **defvsd** to define any new virtual shared disks.
- Use the **cfgvsd** command for all the virtual shared disks in this volume group at all the nodes.
- Use the **startvsd** command at all the virtual shared disk nodes to start these new virtual shared disks.

Reducing a Twin-Tailed Volume Group

The following table summarizes the steps you do on both the primary and secondary nodes to reduce (remove one or more physical volumes from) a twin-tailed volume group:

Table 15. Procedure for Reducing a Twin-Tailed Volume Group

Step Number	Primary Node	Secondary Node
Step 1	Complete prerequisite tasks	Complete prerequisite tasks
Step 2		Export a volume group
Step 3	Vary on a volume group	
Step 4	Remove data from the physical volume	
Step 5	Reduce a twin-tailed volume group	
Step 6	Vary off the volume group	
Step 7		Import volume group
Step 8		Change volume group characteristics
Step 9		Vary off the volume group
Step 10	Complete follow-up tasks	Complete follow-up tasks

Complete the following tasks to reduce a twin-tailed volume group.

Step 1: Complete Prerequisite Tasks

1. Both the primary and secondary nodes must be up and in the active group.
2. Suspend, stop, and unconfigure virtual shared disks in the affected volume group on all nodes.

Step 2: Export Volume Group Information from the Secondary Node

Before making any changes to the LVM elements on the primary node, you must export the appropriate volume group from the secondary node. Exporting the volume group deletes the information about this volume group from the ODM.

To use the **exportvg** command to export the volume group containing the component you are going to change on the secondary node, enter:

```
exportvg volume_group_name
```

Step 3: Vary on a Volume Group on the Primary Node

To use the **varyonvg** command to verify that the affected volume group is varied on after making the change, enter:

```
varyonvg volume_group_name
```

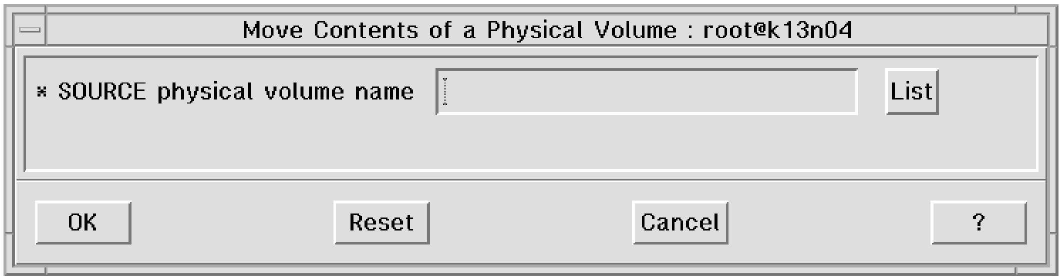
Step 4: Move Data from the Volume to Be Removed

Use the **smit migratepv** fastpath to move data on the physical volume being removed from the volume group to a different physical volume. If you do not, **data will be lost**.

1. As the root user on the primary node, enter:

   ```
   smit migratepv
   ```

   SMIT returns a screen similar to the following:

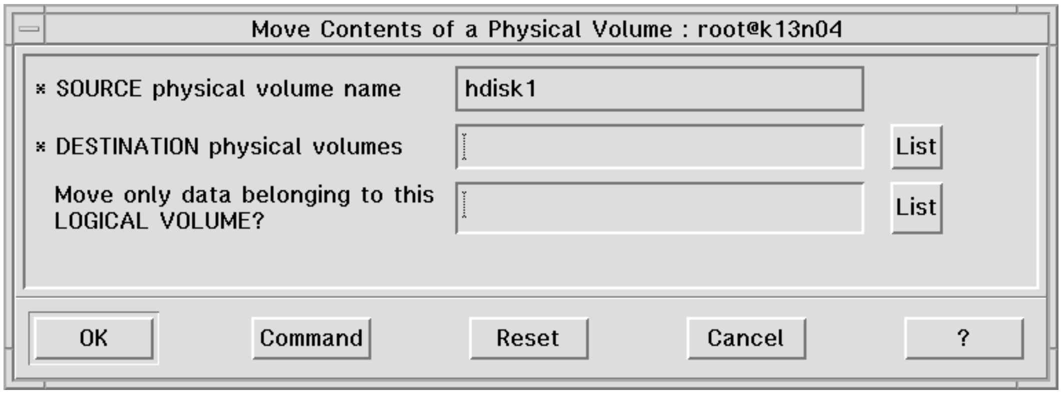


Move Contents of a Physical Volume : root@k13n04

* SOURCE physical volume name

2. Enter the names of the physical volumes being moved in the SOURCE physical volume name field or click on List and select the names from the display. Click on OK.

   SMIT returns a screen similar to the following. The physical volumes being migrated are entered in the SOURCE physical volume names field.



Move Contents of a Physical Volume : root@k13n04

* SOURCE physical volume name: hdisk1
* DESTINATION physical volumes

Move only data belonging to this LOGICAL VOLUME?

3. Enter field values as follows:

   DESTINATION physical volumes
   Enter the names of the physical volumes to which you want to move the data.

   Move only data belonging to this LOGICAL VOLUME?
   Set this field to **no**.

4. Click on OK to migrate the physical volumes.

5. After the command completes, press F12 to exit SMIT and return to the command line.

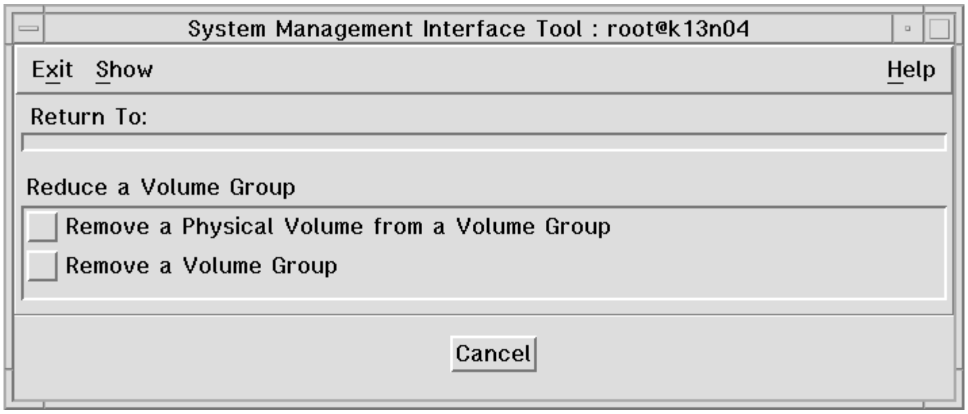
Step 5: Reduce the Shared Volume Group on the Primary Node

Use the **smit reducevg** fastpath to reduce a volume group.

1. As the root user at the primary node, enter:

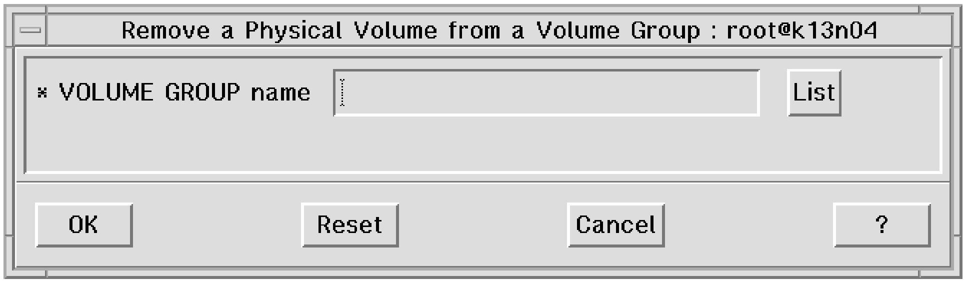
   ```
   smit reducevg
   ```

   SMIT returns a screen similar to the following:


System Management Interface Tool : root@k13n04
Exit
Show
Help
Return To:
Reduce a Volume Group
Remove a Physical Volume from a Volume Group
Remove a Volume Group
Cancel

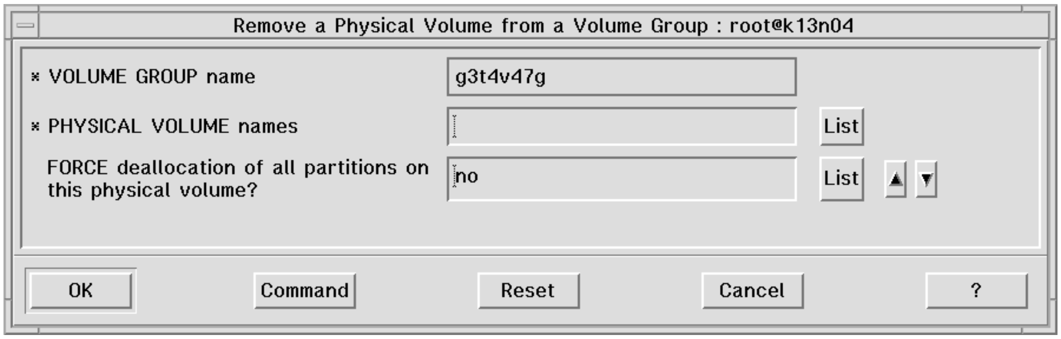
2. Select the **Remove a Physical Volume from a Volume Group** option and click on OK.

   SMIT returns a screen similar to the following:



3. Enter the name of the volume group from which you are removing the physical volumes in the VOLUME GROUP name field. Click on OK.

   SMIT returns a screen similar to the following. The name of the volume group you specified in the preceding screen is entered.



Remove a Physical Volume from a Volume Group : root@k13n04

* VOLUME GROUP name: g3t4v47g

* PHYSICAL VOLUME names:

FORCE deallocation of all partitions on this physical volume?: no

4. Enter field values as follows:

   PHYSICAL VOLUME names
   Enter the name of the physical volumes to remove from the volume group.

   FORCE deallocation of all partitions on this physical volume?
   Set this field to **no**.

5. Click on OK to remove the designated physical volumes from the volume group.
6. After the command completes, press F12 to exit SMIT and return to the command line.

Step 6: Vary Off the Volume Group on the Primary Node

Use the **varyoffvg** command to quiesce the affected volume group after making the change.

To vary off the volume group so that it can be activated as appropriate by the IBM Recoverable Virtual Shared Disk recovery scripts, enter:

```
varyoffvg volume_group_name
```

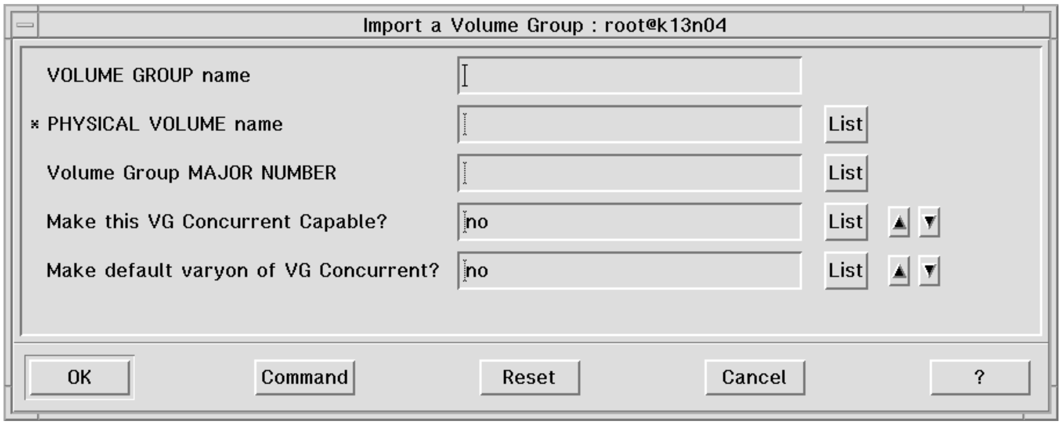
Step 7: Import Volume Group Information onto the Secondary Node

You now return to the secondary node. First, import the volume group on the secondary node. Importing the volume group onto the secondary node synchronizes the ODM definition of the volume group on the node.

1. To use the **smit importvg** fastpath to import the volume group, enter:

   ```
   smit importvg
   ```

   SMIT returns a screen similar to the following:



Import a Volume Group : root@k13n04

```
  VOLUME GROUP name
* PHYSICAL VOLUME name
  Volume Group MAJOR NUMBER
  Make this VG Concurrent Capable?        no
  Make default varyon of VG Concurrent?   no
```

2. Enter field values as follows:

   VOLUME GROUP name Enter the name of the volume group that you are importing. Make sure the volume group name is the same name that you used on the primary node.

   PHYSICAL VOLUME name
   Enter the name of one of the physical volumes that resides in the volume group. Note that a disk can have a different physical name on different nodes. Make sure that you use the disk name as it is defined on the *secondary* node. Every **hdisk** has an ID. If you know that the name of a disk on the primary node is *hdisk01*, use the name that matches that ID on the secondary node. You can use the **lspv** command to determine the name.

   Volume Group MAJOR NUMBER
   Use the default, the next available number in the valid range.

   Make this VG Concurrent Capable?
   Accept the default of **no**. Do not specify **yes**.

   Make default varyon of VG Concurrent?
   Accept the default of **no**. Do not specify **yes**.

3. Click on OK.

4. Press F12 to exit SMIT and return to the command line.

Step 8: Change Volume Group Characteristics on the Secondary Node

By default, a volume group that was just imported is configured to automatically become active at system restart. However, a recoverable virtual shared disk volume group should be varied on as appropriate by the IBM Recoverable Virtual Shared Disk recovery scripts. Therefore, after importing a volume group, use the SMIT Change a Volume Group screen to reconfigure the volume group to remain dormant at startup.

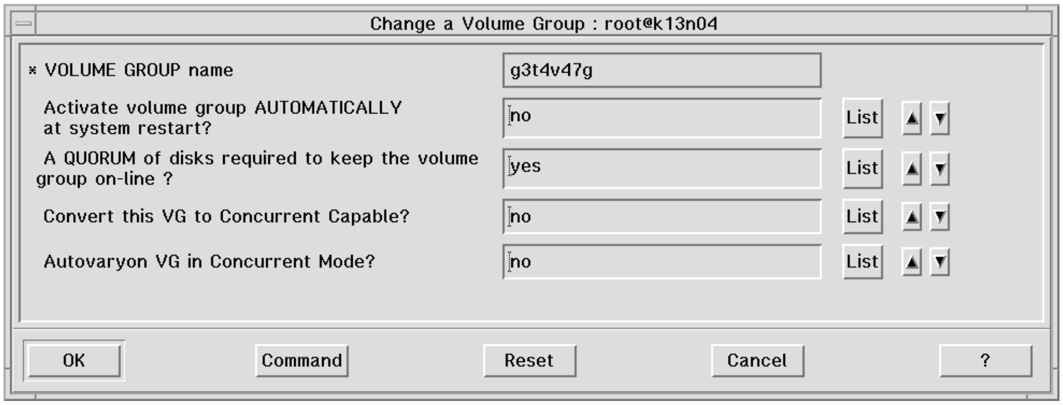
1. To use the **smit chvg** fastpath to change the characteristics of a volume group, enter:

   ```
   smit chvg
   ```

SMIT prompts you to select the volume group.

2. To select the volume group you just imported, click on OK.

   A screen similar to the following appears. The first field contains the volume group name you specified.



Change a Volume Group : root@k13n04

* VOLUME GROUP name	g3t4v47g
Activate volume group AUTOMATICALLY at system restart?	no
A QUORUM of disks required to keep the volume group on-line ?	yes
Convert this VG to Concurrent Capable?	no
Autovaryon VG in Concurrent Mode?	no

3. Enter remaining field values as follows:

   ACTIVATE volume group automatically at system restart?
   Set this field to **no**.

   A QUORUM of disks required to keep the volume group on-line?
   Accept the default, **yes**.

   Convert this VG to Concurrent Capable?
   Accept the default **no**. Do not specify **yes**.

   Autovaryon VG in Concurrent Mode?
   Accept the default **no**. Do not specify **yes**.

4. Click on OK.
5. Press F12 to exit SMIT and return to the command line.

Step 9: Vary Off the Volume Group on the Secondary Node

Use the **varyoffvg** command to quiesce the affected volume group after making the change.

You vary off the volume group so that it can be activated as appropriate by the IBM Recoverable Virtual Shared Disk recovery scripts. Enter:

`varyoffvg` *`volume_group_name`*

Step 10: Complete Follow-up Tasks

Once the you have reduced the volume group, complete the following tasks:

- Make sure the primary and secondary nodes are active.
- Use **undefvsd** to undefine no longer required virtual shared disks.
- Use the **cfgvsd** command for all the virtual shared disks in this volume group at all the nodes.
- Use the **startvsd** command at all the virtual shared disk nodes to start these new virtual shared disks.

Removing a Twin-Tailed Volume Group

See the following table for an overview of the steps needed to remove a twin-tailed volume group.

Table 16. Steps to Remove a Shared Volume Group

Step Number	Primary Node	Secondary Node
Step 1	Complete prerequisite tasks	Complete prerequisite tasks
Step 2		Export volume group information
Step 3	Vary on a volume group	
Step 4		Delete a twin-tailed volume group

Complete the following steps to remove a twin-tailed volume group.

Step 1: Complete Prerequisite Tasks

1. Suspend, stop, and unconfigure the active virtual shared disks on this volume group on all nodes.
2. Undefine the volume group with **vsdelvg** at the control workstation.

Step 2: Export Volume Group Information from the Secondary Node

Before making any changes to the LVM elements on the primary node, you must export the appropriate volume group from the secondary node. Exporting the volume group deletes the information about this volume group from the ODM. To use the **exportvg** command to export the volume group containing the component you are going to change on the secondary nodes, enter:

```
exportvg volume_group_name
```

Step 3: Vary On a Volume Group on the Primary Node

To use the **varyonvg** command to activate the affected volume group after making the change, enter:

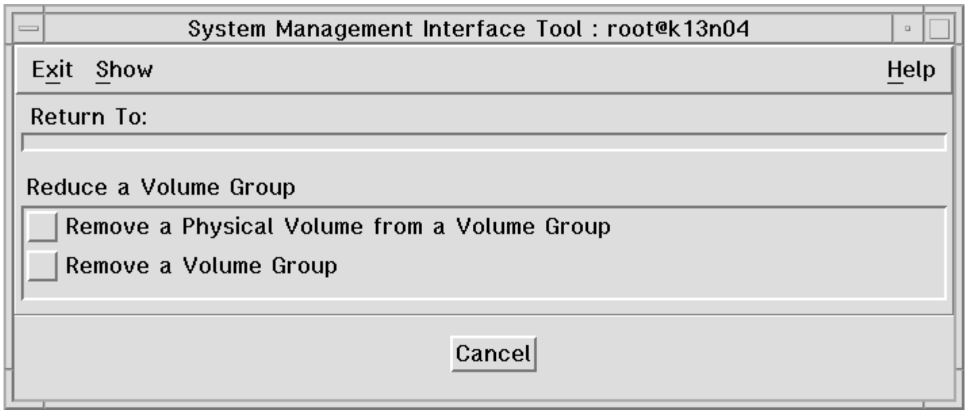
```
varyonvg volume_group_name
```

Step 4: Delete a Twin-tailed Volume Group on the Primary Node

1. As the root user at the primary node, enter:

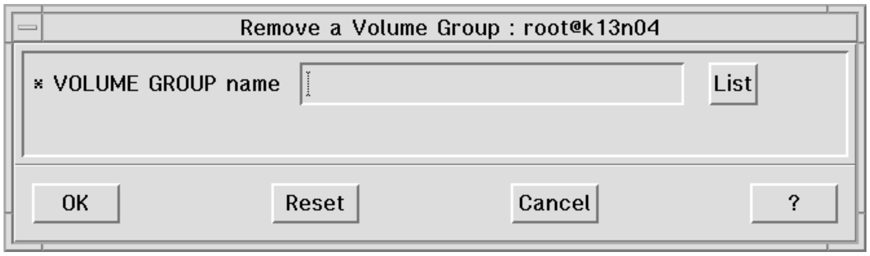
   ```
   smit reducevg
   ```

   SMIT returns a screen similar to the following:


System Management Interface Tool : root@k13n04
Exit
Show
Help
Return To:
Reduce a Volume Group
□ Remove a Physical Volume from a Volume Group
□ Remove a Volume Group
Cancel

2. Select the **Remove a Volume Group** option and click on OK.

   SMIT returns a screen similar to the following:



3. Enter the name of the volume group to be removed. Click on OK.
4. When the command completes, press 10 to leave SMIT and return to the command line.

Appendix C. Setting Up 7134 Disks as Twin-tailed Volume Groups

If you are using 7134 disks to set up twin-tailed volume groups, making and importing the volume groups is the same as on 9333 and 9334 disks, but there is some unique preparation work that needs to be done.

To install the 7134 disks as twin-tailed volume groups:

1. Connect the 7134 drawer to two nodes.
2. Reboot (or run **cfgmgr**) on both nodes.
3. On each node, run

   ```
   lsdev -C | grep “2.0 GB 16 Bit Differential SCSI Disk Drive”
   ```

   A list of the disks, similar to the following, is output:

   ```
   hdisk8  Available 00-14-01-00 2.0 GB 16 Bit Differential SCSI Disk Drive
   hdisk9  Available 00-14-01-10 2.0 GB 16 Bit Differential SCSI Disk Drive
   hdisk14 Available 00-14-01-20 2.0 GB 16 Bit Differential SCSI Disk Drive
   hdisk15 Available 00-14-01-30 2.0 GB 16 Bit Differential SCSI Disk Drive
   hdisk16 Available 00-14-01-40 2.0 GB 16 Bit Differential SCSI Disk Drive
   hdisk17 Available 00-14-01-50 2.0 GB 16 Bit Differential SCSI Disk Drive
   ```

4. Rerun **lsdev** and **grep** for the slot number displayed for the disks in the previous step to get the name of the adapters for these disks. For example, the output from the previous step displayed that the disks are in slot 14:

   ```
   lsdev -C | grep “00\-14”
   ```

   The output from this command looks similar to the following:

   ```
   ascsi0  Available 00-14       Wide SCSI I/O Controller Adapter
   vscsi0  Available 00-14-00    SCSI I/O Controller Protocol Device
   vscsi1  Available 00-14-01    SCSI I/O Controller Protocol Device
   hdisk8  Available 00-14-01-00 2.0 GB 16 Bit Differential SCSI Disk Drive
   hdisk9  Available 00-14-01-10 2.0 GB 16 Bit Differential SCSI Disk Drive
   hdisk14 Available 00-14-01-20 2.0 GB 16 Bit Differential SCSI Disk Drive
   hdisk15 Available 00-14-01-30 2.0 GB 16 Bit Differential SCSI Disk Drive
   hdisk16 Available 00-14-01-40 2.0 GB 16 Bit Differential SCSI Disk Drive
   hdisk17 Available 00-14-01-50 2.0 GB 16 Bit Differential SCSI Disk Drive
   ```

5. As displayed in the above output, **vscsi1** is the controller for these disks because it is at 00-14-01; the disks are also coming from 00-14-01. Move the disks from the Available state to the Defined state by removing all the devices except **ascsi0** (but don't remove their definition from the ODM). For example:

   ```
   rmdev -l vscsi0
   rmdev -l vscsi1
   rmdev -l hdisk8
   rmdev -l hdisk9

   ⋮
   ```

   Note: Do steps 3, 4, and 5 on both nodes.

6. On one node, use SMIT to change the External SCSI ID from **7** to something else (**6**, for example).

   a. Enter **smit**

   b. Select **Devices**

   c. Select **SCSI Adapter**

   d. Select **Change/Show Characteristics of a SCSI Adapter**

   e. Select **ascsi0**

   f. Change **Adapter card SCSI ID** to another ID

7. Run **cfgmgr** on both nodes.

8. Run **lspv** on both nodes. There should be a physical volume ID for all of the 7134 disks. If there is not a physical volume ID for the 7134 disks, try removing the devices and rerunning **cfgmgr**. Note that it should not say **none** in this column.

9. From here, proceed as you normally would to make volume groups, logical volumes, and so forth.

Bibliography

This bibliography helps you find product documentation related to the RS/6000 SP hardware and software products.

You can find most of the IBM product information for RS/6000 SP products on the World Wide Web. Formats for both viewing and downloading are available.

PSSP documentation is shipped with the PSSP product in a variety of formats and can be installed on your system. The man pages for public code that PSSP includes are also available online.

You can order hard copies of the product documentation from IBM. This bibliography lists the titles that are available and their order numbers.

Finally, this bibliography contains a list of non-IBM publications that discuss parallel computing and other topics related to the RS/6000 SP.

Finding Documentation on the World Wide Web

Most of the RS/6000 SP hardware and software books are available from the IBM RS/6000 web site at **http://www.rs6000.ibm.com**. You can view a book or download a Portable Document Format (PDF) version of it. At the time this manual was published, the full path to the "RS/6000 SP Product Documentation Library" page was **http://www.rs6000.ibm.com/resource/aix_resource/sp_books**. However, the structure of the RS/6000 web site can change over time.

Accessing PSSP Documentation Online

On the same medium as the PSSP product code, IBM ships PSSP man pages, HTML files, and PDF files. In order to use these publications, you must first install the **ssp.docs** file set.

To view the PSSP HTML publications, you need access to an HTML document browser such as Netscape. The HTML files and an index that links to them are installed in the **/usr/lpp/ssp/html** directory. Once installed, you can also view the HTML files from the RS/6000 SP Resource Center.

If you have installed the SP Resource Center on your SP system, you can access it by entering the **/usr/lpp/ssp/bin/resource_center** command. If you have the SP Resource Center on CD-ROM, see the **readme.txt** file for information about how to run it.

To view the PSSP PDF publications, you need access to the Adobe Acrobat Reader 3.0.1. The Acrobat Reader is shipped with the AIX Version 4.3 Bonus Pack and is also freely available for downloading from the Adobe web site at URL **http://www.adobe.com**.

Manual Pages for Public Code

The following manual pages for public code are available in this product:

SUP	/usr/lpp/ssp/man/man1/sup.1
NTP	/usr/lpp/ssp/man/man8/xntpd.8
	/usr/lpp/ssp/man/man8/xntpdc.8
Perl (Version 4.036)	/usr/lpp/ssp/perl/man/perl.man
	/usr/lpp/ssp/perl/man/h2ph.man

/usr/lpp/ssp/perl/man/s2p.man

/usr/lpp/ssp/perl/man/a2p.man

Perl (Version 5.003) Man pages are in the /usr/lpp/ssp/perl5/man/man1 directory

Manual pages and other documentation for **Tcl**, **TclX**, **Tk**, and **expect** can be found in the compressed **tar** files located in the **/usr/lpp/ssp/public** directory.

RS/6000 SP Planning Publications

This section lists the IBM product documentation for planning for the IBM RS/6000 SP hardware and software.

IBM RS/6000 SP:

- *Planning, Volume 1, Hardware and Physical Environment*, GA22-7280
- *Planning, Volume 2, Control Workstation and Software Environment*, GA22-7281

RS/6000 SP Hardware Publications

This section lists the IBM product documentation for the IBM RS/6000 SP hardware.

IBM RS/6000 SP:

- *Planning, Volume 1, Hardware and Physical Environment*, GA22-7280
- *Planning, Volume 2, Control Workstation and Software Environment*, GA22-7281
- *Maintenance Information, Volume 1, Installation and Relocation*, GA22-7375
- *Maintenance Information, Volume 2, Maintenance Analysis Procedures*, GA22-7376
- *Maintenance Information, Volume 3, Locations and Service Procedures*, GA22-7377
- *Maintenance Information, Volume 4, Parts Catalog*, GA22-7378

RS/6000 SP Switch Router Publications

The RS/6000 SP Switch Router is based on the Ascend GRF switched IP router product from Ascend Communications, Inc.. You can order the SP Switch Router as the IBM 9077.

The following publications are shipped with the SP Switch Router. You can also order these publications from IBM using the order numbers shown.

- *Ascend GRF Getting Started*, GA22-7368
- *Ascend GRF Configuration Guide*, GA22-7366
- *Ascend GRF Reference Guide*, GA22-7367
- *IBM SP Switch Router Adapter Guide*, GA22-7310.

RS/6000 SP Software Publications

This section lists the IBM product documentation for software products related to the IBM RS/6000 SP. These products include:

- IBM Parallel System Support Programs for AIX (PSSP)
- IBM LoadLeveler for AIX (LoadLeveler)
- IBM Parallel Environment for AIX (Parallel Environment)
- IBM General Parallel File System for AIX (GPFS)

- IBM Engineering and Scientific Subroutine Library (ESSL) for AIX
- IBM Parallel ESSL for AIX
- IBM High Availability Cluster Multi-Processing for AIX (HACMP)
- IBM Client Input Output/Sockets (CLIO/S)
- IBM Network Tape Access and Control System for AIX (NetTAPE)

PSSP Publications

IBM RS/6000 SP:

- *Planning, Volume 2, Control Workstation and Software Environment*, GA22-7281

PSSP:

- *Installation and Migration Guide*, GA22-7347
- *Administration Guide*, SA22-7348
- *Managing Shared Disks*, SA22-7349
- *Performance Monitoring Guide and Reference*, SA22-7353
- *Diagnosis Guide*, GA22-7350
- *Command and Technical Reference*, SA22-7351
- *Messages Reference*, GA22-7352

RS/6000 Cluster Technology (RSCT):

- *Event Management Programming Guide and Reference*, SA22-7354
- *Group Services Programming Guide and Reference*, SA22-7355

As an alternative to ordering the individual books, you can use SBOF-8587 to order the PSSP software library.

LoadLeveler Publications

LoadLeveler:

- *Using and Administering*, SA22-7311
- *Diagnosis and Messages Guide*, GA22-7277

GPFS Publications

GPFS:

- *Installation and Administration Guide*, SA22-7278

Parallel Environment Publications

Parallel Environment:

- *Installation Guide*, GC28-1981
- *Hitchhiker's Guide*, GC23-3895
- *Operation and Use, Volume 1*, SC28-1979
- *Operation and Use, Volume 2*, SC28-1980
- *MPI Programming and Subroutine Reference*, GC23-3894
- *MPL Programming and Subroutine Reference*, GC23-3893
- *Messages*, GC28-1982

As an alternative to ordering the individual books, you can use SBOF-8588 to order the PE library.

Parallel ESSL and ESSL Publications

- *ESSL Products: General Information*, GC23-0529
- *Parallel ESSL: Guide and Reference*, SA22-7273
- *ESSL: Guide and Reference*, SA22-7272

HACMP Publications

HACMP:

- *Concepts and Facilities*, SC23-1938
- *Planning Guide*, SC23-1939
- *Installation Guide*, SC23-1940
- *Administration Guide*, SC23-1941
- *Troubleshooting Guide*, SC23-1942
- *Programming Locking Applications*, SC23-1943
- *Programming Client Applications*, SC23-1944
- *Master Index and Glossary*, SC23-1945
- *HANFS for AIX Installation and Administration Guide*, SC23-1946
- *Enhanced Scalability Installation and Administration Guide*, SC23-1972

CLIO/S Publications

CLIO/S:

- *General Information*, GC23-3879
- *User's Guide and Reference*, GC28-1676

NetTAPE Publications

NetTAPE:

- *General Information*, GC23-3990
- *User's Guide and Reference*, available from your IBM representative

AIX and Related Product Publications

For the latest information on AIX and related products, including RS/6000 hardware products, see *AIX and Related Products Documentation Overview*, SC23-2456. You can order a hard copy of the book from IBM. You can also view it online from the "AIX Online Publications and Books" page of the RS/6000 web site, at URL **http://www.rs6000.ibm.com/resource/aix_resource/Pubs**.

Red Books

IBM's International Technical Support Organization (ITSO) has published a number of redbooks related to the RS/6000 SP. For a current list, see the ITSO website, at URL **http://www.redbooks.ibm.com**.

Non-IBM Publications

Here are some non-IBM publications that you may find helpful.

- Almasi, G., Gottlieb, A., *Highly Parallel Computing*, Benjamin-Cummings Publishing Company, Inc., 1989.
- Foster, I., *Designing and Building Parallel Programs*, Addison-Wesley, 1995.
- Gropp, W., Lusk, E., Skjellum, A., *Using MPI*, The MIT Press, 1994.
- Message Passing Interface Forum, *MPI: A Message-Passing Interface Standard, Version 1.1*, University of Tennessee, Knoxville, Tennessee, June 6, 1995.
- Message Passing Interface Forum, *MPI-2: Extensions to the Message-Passing Interface, Version 2.0*, University of Tennessee, Knoxville, Tennessee, July 18, 1997.
- Ousterhout, John K., *Tcl and the Tk Toolkit*, Addison-Wesley, Reading, MA, 1994, ISBN 0-201-63337-X.
- Pfister, Gregory, F., *In Search of Clusters*, Prentice Hall, 1998.

Glossary of Terms and Abbreviations

This glossary includes terms and definitions from:

- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.
- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies can be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions are identified by the symbol (A) after the definition.
- The *ANSI/EIA Standard - 440A: Fiber Optic Terminology* copyright 1989 by the Electronics Industries Association (EIA). Copies can be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue N.W., Washington, D.C. 20006. Definitions are identified by the symbol (E) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

The following cross-references are used in this glossary:

Contrast with. This refers to a term that has an opposed or substantively different meaning.
See. This refers the reader to multiple-word terms in which this term appears.
See also. This refers the reader to terms that have a related, but not synonymous, meaning.
Synonym for. This indicates that the term has the same meaning as a preferred term, which is defined in the glossary.

This section contains some of the terms that are commonly used in the SP publications.

IBM is grateful to the American National Standards Institute (ANSI) for permission to reprint its definitions from the American National Standard *Vocabulary for Information Processing* (Copyright 1970 by American National Standards Institute, Incorporated), which was prepared by Subcommittee X3K5 on Terminology and Glossary of the American National Standards Committee X3. ANSI definitions are preceded by an asterisk (*).

Other definitions in this glossary are taken from *IBM Vocabulary for Data Processing, Telecommunications, and Office Systems* (SC20-1699) and *IBM DATABASE 2 Application Programming Guide for TSO Users* (SC26-4081).

A

adapter. An adapter is a mechanism for attaching parts. For example, an adapter could be a part that electrically or physically connects a device to a computer or to another device. In the SP system, network connectivity is supplied by various adapters, some optional, that can provide connection to I/O devices, networks of workstations, and mainframe networks. Ethernet, FDDI, token-ring, HiPPI, SCSI, FCS, and ATM are examples of adapters that can be used as part of an SP system.

address. A character or group of characters that identifies a register, a device, a particular part of storage, or some other data source or destination.

AFS. A distributed file system that provides authentication services as part of its file system creation.

AIX. Abbreviation for Advanced Interactive Executive, IBM's licensed version of the UNIX operating system. AIX is particularly suited to support technical computing applications, including high function graphics and floating point computations.

Amd. Berkeley Software Distribution automount daemon.

API. Application Programming Interface. A set of programming functions and routines that provide access between the Application layer of the OSI seven-layer model and applications that want to use the network. It is a software interface.

application. The use to which a data processing system is put; for example, a payroll application, an airline reservation application.

application data. The data that is produced using an application program.

ARP. Address Resolution Protocol.

ATM. Asynchronous Transfer Mode. (See *TURBOWAYS 100 ATM Adapter*.)

Authentication. The process of validating the identity of a user or server.

Authorization. The process of obtaining permission to perform specific actions.

B

batch processing. * (1) The processing of data or the accomplishment of jobs accumulated in advance in such a manner that each accumulation thus formed is processed or accomplished in the same run. * (2) The processing of data accumulating over a period of time. * (3) Loosely, the execution of computer programs serially. (4) Computer programs executed in the background.

BMCA. Block Multiplexer Channel Adapter. The block multiplexer channel connection allows the RS/6000 to communicate directly with a host System/370 or System/390; the host operating system views the system unit as a control unit.

BOS. The AIX Base Operating System.

C

call home function. The ability of a system to call the IBM support center and open a PMR to have a repair scheduled.

CDE. Common Desktop Environment. A graphical user interface for UNIX.

charge feature. An optional feature for either software or hardware for which there is a charge.

CLI. Command Line Interface.

client. * (1) A function that requests services from a server and makes them available to the user. * (2) A term used in an environment to identify a machine that uses the resources of the network.

Client Input/Output Sockets (CLIO/S). A software package that enables high-speed data and tape access between SP systems, AIX systems, and ES/9000 mainframes.

CLIO/S. Client Input/Output Sockets.

CMI. Centralized Management Interface provides a series of SMIT menus and dialogues used for defining and querying the SP system configuration.

connectionless. A communication process that takes place without first establishing a connection.

connectionless network. A network in which the sending logical node must have the address of the receiving logical node before information interchange can begin. The packet is routed through nodes in the network based on the destination address in the packet. The originating source does not receive an acknowledgment that the packet was received at the destination.

control workstation. A single point of control allowing the administrator or operator to monitor and manage the SP system using the IBM AIX Parallel System Support Programs.

css. Communication subsystem.

D

daemon. A process, not associated with a particular user, that performs system-wide functions such as administration and control of networks, execution of time-dependent activities, line printer spooling and so forth.

DASD. Direct Access Storage Device. Storage for input/output data.

DCE. Distributed Computing Environment.

DFS. distributed file system. A subset of the IBM Distributed Computing Environment.

DNS. Domain Name Service. A hierarchical name service which maps high level machine names to IP addresses.

E

Error Notification Object. An object in the SDR that is matched with an error log entry. When an error log entry occurs that matches the Notification Object, a user-specified action is taken.

ESCON. Enterprise Systems Connection. The ESCON channel connection allows the RS/6000 to communicate directly with a host System/390; the host operating system views the system unit as a control unit.

Ethernet. (1) Ethernet is the standard hardware for TCP/IP local area networks in the UNIX marketplace. It is a 10-megabit per second baseband type LAN that allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by collision detection (CSMA/CD). (2) A passive coaxial cable whose interconnections contain devices or components, or both, that are all active. It uses CSMA/CD technology to provide a best-effort delivery system.

Ethernet network. A baseband LAN with a bus topology in which messages are broadcast on a coaxial cabling using the carrier sense multiple access/collision detection (CSMA/CD) transmission method.

event. In Event Management, the notification that an expression evaluated to true. This evaluation occurs each time an instance of a resource variable is observed.

expect. Programmed dialogue with interactive programs.

expression. In Event Management, the relational expression between a resource variable and other elements (such as constants or the previous value of an instance of the variable) that, when true, generates an event. An example of an expression is `X < 10` where `X` represents the resource variable `IBM.PSSP.aixos.PagSp.%totalfree` (the percentage of total free paging space). When the expression is true, that is, when the total free paging space is observed to be less than 10%, the Event Management subsystem generates an event to notify the appropriate application.

F

failover. Also called fallover, the sequence of events when a primary or server machine fails and a secondary or backup machine assumes the primary workload. This is a disruptive failure with a short recovery time.

fall back. Also called fallback, the sequence of events when a primary or server machine takes back control of its workload from a secondary or backup machine.

FDDI. Fiber Distributed Data Interface.

Fiber Distributed Data Interface (FDDI). An American National Standards Institute (ANSI) standard for 100-megabit-per-second LAN using optical fiber cables. An FDDI local area network (LAN) can be up to 100 km (62 miles) and can include up to 500 system units. There can be up to 2 km (1.24 miles) between system units and/or concentrators.

file. * A set of related records treated as a unit, for example, in stock control, a file could consist of a set of invoices.

file name. A CMS file identifier in the form of 'filename filetype filemode' (like: TEXT DATA A).

file server. A centrally located computer that acts as a storehouse of data and applications for numerous users of a local area network.

File Transfer Protocol (FTP). The Internet protocol (and program) used to transfer files between hosts. It is an application layer protocol in TCP/IP that uses TELNET and TCP protocols to transfer bulk-data files between machines or hosts.

foreign host. Any host on the network other than the local host.

FTP. File transfer protocol.

G

gateway. An intelligent electronic device interconnecting dissimilar networks and providing protocol conversion for network compatibility. A gateway provides transparent access to dissimilar networks for nodes on either network. It operates at the session presentation and application layers.

H

HACMP. High Availability Cluster Multi-Processing for AIX.

HACWS. High Availability Control Workstation function, based on HACMP, provides for a backup control workstation for the SP system.

Hashed Shared Disk (HSD). The data striping device for the IBM Virtual Shared Disk. The device driver lets application programs stripe data across physical disks in multiple IBM Virtual Shared Disks, thus reducing I/O bottlenecks.

help key. In the SP graphical interface, the key that gives you access to the SP graphical interface help facility.

High Availability Cluster Multi-Processing. An IBM facility to cluster nodes or components to provide high availability by eliminating single points of failure.

HiPPI. High Performance Parallel Interface. RS/6000 units can attach to a HiPPI network as defined by the ANSI specifications. The HiPPI channel supports burst rates of 100 Mbps over dual simplex cables; connections can be up to 25 km in length as defined by the standard and can be extended using third-party HiPPI switches and fiber optic extenders.

home directory. The directory associated with an individual user.

host. A computer connected to a network, and providing an access method to that network. A host provides end-user services.

I

instance vector. Obsolete term for resource identifier.

Intermediate Switch Board. Switches mounted in the Sp Switch expansion frame.

Internet. A specific inter-network consisting of large national backbone networks such as APARANET, MILNET, and NSFnet, and a myriad of regional and campus networks all over the world. The network uses the TCP/IP protocol suite.

Internet Protocol (IP). (1) A protocol that routes data through a network or interconnected networks. IP acts as an interface between the higher logical layers and the physical network. This protocol, however, does not provide error recovery, flow control, or guarantee the reliability of the physical network. IP is a connectionless protocol. (2) A protocol used to route data from its source to it destination in an Internet environment.

IP address. A 32-bit address assigned to devices or hosts in an IP internet that maps to a physical address. The IP address is composed of a network and host portion.

ISB. Intermediate Switch Board.

K

Kerberos. A service for authenticating users in a network environment.

kernel. The core portion of the UNIX operating system which controls the resources of the CPU and allocates them to the users. The kernel is memory-resident, is said to run in “kernel mode” and is protected from user tampering by the hardware.

L

LAN. (1) Acronym for Local Area Network, a data network located on the user's premises in which serial transmission is used for direct data communication among data stations. (2) Physical network technology that transfers data a high speed over short distances. (3) A network in which a set of devices is connected to another for communication and that can be connected to a larger network.

local host. The computer to which a user's terminal is directly connected.

log database. A persistent storage location for the logged information.

log event. The recording of an event.

log event type. A particular kind of log event that has a hierarchy associated with it.

logging. The writing of information to persistent storage for subsequent analysis by humans or programs.

M

mask. To use a pattern of characters to control retention or elimination of portions of another pattern of characters.

menu. A display of a list of available functions for selection by the user.

Motif. The graphical user interface for OSF, incorporating the X Window System. Also called OSF/Motif.

MTBF. Mean time between failure. This is a measure of reliability.

MTTR. Mean time to repair. This is a measure of serviceability.

N

naive application. An application with no knowledge of a server that fails over to another server. Client to server retry methods are used to reconnect.

network. An interconnected group of nodes, lines, and terminals. A network provides the ability to transmit data to and receive data from other systems and users.

NFS. Network File System. NFS allows different systems (UNIX or non-UNIX), different architectures, or vendors connected to the same network, to access remote files in a LAN environment as though they were local files.

NIM. Network Installation Management is provided with AIX to install AIX on the nodes.

NIM client. An AIX system installed and managed by a NIM master. NIM supports three types of clients:

- Standalone
- Diskless
- Dataless

NIM master. An AIX system that can install one or more NIM clients. An AIX system must be defined as a NIM master before defining any NIM clients on that system. A NIM master managers the configuration database containing the information for the NIM clients.

NIM object. A representation of information about the NIM environment. NIM stores this information as objects in the NIM database. The types of objects are:

- Network
- Machine
- Resource

NIS. Network Information System.

node. In a network, the point where one or more functional units interconnect transmission lines. A computer location defined in a network. The SP system can house several different types of nodes for both serial and parallel processing. These node types can include thin nodes, wide nodes, 604 high nodes, as well as other types of nodes both internal and external to the SP frame.

Node Switch Board. Switches mounted on frames that contain nodes.

NSB. Node Switch Board.

NTP. Network Time Protocol.

O

ODM. Object Data Manager. In AIX, a hierarchical object-oriented database for configuration data.

P

parallel environment. A system environment where message passing or SP resource manager services are used by the application.

Parallel Environment. A licensed IBM program used for message passing applications on the SP or RS/6000 platforms.

parallel processing. A multiprocessor architecture which allows processes to be allocated to tightly coupled multiple processors in a cooperative processing environment, allowing concurrent execution of tasks.

parameter. * (1) A variable that is given a constant value for a specified application and that may denote the application. * (2) An item in a menu for which the operator specifies a value or for which the system provides a value when the menu is interpreted. * (3) A name in a procedure that is used to refer to an argument that is passed to the procedure. * (4) A particular piece of information that a system or application program needs to process a request.

partition. See system partition.

Perl. Practical Extraction and Report Language.

perspective. The primary window for each SP Perspectives application, so called because it provides a unique view of an SP system.

pipe. A UNIX utility allowing the output of one command to be the input of another. Represented by the | symbol. It is also referred to as filtering output.

PMR. Problem Management Report.

POE. Formerly Parallel Operating Environment, now Parallel Environment for AIX.

port. (1) An end point for communication between devices, generally referring to physical connection. (2) A 16-bit number identifying a particular TCP or UDP resource within a given TCP/IP node.

predicate. Obsolete term for expression.

Primary node or machine. (1) A device that runs a workload and has a standby device ready to assume the primary workload if that primary node fails or is taken out of service. (2) A node on the SP Switch that initializes, provides diagnosis and recovery services, and performs other operations to the switch network. (3) In IBM Virtual Shared Disk function, when physical disks are connected to two nodes (twin-tailed), one node is designated as the primary node for each disk and the other is designated the secondary, or backup, node. The primary node is the server node for IBM Virtual Shared Disks defined on the physical disks under normal conditions. The secondary node can become the server node for the disks if the primary node is unavailable (off-line or down).

Problem Management Report. The number in the IBM support mechanism that represents a service incident with a customer.

process. * (1) A unique, finite course of events defined by its purpose or by its effect, achieved under defined conditions. * (2) Any operation or combination of operations on data. * (3) A function being performed or waiting to be performed. * (4) A program in operation. For example, a daemon is a system process that is always running on the system.

protocol. A set of semantic and syntactic rules that defines the behavior of functional units in achieving communication.

Q

quorum. The Recoverable Virtual Shared Disk subsystem uses the notion of quorum, the majority of the virtual shared disk nodes, to cope with communication failures. If the nodes in a system partition are divided by a network failure so that the nodes in one group cannot communicate with the nodes in the other group, the quorum is used to decide which group to continue operating and which group to deactivate.

R

RAID. Redundant array of independent disks.

rearm expression. In Event Management, an expression used to generate an event that alternates with an original event expression in the following way: the event expression is used until it is true, then the rearm expression is used until it is true, then the event expression is used, and so on. The rearm expression is commonly the inverse of the event expression (for example, a resource variable is on or off). It can also be used with the event expression to define an upper and lower boundary for a condition of interest.

rearm predicate. Obsolete term for rearm expression

remote host. *See foreign host*.

resource. In Event Management, an entity in the system that provides a set of services. Examples of resources include hardware entities such as processors, disk drives, memory, and adapters, and software entities such as database applications, processes, and file systems. Each resource in the system has one or more attributes that define the state of the resource.

resource identifier. In Event Management, a set of elements, where each element is a name/value pair of the form `name=value`, whose values uniquely identify the copy of the resource (and by extension, the copy of the resource variable) in the system.

resource monitor. A program that supplies information about resources in the system. It can be a command, a daemon, or part of an application or subsystem that manages any type of system resource.

resource variable. In Event Management, the representation of an attribute of a resource. An example of a resource variable is `IBM.AIX.PagSp.%totalfree`, which represents the percentage of total free paging space. `IBM.AIX.PagSp` specifies the resource name and `%totalfree` specifies the resource attribute.

RISC. Reduced Instruction Set Computing (RISC), the technology for today's high performance personal computers and workstations, was invented in 1975. Uses a small simplified set of frequently used instructions for rapid execution.

rlogin (remote LOGIN). A service offered by Berkeley UNIX systems that allows authorized users of one machine to connect to other UNIX systems across a network and interact as if their terminals were connected directly. The rlogin software passes information about the user's environment (for example, terminal type) to the remote machine.

RPC. Acronym for Remote Procedure Call, a facility that a client uses to have a server execute a procedure call. This facility is composed of a library of procedures plus an XDR.

RSH. A variant of RLOGIN command that invokes a command interpreter on a remote UNIX machine and passes the command line arguments to the command interpreter, skipping the LOGIN step completely. See also *rlogin*.

S

SCSI. Small Computer System Interface.

Secondary node. In IBM Virtual Shared Disk function, when physical disks are connected to two nodes (twin-tailed), one node is designated as the primary node for each disk and the other is designated as the secondary, or backup, node. The secondary node acts as the server node for the IBM Virtual Shared disks defined on the physical disks if the primary node is unavailable (off-line or down).

server. (1) A function that provides services for users. A machine may run client and server processes at the same time. (2) A machine that provides resources to the network. It provides a network service, such as disk storage and file transfer, or a program that uses such a service. (3) A device, program, or code module on a network dedicated to providing a specific service to a network. (4) On a LAN, a data station that provides facilities to other data stations. Examples are file server, print server, and mail server.

shell. The shell is the primary user interface for the UNIX operating system. It serves as command language interpreter, programming language, and allows foreground and background processing. There are three different implementations of the shell concept: Bourne, C and Korn.

Small Computer System Interface (SCSI). An input and output bus that provides a standard interface for the attachment of various direct access storage devices (DASD) and tape drives to the RS/6000.

Small Computer Systems Interface Adapter (SCSI Adapter). An adapter that supports the attachment of various direct-access storage devices (DASD) and tape drives to the RS/6000.

SMIT. The System Management Interface Toolkit is a set of menu driven utilities for AIX that provides functions such as transaction login, shell script creation, automatic updates of object database, and so forth.

SNMP. Simple Network Management Protocol. (1) An IP network management protocol that is used to monitor attached networks and routers. (2) A TCP/IP-based protocol for exchanging network management information and outlining the structure for communications among network devices.

socket. (1) An abstraction used by Berkeley UNIX that allows an application to access TCP/IP protocol functions. (2) An IP address and port number pairing. (3) In TCP/IP, the Internet address of the host computer on which the application runs, and the port number it uses. A TCP/IP application is identified by its socket.

standby node or machine. A device that waits for a failure of a primary node in order to assume the identity of the primary node. The standby machine then runs the primary's workload until the primary is back in service.

subnet. Shortened form of subnetwork.

subnet mask. A bit template that identifies to the TCP/IP protocol code the bits of the host address that are to be used for routing for specific subnetworks.

subnetwork. Any group of nodes that have a set of common characteristics, such as the same network ID.

subsystem. A software component that is not usually associated with a user command. It is usually a daemon process. A subsystem will perform work or provide services on behalf of a user request or operating system request.

SUP. Software Update Protocol.

Sysctl. Secure System Command Execution Tool. An authenticated client/server system for running commands remotely and in parallel.

syslog. A BSD logging system used to collect and manage other subsystem's logging data.

System Administrator. The user who is responsible for setting up, modifying, and maintaining the SP system.

system partition. A group of nonoverlapping nodes on a switch chip boundary that act as a logical SP system.

T

tar. Tape ARchive, is a standard UNIX data archive utility for storing data on tape media.

Tcl. Tool Command Language.

TclX. Tool Command Language Extended.

TCP. Acronym for Transmission Control Protocol, a stream communication protocol that includes error recovery and flow control.

TCP/IP. Acronym for Transmission Control Protocol/Internet Protocol, a suite of protocols designed to allow communication between networks regardless of the technologies implemented in each network. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It assumes that the underlying protocol is the Internet Protocol.

Telnet. Terminal Emulation Protocol, a TCP/IP application protocol that allows interactive access to foreign hosts.

Tk. Tcl-based Tool Kit for X Windows.

TMPCP. Tape Management Program Control Point.

token-ring. (1) Network technology that controls media access by passing a token (special packet or frame) between media-attached machines. (2) A network with a ring topology that passes tokens from one attaching device (node) to another. (3) The IBM Token-Ring LAN connection allows the RS/6000 system unit to participate in a LAN adhering to the IEEE 802.5 Token-Passing Ring standard or the ECMA standard 89 for Token-Ring, baseband LANs.

transaction. An exchange between the user and the system. Each activity the system performs for the user is considered a transaction.

transceiver (transmitter-receiver). A physical device that connects a host interface to a local area network, such as Ethernet. Ethernet transceivers contain electronics that apply signals to the cable and sense collisions.

transfer. To send data from one place and to receive the data at another place. Synonymous with move.

transmission. * The sending of data from one place for reception elsewhere.

TURBOWAYS 100 ATM Adapter. An IBM high-performance, high-function intelligent adapter that provides dedicated 100 Mbps ATM (asynchronous transfer mode) connection for high-performance servers and workstations.

U

UDP. User Datagram Protocol.

UNIX operating system. An operating system developed by Bell Laboratories that features multiprogramming in a multiuser environment. The UNIX operating system was originally developed for use on minicomputers, but has been adapted for mainframes and microcomputers. **Note:** The AIX operating system is IBM's implementation of the UNIX operating system.

user. Anyone who requires the services of a computing system.

User Datagram Protocol (UDP). (1) In TCP/IP, a packet-level protocol built directly on the Internet Protocol layer. UDP is used for application-to-application programs between TCP/IP host systems. (2) A transport protocol in the Internet suite of protocols that provides unreliable, connectionless datagram service. (3) The Internet Protocol that enables an application programmer on one machine or process to send a datagram to an application program on another machine or process.

user ID. A nonnegative integer, contained in an object of type *uid_t*, that is used to uniquely identify a system user.

V

Virtual Shared Disk, IBM. The function that allows application programs executing at different nodes of a system partition to access a raw logical volume as if it were local at each of the nodes. In actuality, the logical volume is local at only one of the nodes (the server node).

W

workstation. * (1) A configuration of input/output equipment at which an operator works. * (2) A terminal or microcomputer, usually one that is connected to a mainframe or to a network, at which a user can perform applications.

X

X Window System. A graphical user interface product.

Index

Communicating Your Comments to IBM

IBM Parallel System Support Programs for AIX
Managing Shared Disks
Version 3 Release 1

Publication No. SA22-7349-00

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a reader's comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
  - FAX: (International Access Code)+1+914+432-9405
- If you prefer to send comments electronically, use this network ID:
  - IBM Mail Exchange: USIB6TC9 at IBMMAIL
  - Internet e-mail: mhvrcfs@us.ibm.com
  - World Wide Web: http://www.s390.ibm.com/os390

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies

Optionally, if you include your telephone number, we will be able to respond to your comments by phone.

Reader's Comments — We'd Like to Hear from You

IBM Parallel System Support Programs for AIX
Managing Shared Disks
Version 3 Release 1

Publication No. SA22-7349-00

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Today's date: ____________________________________

What is your occupation?

Newsletter number of latest Technical Newsletter (if any) concerning this publication:

How did you use this publication?

[ ] As an introduction
[ ] As a reference manual
[ ] For another purpose (explain)

[ ] As a text (student)
[ ] As a text (instructor)

____

____

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number: Comment:

____
Name

____
Company or Organization

____
Phone No.

____
Address

____

____

Reader's Comments — We'd Like to Hear from You
SA22-7349-00

Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
522 South Road
Poughkeepsie NY 12601-5400

Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line

Program Number: 5765-D51

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SA22-7349-00