

reference
guide

hp StorageWorks storage system scripting utility enterprise virtual array v3.0

Product Version: 3.0

Third Edition (May 2003)

Part Number: AA-RU5HA-TE

This guide describes the commands available in the Storage System Scripting Utility (SSSU) for the Enterprise Virtual Array V3.0. These commands configure and control HSV controllers.



© 2003 Hewlett-Packard Company

Hewlett-Packard Company makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

This document contains proprietary information, which is protected by copyright. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of Hewlett-Packard. The information contained in this document is subject to change without notice.

Compaq Computer Corporation is a wholly-owned subsidiary of Hewlett-Packard Company.

Microsoft®, Windows®, and Windows NT® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Hewlett-Packard Company shall not be liable for technical or editorial errors or omissions contained herein. The information is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for Hewlett-Packard Company products are set forth in the express limited warranty statements for such products. Nothing herein should be construed as constituting an additional warranty.

Printed in the U.S.A.

SSSU V3.0 for Enterprise Virtual Array Reference Guide
Third Edition (May 2003)
Part Number: AA-RU5HA-TE

About this Guide	13
Intended Audience	14
Related Documentation	15
Conventions	16
Document Conventions	16
Text Symbols	16
Getting Help	18
HP Technical Support	18
HP Storage Website	18
HP Authorized Reseller	18
1 Introduction	19
Description	20
Formatting and Output of Returned Data	20
Installing the Storage System Scripting Utility	21
Starting the Storage System Scripting Utility	22
Syntax	22
Examples	22
Understanding Paths and Naming	23
Hosts Examples	23
Virtual Disk Example	24
ADD VDISK “\Virtual Disks\engineering\gene_research” SIZE=2	24
Disk Groups Example	24
Getting Help	25
2 SSSU Command Reference	27
Command Introduction	28
ADD	29
ADD COPY	29
Syntax	29

Switches.....	29
DISK_GROUP=.....	29
OS_UNIT_ID=.....	30
VDISK=.....	30
WORLD_WIDE_LUN_NAME=.....	30
WAIT_FOR_COMPLETION.....	30
NOWAIT_FOR_COMPLETION.....	30
Required Switch.....	30
Examples.....	31
ADD DR_GROUP.....	31
Syntax.....	31
Switches.....	31
COMMENT=.....	31
DESTINATION_SYSTEM=.....	32
DESTINATION_DISK_GROUP=.....	32
DESTINATION_VDISK_NAME=.....	32
VDISK=.....	32
Required Switches.....	32
Example.....	32
ADD DISK_GROUP.....	32
Syntax.....	33
Switches.....	33
COMMENT=.....	33
DEVICE_COUNT=.....	33
OCCUPANCY_ALARM=.....	33
SPARE_POLICY=.....	34
Example.....	34
ADD FOLDER.....	34
Syntax.....	35
Switch.....	35
COMMENT=.....	35
Examples.....	35
ADD HOST.....	36
Syntax.....	36
Switches.....	36
COMMENT=.....	36
OPERATING_SYSTEM=.....	36
WORLD_WIDE_NAME=.....	37

Required Switch	37
Example	37
ADD LUN	37
Syntax	37
Switches	37
HOST=	37
VDISK=	37
Required Switches	38
Examples	38
ADD SNAPSHOT	38
Syntax	38
Switches	39
ALLOCATION_POLICY=	39
OS_UNIT_ID=	39
VDISK=	39
WORLD_WIDE_LUN_NAME=	39
Required Switch	40
Examples	40
ADD SYSTEM	40
Syntax	41
Switches	41
COMMENT=	41
CONSOLE_LUN_ID=	41
DEVICE_COUNT=	41
SPARE_POLICY=	42
Example	42
ADD VDISK	42
Syntax	42
Switches	43
COMMENT=	43
DISK_GROUP=	43
MIRRORED_WRITEBACK	44
NOMIRRORED_WRITEBACK	44
OS_UNIT_ID=	44
PREFERRED_PATH=	44
NOPREFERRED_PATH	44
READ_CACHE	44
NOREAD_CACHE	45

REDUNDANCY=	45
SIZE=	45
WORLD_WIDE_LUN_NAME=	45
WRITE_PROTECT	45
NOWRITE_PROTECT	45
WAIT_FOR_COMPLETION.	46
NOWAIT_FOR_COMPLETION.	46
Required Switch	46
Switch Defaults	46
Examples	47
CAPTURE CONFIGURATION	48
Syntax	48
Step 1 Scripts	48
Step1A script.	48
Step1B Script	48
Step1C Script	49
Step2 Script	49
Step3 script	49
Restoring Configurations.	49
Example	49
CAPTURE VALIDATE	50
Syntax	50
Example	50
DELETE	51
DELETE DR_GROUP	51
Syntax	51
Example	51
DELETE DISK_GROUP	51
Syntax	51
Switches.	51
Example	52
DELETE FOLDER	52
Syntax	52
Switches.	52
Examples	52
DELETE HOST	52
Syntax	52
Switches.	53

Example	53
DELETE LUN	53
Syntax	53
Switches	53
Example	53
DELETE SYSTEM	53
Syntax	53
Switches	54
Example	54
DELETE VDISK	54
Syntax	54
Switches	54
WAIT_FOR_COMPLETION	54
NOWAIT_FOR_COMPLETION	54
Examples	55
EXIT	56
Syntax	56
Example	56
FILE	57
Syntax	57
Examples	57
HELP	58
Syntax	58
Examples	58
RESTART	59
Syntax	59
Switches	59
ALL_PEERS	59
NOALL_PEERS	59
Examples	59
SELECT	60
SELECT SYSTEM	60
Syntax	60
Examples	60
SELECT MANAGER	60
Syntax	60
Switches	60
PASSWORD=.	61

USERNAME=	61
Required Switches	61
Example	61
SET	62
SET DR_GROUP	62
Syntax	62
Switches	62
ADD_VDISK=	63
COMMENT=	63
DELETE_VDISK=	63
DETACH_VDISK=	63
FAILOVER	63
FAILSAFE	63
NOFAILSAFE	64
NAME=	64
SUSPEND	64
NOSUSPEND	64
Example	64
SET DISK	64
Syntax	64
Switches	64
COMMENT=	65
NAME=	65
Example	65
SET DISK_GROUP	65
Syntax	65
Switches	65
ADD=	65
COMMENT=	65
DELETE=	66
NAME=	66
OCCUPANCY_ALARM=	66
Example	66
SET FOLDER	66
Syntax	66
Switches	66
COMMENT=	66
NAME=	67

Example	67
SET HOST	67
Syntax	67
Switches.	67
ADD_WORLD_WIDE_NAME=.	67
DELETE_WORLD_WIDE_NAME=.	67
COMMENT=.	68
OPERATING_SYSTEM=.	68
Examples	68
SET MONITOR.	68
Syntax	68
Switches.	69
COMMENT=.	69
NAME=.	69
Example	69
SET OPTIONS.	69
Syntax	69
Switches.	70
COMMAND_DELAY=.	70
NOCOMMAND_DELAY	70
DISPLAY_WIDTH=.	70
ON_ERROR=.	70
RETRIES=.	71
NORETRIES.	71
SAFE_DELETE	71
NOSAFE_DELETE	71
Examples	72
SET SYSTEM	73
Syntax	73
Switches.	73
COMMENT=.	73
CONSOLE_LUN_ID=.	73
NAME=.	73
Example	73
SET VDISK	74
Syntax	74
Switches.	74
COMMENT=.	74

MIRRORED_WRITEBACK	74
NOMIRRORED_WRITEBACK	75
NAME=.....	75
OS_UNIT_ID=.....	75
PREFERRED_PATH=.....	75
NOPREFERRED_PATH	75
READ_CACHE	75
NOREAD_CACHE	76
SIZE=	76
WORLD_WIDE_LUN_NAME=.....	76
WRITE_PROTECT	76
NOWRITE_PROTECT	76
Example.....	76
SHOW.....	77
SHOW DR_GROUP	77
Syntax	77
Switches.....	77
FULL.....	77
Example.....	77
SHOW SYSTEM.....	77
Syntax	77
Switch	78
FULL.....	78
Examples	78
SHOW DISK	79
Syntax	79
Switch	79
FULL.....	79
Examples	79
SHOW DISK_GROUP	79
Syntax	79
Switch	80
FULL.....	80
Examples	80
SHOW HOST.....	80
Syntax	80
Switch	80
FULL.....	80

Examples	80
SHOW LUN	81
Syntax	81
Switch	81
FULL	81
Examples	81
SHOW MANAGER	82
Syntax	82
Switch	82
Example	82
SHOW MONITOR	82
Syntax	82
Switch	82
FULL	82
Examples	82
SHOW OPTIONS	83
Syntax	83
Example	83
SHOW POWER	83
Syntax	83
Switch	83
FULL	84
Examples	84
SHOW VDISK	84
Syntax	84
Switch	84
FULL	84
Examples	84
SHOW WORLD_WIDE_NAME	85
Syntax	85
Example	85
SHUTDOWN	86
Syntax	86
Switches	86
ALL_PEERS	86
NOALL_PEERS	86
Examples	86

A Configuration Examples87
Creating a Simple Configuration	88
Creating the Storage System	88
Creating and Presenting a Virtual Disk	90
Sample Configuration Command Output	91
Using the SHOW Commands	92
SHOW SYSTEM	92
SHOW VDISK	93
SHOW HOST	94
SHOW LUN	95
 Glossary97
 Index107
 Tables	
1 Document Conventions	16
2 Definition of Terms	20



about this guide

This reference guide describes the HP StorageWorks Storage System Scripting Utility (SSSU) commands.

“About this Guide” topics include:

- [Intended Audience](#), page 14
- [Related Documentation](#), page 15
- [Conventions](#), page 16
- [Getting Help](#), page 18

Intended Audience

This guide is intended for storage administrators, who are experienced with the following:

- Enterprise Virtual Array V3.0
- Configuration of SAN fabrics
- Windows 2000, Windows NT, Sun Solaris, OpenVMS, Tru64 UNIX, HP-UX, and IBM AIX operating systems

Related Documentation

This section lists documents that provide additional information:

- *HP SANworks Host Operating System V3.0 Kit for Enterprise Virtual Array Installation and Configuration Guide* (available for Sun Solaris, IBM AIX, Windows, Tru64 UNIX, OpenVMS, and HP-UX)
- *HP SANworks Host Operating System V3.0 Kit for Enterprise Virtual Array Release Notes* (available for Sun Solaris, IBM AIX, Windows, Tru64 UNIX, OpenVMS, and HP-UX)
- *HP StorageWorks Enterprise Virtual Array Users Guide*
- *HP StorageWorks Enterprise Virtual Array - How to Get More Information*
- *HP StorageWorks Enterprise Virtual Array Release Notes*
- *HP StorageWorks Enterprise Virtual Array Storage System Read Me First*
- *HP StorageWorks Enterprise Virtual Array Storage System World Wide Name Label*
- *HP StorageWorks Interactive Help for command view eva*
- *HP StorageWorks Enterprise Virtual Array Configuration Guide*
- *HP StorageWorks Enterprise Virtual Array Upgrade Instructions*
- *HP StorageWorks Enterprise Virtual Array License Instructions*
- *HP StorageWorks Enterprise Virtual Array Catalog of Associated Documentation*
- *HP StorageWorks Enterprise Virtual Array Storage System Installation Instructions*
- *HP StorageWorks Heterogeneous Open SAN Design Reference Guide*

Conventions

Conventions consist of the following:

- [Document Conventions](#)
- [Text Symbols](#)
- [Getting Help](#)

Document Conventions

The document conventions included in [Table 1](#) apply in most cases.

Table 1: Document Conventions

Element	Convention
Cross-reference links	Blue text: Figure 1
Key and field names, menu items, buttons, and dialog box titles	Bold
File names, application names, and text emphasis	<i>Italics</i>
User input, command and directory names, and system responses (output and messages)	Monospace font COMMAND NAMES are uppercase monospace font unless they are case sensitive
Variables	<monospace, italic font>
Website addresses	Blue, underlined sans serif font text: http://www.hp.com

Text Symbols

The following symbols may be found in the text of this guide. They have the following meanings.



WARNING: Text set off in this manner indicates that failure to follow directions in the warning could result in bodily harm or death.



Caution: Text set off in this manner indicates that failure to follow directions could result in damage to equipment or data.

Note: Text set off in this manner presents commentary, sidelights, or interesting points of information.

Getting Help

If you still have a question after reading this guide, contact an HP authorized service provider or access our website: <http://www.hp.com>.

HP Technical Support

Telephone numbers for worldwide technical support are listed on the following HP website: <http://www.hp.com/support/>. From this website, select the country of origin.

Note: For continuous quality improvement, calls may be recorded or monitored.

Be sure to have the following information available before calling:

- Technical support registration number (if applicable)
- Product serial numbers
- Product model names and numbers
- Applicable error messages
- Operating system type and revision level
- Detailed, specific questions

HP Storage Website

The HP website has the latest information on this product, as well as the latest drivers. Access storage at: <http://www.hp.com/country/us/eng/prodserv/storage.html>. From this website, select the appropriate product or solution.

HP Authorized Reseller

For the name of your nearest HP authorized reseller:

- In the United States, call 1-800-345-1518
- In Canada, call 1-800-263-5868
- Elsewhere, see the HP website for locations and telephone numbers: <http://www.hp.com>.

Introduction



This chapter introduces the HP StorageWorks Enterprise Storage System Scripting Utility (SSSU) V3.0 for the HP StorageWorks Enterprise Virtual Array. The following topics are covered:

- [Description](#), page 20
- [Installing the Storage System Scripting Utility](#), page 21
- [Starting the Storage System Scripting Utility](#), page 22
- [Understanding Paths and Naming](#), page 23
- [Getting Help](#), page 25

Description

The Storage System Scripting Utility (SSSU) for Enterprise Virtual Array is a command line application that allows you to configure and control HSV controllers.

You can execute configuration requests using the command view `eva` or `SSSU`. The command view `eva` is the graphical user interface that allows you to control and monitor a storage system. Use command view `eva` to handle simple or initial configuration tasks easily and expediently. Use `SSSU` to script and run repetitious and complex configuration tasks.

Formatting and Output of Returned Data

All returned `SHOW` command data is formatted in the form of:

```
XMLtag : Data
```

The XML tag displays before the data, and then the tag's data displays. No additional formatting is available, except where noted in the command reference (see [Chapter 2](#)).

Installing the Storage System Scripting Utility

SSSU is installed from your Enterprise Virtual Array host operating system kits. Refer to Chapter 2 in your Enterprise Virtual Array Installation and Configuration Guide for detailed instructions for installing SSSU.

Once installed, SSSU can run as a simple executable. For convenience, you can add the directory containing the SSSU executable to your path, or copy the executable to a directory already in your path. Ensure that the SSSU executable's attributes are set with the correct security and execution flags appropriate for your environment.

The executable file is named *sssu* or *SSSU.EXE*, depending on the operating system environment.

Starting the Storage System Scripting Utility

SSSU starts at a command prompt window, shell, or equivalent.

Note: The [SET OPTIONS](#) command on page 69 controls how SSSU behaves. The options you set with the SET OPTIONS command are in effect for the current session only. Each time you start SSSU, the default options are reinstated.

Note: The [CAPTURE CONFIGURATION](#) on page 48 is the only way to capture, save, and recreate a storage system's configuration. After you have successfully created a storage system, use the CAPTURE CONFIGURATION command to create scripts that you can use to recreate a storage system in the event of failure, or to create an exact replica.

Syntax

SSSU <additional arguments>

If SSSU is started without arguments, a generic application `NoSystemSelected>` prompt is displayed on the terminal and input can be accepted.

If SSSU is started with arguments, those commands are echoed to the input terminal and executed, and then the utility exits.

Note: You must enclose in double quotes any commands or object names that have embedded blanks (spaces).

Examples

```
SSSU
```

```
SSSU "FILE snapD1.txt"
```

The first example starts SSSU without additional arguments and prompts you for commands.

The second example starts SSSU and then executes the file `snapD1.txt` from the current directory.

Understanding Paths and Naming

This section describes important information about paths and naming conventions with SSSU.

Note: For detailed descriptions of the following commands, refer to [Chapter 2](#).

The following important root folders let you organize your storage system:

- “\Hosts\”
- “\Virtual Disks\”
- “\Disk Groups\”
- “\Data Replication\”
- “\Hardware\” - This folder exists within SSSU, however you cannot create any objects within this folder.

Note: You must use double quotes (“”) around names because of the space within the name.

You cannot create root-level folders. However, you can create additional folders within the \Hosts\ and “\Virtual Disks\” folders to organize your storage system (see “[ADD FOLDER](#)” on page 34).

Note: SSSU requires that you qualify specified names with full paths. However, if you are using the default placement when adding Hosts and Virtual Disks, you do not need to include the full path.

Hosts Examples

For example, to add a host named *engineering* to the root Hosts folder:

```
ADD HOST engineering WORLD=1111-2222-3333-4444
```

Whenever you refer to this host, you must give the full path. For example, when adding a LUN to this host:

```
ADD LUN 4 HOST=\Hosts\engineering VDISK="\Virtual Disks\Yekao\ACTIVE"
```

If you create a folder structure within the root Hosts, you must include all levels of the folder structure in your commands.

Here is an example of how to add the *engineering* host to the *resources* folder within the Host root folder:

```
ADD HOST \Hosts\resources\engineering WORLD=1111-2222-3333-4444
```

Virtual Disk Example

Because of the space in the root name for the Virtual Disks folder, you must always enclose strings that include the name of this folder within double quotes (“”).

For example, to add a virtual disk family *new_code* to the root folder Virtual Disks:

```
ADD VDISK "\Virtual Disks\new_code" SIZE=10
```

This example creates a 10 GB virtual disk family in the default disk group *new_code*.

Note: This is a virtual disk family. The actual virtual disk name (the active virtual disk) is *new_code\ACTIVE*.

Whenever you refer to this virtual disk, you must always give the full path and enclose it in double quotes. For example, when you reference this same virtual disk (*new_code*) within another command, give the full path, add the *ACTIVE*, and be sure to include everything within double quotes:

```
SET VDISK "\Virtual Disks\new_code\ACTIVE" WRITE_PROTECT
```

If you create a virtual disk (ADD VDISK) within a deeper folder structure, you need to include this full path within double quotes.

```
ADD VDISK "\Virtual Disks\engineering\gene_research" SIZE=2
```

Disk Groups Example

When a path includes the root folder “\Disk Groups\” you must enclose the entire path in double quotes:

```
SHOW DISK_GROUP "\Disk Groups\admin"
```

```
ADD VDISK "\Virtual Disks\new_code" SIZE=10 DISK_GROUP="\Disk Groups\admin"
```

Note: Although this guide shows commands spanning more than one line, always enter the SSSU command on one line unless specified to do otherwise.

Getting Help

Use the `HELP` command to display command syntax. See “[HELP](#)” on page 58 for detailed information.

This version of SSSU contains the following changes:

- `SYSTEM` replaces the `CELL` object type. For example, `ADD SYSTEM` and `SHOW SYSTEM` replace `ADD CELL` and `SHOW CELL`.
- The prompt `NoCellSelected>` becomes `NoSystemSelected>`.
- `DISK_GROUP` replaces the `GROUP` object type. For example, `ADD DISK_GROUP` replaces `ADD GROUP`.
- `VDISK` replaces the `STORAGE` object type. For example, `ADD VDISK` replaces `ADD STORAGE`.

All three of these older names (`CELL`, `GROUP`, and `STORAGE`) still work, as HP does not break existing scripts, but the old names do not show up in any help. For example, `CELL`, `GROUP` or `STORAGE` do not display in the list generated by `ADD ?`.

SSSU Command Reference

2

This chapter defines all the commands available in SSSU, including:

- [ADD](#), page 29
- [CAPTURE CONFIGURATION](#), page 48
- [DELETE](#), page 51
- [EXIT](#), page 56
- [FILE](#), page 57
- [HELP](#), page 58
- [RESTART](#), page 59
- [SELECT](#), page 60
- [SET](#), page 62
- [SHOW](#), page 77
- [SHUTDOWN](#), page 86

In addition, this chapter includes syntax and examples. The commands are presented in alphabetical order.

Note: Although this guide shows commands spanning more than one line, always enter the SSSU command on one line unless specified to do otherwise.

Command Introduction

Most of the commands have subcommands or switches that you can also use. The generic term “options” is used for both subcommands and switches in the utility’s help. These options are described under the entry for the command in this chapter.

You can display a list of options for each command directly within the interface by entering a “?” after the command or option name. For example, “ADD ?” displays all of the subcommands available with the ADD command, and “ADD SYSTEM ?” displays the available switches for the ADD SYSTEM command (see “[Getting Help](#)” on page 25).

Remember when you issue commands:

- Use the full path to qualify specified names
- If a path name contains a space, enclose the entire name in double quotes (“”)

For a full discussion of these requirements, see “[Understanding Paths and Naming](#)” on page 23.

[Appendix A](#), “[Configuration Examples](#),” provides a simple configuration example for using these commands to create and present a single LUN to a host. It also includes sample output from SHOW commands.

ADD

Use the ADD commands to create new systems, folders, disk_groups, dr_groups, virtual disks, copies, snapshots, as well as to add LUNs and hosts within the Enterprise Virtual Array.

Note: Creating copies, dr_groups, and snapshots of virtual disks is dependent on licensing level. See the documentation that came with your hardware for more information.

ADD COPY

Use the ADD COPY command to create a copy of the specified virtual disk. A copy is a new virtual disk family. The ADD COPY command is equivalent to creating a snapclone within command view eva.

Syntax

```
ADD COPY <copy_name> VDISK=<vdisk_family>
```

Switches

The following switches are available with the ADD COPY command:

- DISK_GROUP=
- OS_UNIT_ID=
- STORAGE_TO_COPY
- VDISK=
- WORLD_WIDE_LUN_NAME=
- WAIT_FOR_COMPLETION
- NOWAIT_FOR_COMPLETION

These switches are described in the following paragraphs.

DISK_GROUP=

The name of the disk group where you want to create the virtual disk copy. The disk group must already exist to use this switch. If not specified, SSSU uses the same group as the source VDISK.

Note: You must specify the complete path from the root directory for the DISK_GROUP and enclose the entire string in double quotes.

OS_UNIT_ID=

The ID that is presented to the host operating system. If set to zero, no ID is presented to the host.

Note: This is a required field for OpenVMS and is recommended for use with Tru64 UNIX. It is optional for all other operating systems.

VDISK=

The virtual disk name to be copied.

Note: You must specify the complete path from the root directory for the VDISK and enclose the entire string in double quotes.

WORLD_WIDE_LUN_NAME=

Sets the World Wide LUN Name on unrepresented storage. This switch is rejected when issued against a VDISK that is presented to a host.

WAIT_FOR_COMPLETION

Waits until the command's operation completes before displaying the next SSSU prompt (and returning control) or before running another script line. For large (greater than 1 TB) virtual disks, using this option can result in a long period before the prompt displays.

NOWAIT_FOR_COMPLETION

Default. Returns control as soon as possible. Some operations are invalid until the initial operation completes in the background.

Required Switch

The following switch is required with the ADD COPY command:

- VDISK=

Examples

```
ADD COPY wednesday_nite VDISK= "\Virtual Disks\payroll\ACTIVE"  
ADD COPY save_reports VDISK= "\Virtual Disks\daily_biz\ACTIVE"  
DISK_GROUP= "\Disk Groups\small_disks"
```

The first example copies the virtual disk *payroll* to a copy *wednesday_nite*.

The second example creates a copy of *daily_biz* as *save_reports* within the specified disk group.

Note: When you use the ADD COPY command, you are actually creating a VDISK. If you want to delete the VDISK, you must use the DELETE VDISK command.

ADD DR_GROUP

Use the ADD DR_GROUP command to create a data replication group containing the specified source virtual disk. This source VDisk will be replicated on the specified destination system.

Syntax

```
ADD DR_GROUP <dr_group_name>
```

Switches

- COMMENT=
- DESTINATION_SYSTEM=
- DESTINATION_DISK_GROUP=
- DESTINATION_VDISK_NAME=
- VDISK=

These switches are described in the following paragraphs.

COMMENT=

Used to attach comments to the DR Group being created. This string must be enclosed in quotes. The maximum number of characters for this optional switch is 64.

DESTINATION_SYSTEM=

The system on which the destination VDISKS are created on linked systems within this DR Group.

DESTINATION_DISK_GROUP=

The disk group name in which the VDISK on the destination system is created.

DESTINATION_VDISK_NAME=

The name of the VDISK that is created on the destination system. The default name is the same as that of the source.

VDISK=

The name of a source system VDISK to be placed within the DR Group. A destination VDISK is automatically created on the destination system.

Required Switches

The following switches are required for the ADD DR_GROUP command:

- DESTINATION_SYSTEM =
- VDISK=

Example

```
ADD DR_GROUP Transactions DESTINATION_SYSTEM=Makawao  
VDISK="/Virtual Disks/Hawaii/ACTIVE"
```

This example creates a DR_GROUP named *Transactions* that contains the source virtual disk "*Virtual Disks\Hawaii\ACTIVE*." This DR_GROUP will be connected to the destination system *Makawao*, on which the destination virtual disk "*Virtual Disks\Hawaii\ACTIVE*" (same as the source by default) will be created and added to the DR_GROUP destination.

ADD DISK_GROUP

Use the ADD DISK_GROUP command to add a disk group to the system. It is best practice to use multiple disk groups because they provide additional failure isolation.

If all the disks are in one group, any disk failure could cause all the virtual disks to run in degraded mode until the data on the disk that was lost is recreated on other disks. By creating multiple disk groups, a disk failure only affects the virtual disks in that one group.

Syntax

```
ADD DISK_GROUP <group_name>
```

Switches

The following switches are available with the ADD DISK_GROUP command:

- COMMENT=
- DEVICE_COUNT=
- OCCUPANCY_ALARM=
- SPARE_POLICY=

These switches are described in the following paragraphs.

COMMENT=

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed and the string must be enclosed in double quotes.

DEVICE_COUNT=

The number of physical disks to use for this group. The limit is the number of available disks in the storage system. The default and minimum value is 8.

OCCUPANCY_ALARM=

The point when a defined percentage of space is used. When this point is reached, an event log is generated and sent to the Management Appliance (optionally the host system) informing the administrator that the group is becoming full. Do not use the percent sign (%) after the number.

SPARE_POLICY=

This parameter determines the amount, if any, of storage space set aside for use in the event disks fail.



Caution: Vraid1 and Vraid5 virtual disks families that have a SINGLE or DOUBLE group spare policy ensure that there is adequate space to regenerate data. If you set the spare policy to NONE, the disk group may not have enough available space to regenerate Vraid1 and Vraid5 data.

- NONE—reserves no space within a disk group to allow for data reconstruction for failure of disk drives
- SINGLE—reserves space within a disk group to allow for data reconstruction for failure of a single disk drive
- DOUBLE—reserves space within a disk group to allow for data reconstruction for failure of two disk drives

Note: The space set aside is not in numbers of physical disks. It is the equivalent amount of storage space spread across all disks.

Example

```
ADD DISK_GROUP "\Disk Groups\human_resources" DEVICE_COUNT=12
SPARE_POLICY=SINGLE OCCUPANCY_ALARM=75
```

This example creates a new disk group *human_resources*. It has 12 physical disks with the equivalent of one disk set aside as a spare and sends an event log to the Management Appliance when 75% of capacity is reached.

ADD FOLDER

Use the ADD FOLDER command to create a new folder within the specified folder to aid in organizing your storage system. You can only create folders under the “Virtual Disks” and “Hosts” root folders. You cannot create root folders.

For example, if you have a controller that is serving *HR* and *Engineering*, you could create four folders—two to separate the virtual disks and two to separate the hosts:

- “Virtual Disks\Engineering” and “Virtual Disks\HR”
- \Hosts\Engineering and \Hosts\HR

Creating these folders allows you to put engineering virtual disks and hosts in the Engineering folders and Human Resources storage/hosts in the HR folders. This makes it easier to keep track of the pieces within your storage system. If you want to nest folders inside folders, you must add one folder at a time.

Syntax

```
ADD FOLDER <folder_name>
```

Switch

Only the COMMENT switch is available with the ADD FOLDER command.

COMMENT=

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed and the string must be enclosed in double quotes.

Examples

```
ADD FOLDER \Hosts\human_resources
```

```
ADD FOLDER "\Virtual Disks\colorado\colorado springs\engineering"
```

The first example creates a new folder *human_resources* within the root folder *Hosts*. The second example creates a folder *engineering* in the folder *colorado springs*, which is two levels deep from the root folder *Virtual Disks* ("*Virtual Disks\colorado\colorado springs*").

The folder structure in the second example must already have been in place before creating the new folder *engineering*. Also note that the entire path is enclosed in double quotes because of the space in the root folder name (*Virtual Disks*).

ADD HOST

Use the ADD HOST command to add a host and its World Wide Name (WWN) to the list of hosts that can connect to virtual disks within the current system.

The ADD HOST command adds the first Fibre Channel Adapter (FCA) only. The [SET HOST](#) command (see page 67) adds each subsequent FCA.

Syntax

```
ADD HOST <host_name> WORLD_WIDE_NAME=<world_wide_name>
```

Switches

The following switches are available with the ADD HOST command:

- COMMENT=
- OPERATING_SYSTEM=
- WORLD_WIDE_NAME=

These switches are described in the following paragraphs.

COMMENT=

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed and the string must be enclosed in double quotes.

OPERATING_SYSTEM=

The type of operating system for the specified host. Choose one of the following operating systems. (they are spelled here as SSSU expects):

- HPUX
- IBMAIX
- OPEN_VMS
- SOLARIS
- TRU64
- WINDOWS
- CUSTOM
- UNKNOWN

If unspecified, the operating system defaults to Windows.

WORLD_WIDE_NAME=

The World Wide Name of the host port.

Required Switch

The following switch must be set to ensure that command view eva recognizes the host added with the ADD HOST command.

- WORLD_WIDE_NAME

Example

```
ADD HOST \Hosts\development WORLD_WIDE_NAME=5000-1fe1-ff00-0000
```

This example adds a host *development* with an adapter at a WWN of *5000-1fe1-ff00-0000*.

ADD LUN

The ADD LUN command makes previously created VDISKs available to a host.

Syntax

```
ADD LUN <LUN_number> VDISK=<storage_name> HOST=<host_name>
```

LUN_number is 1 through 255.

Switches

The following switches are available with the ADD LUN command:

- HOST=
- VDISK=

These switches are described in the following paragraphs.

HOST=

The name of the host to which the LUN is presented.

VDISK=

The name of the VDISK that is presented to the host.

Note: You must specify the complete path from the root directory to both VDISK and HOST. In the case of VDISK, you must enclose the entire string in double quotes.

Required Switches

Set the following switches so that command view eva recognizes the LUN you add with the ADD LUN command.

- HOST
- VDISK

Examples

```
ADD LUN 12 VDISK= "\Virtual Disks\act\payroll\ACTIVE" HOST= \Hosts\sanfran
ADD LUN 175 VDISK= "\Virtual Disks\user_disk\ACTIVE" HOST= \Hosts\corporate
```

The first example adds LUN 12, which presents the ACTIVE virtual disk for the family *payroll* to the host called *sanfran*.

The second example adds LUN 175, which presents the ACTIVE virtual disk for the family *user_disk* to host *corporate*.

ADD SNAPSHOT

The ADD SNAPSHOT command creates a duplicate of a virtual disk family's active virtual disk.

Note: Creating snapshots of virtual disks depends on your licensing level. Attempting to add a snapshot without the license returns an error message. This message indicates that you need to enter your licensing information in the command view eva. You must have a registered license.

Syntax

```
ADD SNAPSHOT <snapshot_name> VDISK=<vdisk_to_snapshot>
```

Switches

The following switches are available with the ADD SNAPSHOT command.

- ALLOCATION_POLICY=
- OS_UNIT_ID
- VDISK=
- WORLD_WIDE_LUN_NAME=

These switches are described in the following paragraphs.

ALLOCATION_POLICY=

Indicates how the space for the snapshot is allocated.

- DEMAND—Storage is allocated for the snapshot only when required. As the snapshot and the original storage's information diverges, the space allocated for the snapshot increases.
- FULLY—All storage required to fully contain a snapshot is allocated when the snapshot is created. This is the conservative method to create a snapshot, because adequate space is guaranteed for a full snapshot. This is the default.

OS_UNIT_ID=

The ID that is presented to the host operating system. If set to zero, no ID is presented to the host.

Note: This is a required field for OpenVMS and is recommended for use with Tru64 UNIX. It is optional for all other operating systems.

VDISK=

The source virtual disk for this snapshot.

Note: You must specify the complete path from the root directory to the VDISK and enclose the entire string in double quotes.

WORLD_WIDE_LUN_NAME=

Sets the World Wide LUN Name.

Required Switch

The following switch is required with the ADD SNAPSHOT command:

- VDISK=

Examples

```
ADD SNAPSHOT payroll_backup VDISK="\Virtual Disks\payroll\ACTIVE"  
ALLOCATION_POLICY=DEMAND
```

```
ADD SNAPSHOT wed_nite_biz VDISK="\Virtual Disks\daily_biz\ACTIVE"  
ALLOCATION_POLICY=FULLY
```

The first example creates a snapshot *payroll_backup*, from the ACTIVE partition of Vdisk family *payroll* that uses capacity only as needed.

The second example creates a snapshot *wed_night_biz*, from the ACTIVE partition of the family *daily_biz* while reserving all capacity necessary to create the snapshot.

Note: To delete snapshots created by ADD SNAPSHOT, use the DELETE VDISK command.

ADD SYSTEM

Use the ADD SYSTEM command to initialize an uninitialized storage system and to create a default disk group. You must select an uninitialized storage system before issuing the ADD SYSTEM command. If the system is already initialized, the command is rejected.

Attempting to add (create) a storage system without the license returns an error message. This message indicates that you need to enter your licensing information in the command view *eva*. You must have a license and follow the registration process.

Use the SHOW SYSTEM command (see “[SHOW SYSTEM](#)” on page 77) to display the names of the available uninitialized storage systems. The uninitialized storage system display as: *Uninitialized Storage System#>*, in which # represents a number. After initialization is complete, SSSU changes its default prompt back to *NoSystemSelected>*.

Because the system’s name changed from the uninitialized string to the name given with the ADD SYSTEM command, you must reselect the system. Issue the SELECT SYSTEM command using the new name of the system.

Note: [Appendix A](#) includes a configuration example that explains the sequence of commands you can use to initialize an uninitialized storage system (see “[Creating a Simple Configuration](#)” on page 88).

Syntax

```
ADD SYSTEM <system_name>
```

Switches

The following switches are available with the ADD SYSTEM command:

- COMMENT=
- CONSOLE_LUN_ID=
- DEVICE_COUNT=
- SPARE_POLICY=

These switches are described in the following paragraphs.

COMMENT=

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed and the string must be enclosed in double quotes.

CONSOLE_LUN_ID=

The LUN used for console communication after system creation. If set to zero, no console LUN is presented to the host.

Note: This value is used for IBM AIX (set to zero), OpenVMS (required), and Tru64 UNIX (recommended). Other host operating systems ignore the value. See the host operating system installation guides for more information.

DEVICE_COUNT=

The number of physical disks to use for the default disk group of this system. The limit is the number of available disks in the storage system. The default and minimum value is 8.

SPARE_POLICY=

This parameter determines the amount, if any, of storage space set aside for safety in the event disks fail. The default SPARE_POLICY is SINGLE.

- NONE—reserves no space within a disk group to allow for data reconstruction for failure of disk drives
- SINGLE—reserves space within a disk group to allow for data reconstruction for failure of a single disk drive. This is the default.
- DOUBLE—reserves space within a disk group to allow for data reconstruction for failure of two disk drives

Note: The space set aside is not in numbers of physical disks. It is the equivalent amount of storage space spread across all disks.

Example

```
ADD SYSTEM payroll DEVICE_COUNT=12 SPARE_POLICY=SINGLE
```

This example creates an initialized system *payroll* with 12 physical disks and a spare policy of *SINGLE*.

ADD VDISK

Use the ADD VDISK command to create a virtual disk with a specified name and parameters.

Note: This command actually creates a virtual disk family and the ACTIVE partition. After you create the family, an initial virtual disk "ACTIVE" is placed as the only virtual disk in the family. This is referred to as the active virtual disk. When you later refer to this virtual disk, specify \ACTIVE after the family name. See the examples that follow the descriptions of the switches.

Syntax

```
ADD VDISK <storage_name> SIZE=<n>
```

The *n* is the virtual disk size in whole GBs.

Switches

The following switches are available with the ADD VDISK command:

- COMMENT=
- DISK_GROUP=
- MIRRORED_WRITEBACK
- NOMIRRORED_WRITEBACK
- OS_UNIT_ID=
- PREFERRED_PATH=
- NOPREFERRED_PATH
- READ_CACHE
- NOREAD_CACHE
- REDUNDANCY=
- SIZE=
- WORLD_WIDE_LUN_NAME=
- WRITE_PROTECT
- NOWRITE_PROTECT
- WAIT_FOR_COMPLETION
- NOWAIT_FOR_COMPLETION

These switches are described in the following paragraphs.

COMMENT=

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed and the string must be enclosed in double quotes.

DISK_GROUP=

The disk group name for where you want to create the virtual disk. The disk group must already exist to use this switch. If not specified, the default disk group is used.

Note: You must specify the complete path from the root directory to the DISK_GROUP and enclose the entire string in double quotes.

MIRRORED_WRITEBACK

All writes are stored in mirrored caches. After the data is stored in both caches, the write is complete. If one mirrored cache is unusable, the write is not complete until the data is in both caches.

NOMIRRORED_WRITEBACK

No writes are stored in mirrored caches. After the data is stored in one cache, the write is complete. The write is complete even if a mirror copy of cache is not available.

OS_UNIT_ID=

The ID that is presented to the host operating system. If set to zero, no ID is presented to the host.

Note: This is a required field for OpenVMS and is recommended for use with Tru64 UNIX. It is optional for all other operating systems.

PREFERRED_PATH=

The preferred controller path that is specified to handle all I/O for the virtual disks. If a controller fails, the path always reverts to the working controller.

- **PATH_A_BOTH**—Controller path A fails over to controller B. When controller A restarts, the virtual disks fail back to controller A. This is failover/failback mode.
- **PATH_A_FAILOVER**—Controller path A fails over to controller B. When controller A restarts, the virtual disks do not fail back over to controller A. This is failover only mode.
- **PATH_B_BOTH**—Controller path B fails over to controller A. When controller B restarts, the virtual disks fail back to controller B. This is failover/failback mode.
- **PATH_B_FAILOVER**—Controller path B fails over to controller A. When controller B restarts, the virtual disks do not fail back over to controller B. This is failover only mode.

NOPREFERRED_PATH

Allows the I/O to be handled by either controller.

READ_CACHE

Reads are satisfied from the controller's cache.

NOREAD_CACHE

Reads are always satisfied from the physical disks, not the controller's cache.

REDUNDANCY=

The amount of data protection set when creating the virtual disk. If not specified, the default is Vraid0.

- **Vraid0**—Provides no data protection. It distributes data among its member disks into stripes, and uses all members to process I/O requests. This method has no overhead associated with duplication of information and provides the highest performance.
- **Vraid1**—Provides the highest level of data protection but uses the most space. It duplicates data written to one disk onto a backup disk. In a multidisk configuration, Vraid1 mirrors each pair of disks to each other. These disk pairs can then be striped to create a virtual disk.
- **Vraid5**—Provides a moderate level of data protection. This method distributes the parity information among all disk members. If one drive fails, the failed disk can be recreated after it is replaced.

SIZE=

This is a required switch. Size of the storage to be created. You can specify size in whole gigabytes only, fractions are not allowed (1GB – 2000GB, limited by the actual amount of space available within the disk group).

WORLD_WIDE_LUN_NAME=

Sets the World Wide LUN Name.

Note: This switch is commonly used to allow a host to point to a new version of the virtual disk by giving the new virtual disk the same WWN as the old virtual disk.

WRITE_PROTECT

Does not allow writing to the virtual disk for all presented LUNs/host.

NOWRITE_PROTECT

Allows writing to the virtual disk for all presented LUNs/host.

WAIT_FOR_COMPLETION

Default. Waits until the command's operation completes before displaying the next SSSU prompt (and returning control) or before running another script line. For large (greater than 1 TB) virtual disks, using this option can result in a long period before the prompt displays.

NOWAIT_FOR_COMPLETION

Returns control as soon as possible. Some operations are invalid until the initial operation completes in the background. For example, you cannot present the Vdisk (add a LUN) until the command completes.

Required Switch

The following switch is required with the ADD VDISK command:

- SIZE=

Switch Defaults

The following defaults are used for the ADD VDISK command when nothing is specified:

- MIRRORED_WRITEBACK
- NOWRITE_PROTECT
- OS_UNIT_ID= defaults to zero
- PREFERRED_PATH= defaults to NOPREFERRED_PATH
- REDUNDANCY= defaults to Vraid0
- DISK_GROUP= *"/Disk Groups/Default Disk Group"*
- READ_CACHE

Examples

```
ADD VDISK "\Virtual Disk\scratch" SIZE=10 REDUNDANCY=VRAID5  
READ_CACHE
```

```
ADD VDISK "\Virtual Disks\engineering\gene_research" SIZE=2  
DISK_GROUP= "\Disk Groups\small_disks" MIRRORED_WRITEBACK
```

The first example creates a 10 GB virtual disk scratch within the default disk group using read cache and a Vraid5 redundancy level.

The second example creates a 2 GB disk at Vraid0 redundancy level in the small_disks group using mirrored writeback cache.

Note: Remember that this name is the <family_name> and that "*\Virtual Disks\<family_name>\ACTIVE*" is the name of the ACTIVE virtual disk. You must use this full path (enclosed in double quotes) when referring to it with any other commands.

CAPTURE CONFIGURATION

Syntax

```
CAPTURE CONFIGURATION <file_name>
```

CAPTURE CONFIGURATION queries the currently selected system and creates from one to five SSSU scripts. Not all systems require all five scripts. For example, step1A is always generated and may be the only file that is required. You use these scripts to recreate the original configuration and in some cases to assist in the recovery of any site.

The scripts output to the console unless you specify a file. The scripts generate five files. Given a file name, `_StepXX` is appended after the file name and before the extension. `XX` is the restore step name, which is 1A, 1B, 1C, 2 and 3. For example, specifying `CAPTURE CONFIGURATION newyear.txt` causes SSSU to create the files `newyear_Step1A.txt`, `newyear_Step1B.txt`, `newyear_Step1C.txt`, `newyear_Step2.txt` and `newyear_Step3.txt`.

SSSU checks to see if any files by the names to be generated exist. If so, SSSU prompts you to replace existing files with the new files or abort the command completely.

When creating files that contain the scripts, a progress bar displays on the console. This command may take a long time to complete, depending on the size of the configuration.

Note: Do not reconfigure the selected system while this command is executing.

Step 1 Scripts

The division of Step1 into three smaller steps facilitates data replication (DR) recovery.

Step1A script

Creates the storage system, disk groups, hosts, and VDISKS that are not used for data replication, either source or destination, and LUNs for the created disks. Step1A creates a basic system that does not include DR VDISKS or groups. CAPTURE CONFIG always creates this step.

Step1B Script

Creates all source VDISKS used in DR groups on this controller.

Step1C Script

Presents all source VDISKS (creates LUNS) used for DR groups to their hosts. This step exists to assist in recovering from a DR failure in which a source site was lost. LUNs can be presented in their original configuration by running the correct Step1C script.

Step2 Script

Recreates all DR-specific configuration information for which this system is the source. This involves the configuration's source DR_GROUPS and their members only. Presentations of remote VDISKS are not restored by this command, see Step3.

This step provides flexibility when a site is completely lost. You must run Step1A, Step1B, and Step1C on both source and destination systems before Step2 can be run.

Step3 script

Presents remote VDISKS (creates LUNS) used for DR groups to their hosts.

You must run Step2 on both source and destination systems before Step3 can be run.

Restoring Configurations

To restore system configuration from captured scripts, you must run the following scripts in the following order on each system that are part of the DR configuration:

1. Run scripts 1A, 1B, and 1C successfully against ALL systems in the DR configuration.
2. Run step2 script successfully against ALL systems in the DR configuration.
3. Run step3 script successfully against ALL systems in the DR configuration.

Once these scripts have been run against all systems, the DR configuration is restored to its original state.

Example

```
CAPTURE CONFIGURATION c:\sales.txt
```

This example creates the files *sales_Step1A.txt*, *sales_Step1B.txt*, *sales_Step1C.txt*, *sales_Step2.txt* and *sales_Step3.txt*.

CAPTURE VALIDATE

Syntax

```
CAPTURE VALIDATE <file_name>
```

CAPTURE VALIDATE checks the specified script file for modification by calculating its checksum and comparing the result against the checksum saved in the file. A script may have been modified (failed the CAPTURE VALIDATE command) but may still work. SSSU runs a script even if its checksum shows file modification.

Example

```
CAPTURE VALIDATE c:\reconfigure_sales_dr_step2.txt
```

This example determines whether the *reconfigure_sales_dr_step2.txt* file was modified.

DELETE

Use the DELETE commands to remove systems, folders, disk groups, DR groups, VDISKS, hosts, and LUNs from the storage system. You can control the behavior of all DELETE commands with the SET OPTIONS command (see “[SET OPTIONS](#)” on page 69 for details on how to use the SAFE_DELETE and NOSAFE_DELETE switches).

Note: All of the DELETE commands require the full path from the root directory to the object you want to delete.

DELETE DR_GROUP

Deletes the DR Group on the source and destination system. Be sure that VDISKS are unrepresented, that the source and destination are not communicating, and that the group is empty. The SET OPTION NOSAFE_DELETE does not override this requirement. You can delete a DR Group on the source system.

Syntax

```
DELETE DR_GROUP <dr_group_name>
```

Example

```
DELETE DR_GROUP "\\Data Replication\payroll"
```

This example deletes the DR_GROUP *payroll*.

DELETE DISK_GROUP

Use the DELETE DISK_GROUP command to delete a disk group from the configuration. The command is rejected if any virtual disks are present in the disk group.

Syntax

```
DELETE DISK_GROUP <group_name>
```

Switches

There are no switches available with the DELETE DISK_GROUP command.

Example

```
DELETE DISK_GROUP "\Disk Groups\human_resources"
```

This example deletes a disk group *human_resources*. Note the need for the double quotes because of the space in the Disk Groups folder name.

DELETE FOLDER

Use the DELETE FOLDER command to delete the specified folder. To delete multiple layers of folders, you must do so sequentially.

Note: Ensure folders are empty before deleting them, as the DELETE FOLDER command is rejected if there are contents inside the folder.

Syntax

```
DELETE FOLDER <folder_name>
```

Switches

There are no switches available with the DELETE FOLDER command.

Examples

```
DELETE FOLDER \Hosts\human_resources\  
DELETE FOLDER \Hosts\colorado\colorado_springs\engineering\  

```

The first example deletes a folder *human_resources* within the default Hosts folder. The second example deletes the folder *engineering* nested in the path *\Hosts\colorado\colorado_springs*.

DELETE HOST

Use the DELETE HOST command to remove the specified host from the list of hosts that connect to the EVA. The command is rejected if any virtual disks are presented to the specified host.

Syntax

```
DELETE HOST <host_name>
```

Switches

There are no switches available with the DELETE HOST command.

Example

```
DELETE HOST \Hosts\development
```

This example deletes the host *development*.

DELETE LUN

Use the DELETE LUN command to remove access to a virtual disk through the LUN from a host.

Syntax

```
DELETE LUN <LUN_name>
```

Switches

There are no switches available with the DELETE LUN command.

Example

```
DELETE LUN "\Hosts\accounting_department\12"
```

This example removes access to LUN 12 from the host *accounting_department*.

DELETE SYSTEM

Use the DELETE SYSTEM command to permanently remove the system from an EVA. The system will no longer be accessible, and all data will be lost.



Caution: The DELETE SYSTEM command removes the entire system configuration. All customer data on the entire system is lost when you run this command. In addition, all information about LUNs and hosts on this system is lost.

Syntax

```
DELETE SYSTEM <system_name>
```

Switches

There are no switches available with the DELETE SYSTEM command.

Example

```
DELETE SYSTEM payroll
```

This examples deletes the system *payroll*.

DELETE VDISK

Use the DELETE VDISK command to permanently remove the specified virtual disk. The virtual disk is no longer accessible, and all data is lost.



Caution: The DELETE VDISK command removes the entire virtual disk from the storage system. All customer data on the storage specified is destroyed when you run this command. In addition all information about the LUNs presented from this virtual disk is lost as well.

Syntax

```
DELETE VDISK <storage_name>
```

Switches

- WAIT_FOR_COMPLETION
- NOWAIT_FOR_COMPLETION

The following paragraphs describe these switches.

WAIT_FOR_COMPLETION

Waits until the command's operation completes before displaying the next SSSU prompt (and returning control) or before running another script line. For large (greater than 1 TB) virtual disks, using this option can result in a long period before the prompt displays.

NOWAIT_FOR_COMPLETION

Default. Returns control as soon as possible. Some operations are invalid until the initial operation completes in the background.

Examples

```
DELETE VDISK "\Virtual Disks\d12\ACTIVE" WAIT_FOR_COMPLETION
DELETE VDISK "\Virtual Disks\daily_biz\wed_nite_biz"
```

The first example deletes *ACTIVE* from the virtual disk family *d12*.

`WAIT_FOR_COMPLETION` prevents the SSSU prompt from displaying until the virtual disk deletion is completed.

Note: You cannot delete *ACTIVE* if a snapshot of the virtual disk exists. You must delete the snapshot, and then delete *ACTIVE*.

The second example deletes the snapshot *wed_nite_biz* from the family *daily_biz*.

EXIT

The EXIT command terminates the SSSU session.

If SSSU is accepting input from the terminal or the command line, EXIT causes the program to terminate.

If SSSU is processing a script from a file while encountering a FILE command, the behavior of the EXIT command depends on how the FILE command was issued:

- If the FILE command was issued from the command line, an EXIT command in the file causes SSSU to terminate.
- If the FILE command was issued from the terminal, an EXIT command in the file causes SSSU to return to accepting input from the terminal.
- If a file is currently executing as a result of another FILE command, an EXIT command in that file returns control to the previous file. SSSU does not run any commands in a script beyond the EXIT command. EXIT causes an immediate return to the “calling” layer.

Syntax

```
EXIT
```

Example

```
EXIT
```

SSSU terminates.

FILE

The FILE command causes the current mode of input to suspend, and redirects the scripting utility to accept input from the specified file. Either the end of the file or an EXIT command in the specified file causes SSSU to again accept input from the previous input source.

FILE commands can be nested. Nesting FILE commands means that a file being executed through a FILE command can have FILE commands within its command set. The only limitation on how deep FILE commands may be nested is based on the host system's resources.

Syntax

```
FILE <filename.ext>
```

The *ext* stands for extension. An extension is not required by SSSU, although you can choose one appropriate to your environment.

Examples

```
FILE snapd1.txt
```

```
FILE d:\scripts\snapshots\d27.txt
```

The first example executes the file *snapd1.txt* from the current directory. The second example executes a file *d27.txt* located in the specified directory.

HELP

The HELP command displays information on using the SSSU help.

To get HELP on specific command syntax enter a space and a question mark (?) wherever you would normally include a parameter, at any level of a command line. The help system will then list the options available for that parameter.

For detailed help, you may also have to enter a placeholder name instead of a real name in the command. For example, for detailed help on the ADD VDISK command, you may include a placeholder instead of a VDISK name as shown in the example. Keep in mind, though, that you can use a real object name if it exists.

Syntax

```
HELP
```

Examples

```
HELP
```

```
Command syntax help is available at any point within a command  
by typing a '?'
```

The first example accesses HELP.

```
Add VDISK placeholder REDUNDANCY= ?
```

```
The system displays:
```

```
The options are:
```

```
VRAID0  
VRAID1  
VRAID5
```

The second example requests the display of the different valid values for the switch REDUNDANCY. For this command, if you do not include the placeholder type of object name, all the options are not displayed.

RESTART

Use the RESTART command to restart a controller shown by the [SHOW POWER](#) command.

A power object is a controller.

Syntax

```
RESTART <controller_name>
```

Switches

The following switches are available for the RESTART command:

- ALL_PEERS
- NOALL_PEERS

These switches are described in the following paragraphs.

ALL_PEERS

Restarts all peer controllers (both controllers) on this system.

NOALL_PEERS

Default. Restarts only the specified controller.

Examples

```
RESTART "\Hardware\Rack 1\Enclosure 7\Controller B" ALL_PEERS
```

```
RESTART "\Hardware\Rack 1\Enclosure 7\Controller A" NOALL_PEERS
```

The first example restarts *Controller B*, including its peer controller.

The second example restarts *Controller A*, but not its peer controller.

SELECT

Use the SELECT command to select a system. After you select a system, the prompt changes to reflect the selected system.

SELECT SYSTEM

Use the SELECT SYSTEM command to direct the command prompt to the selected system. All configuration commands then affect the selected system. If the system name has spaces in it, the name must be surrounded by double quotes (“”).

Syntax

```
SELECT SYSTEM <system_name>
```

Examples

```
SELECT SYSTEM employees
SELECT SYSTEM "payroll storage"
```

The first example selects the system *employees* and returns the prompt *employees>*. The second example selects a system *payroll storage* and returns the prompt *payroll storage>*.

SELECT MANAGER

Use the SELECT MANAGER command to direct the command prompt to a selected command view *eva* or management appliance. All configuration commands run on the selected command view *eva* or management appliance and affect its storage system configuration. The *<manager_name>* can be a server, appliance name, or an IP address.

Syntax

```
SELECT MANAGER <manager_name> USERNAME=<username>
PASSWORD=<password>
```

Switches

The following switches are available with the SELECT MANAGER command:

- PASSWORD=
- USERNAME=

These switches are described in the following paragraphs.

PASSWORD=

Password associated with the username that you use to access the command view eva.

USERNAME=

Username that you use to access the command view eva.

Required Switches

The following switches are required with the SELECT MANAGER command:

- PASSWORD=
- USERNAME=

Example

```
SELECT MANAGER north_campus USERNAME=XXX PASSWORD=XXXX
```

This example selects the manager *north_campus*. The username and password for the manager are validated in the background on every command.

SET

Use the SET commands to set the properties of systems, disks, folders, DR groups, disk groups, hosts, and virtual disks. In addition, the SET commands let you reset existing storage settings and application options.

- All SET commands require the full path from the root directory to the object you want to rename.
- You can only perform one property change per SET command.

SET DR_GROUP

Use SET DR_GROUP to set the mode of operation of an existing DR Group.

Syntax

```
SET DR_GROUP <dr_group_name>
```

Switches

- ADD_VDISK=
- COMMENT=
- DELETE_VDISK=
- DETACH_VDISK=
- FAILOVER
- FAILSAFE
- NAME=
- NOFAILSAFE
- SUSPEND
- NOSUSPEND

The following paragraphs describe these switches.

ADD_VDISK=

Name of a source system VDISK to be added to the DR Group. A destination VDISK is automatically created on the destination system. When adding a VDISK to the DR Group, the following two optional switches may be specified:

DESTINATION_DISK_GROUP=

The disk group name on the destination system on which the VDISK is created. This switch can only be specified after the ADD_VDISK switch.

DESTINATION_VDISK_NAME=

The name of the VDISK that is created on the destination system. The default name is the same as that of the source. This optional switch can only be specified after the ADD_VDISK switch.

COMMENT=

An optional field for attaching comments to the DR Group being created. This string must be enclosed in quotes. The maximum number of characters for this switch is 64.

DELETE_VDISK=

Name of a source system VDISK to be removed from the DR Group. The corresponding destination VDISK on the destination system is removed from the DR Group and deleted from the system.

DETACH_VDISK=

Name of a source system VDISK to be removed from the DR Group. The corresponding destination VDISK on the destination system is removed from the DR Group but not deleted. The detached destination VDISK continues to exist as an independent VDISK.

FAILOVER

Reverses the roles of the DR Groups; the source becomes the destination and the destination becomes the source. This is used to both “fail over” to the destination site as well as to “fail back” when normal operation is to be resumed.

FAILSAFE

When the connection between the source and the destination system fails, all writes are immediately halted and none is reported as completing until either the connection is restored or NOFAILSAFE is set.

NOFAILSAFE

This is the default when you first create the DR Group. When the connection between the source and the destination system fails, writes are directed to the log until the connection is restored. At this point, the pending destination writes are pushed across to the destination system. If the log becomes full, when the connection is restored, the system makes a full copy of the source VDISK to the destination VDISK. Otherwise, the log resynchs the source and destination VDISKS.

NAME=

New name for the specified DR_GROUP.

SUSPEND

Stops replication of data from the source to the destination. This causes the same behavior as if the connection failed.

NOSUSPEND

Allows replication of data from the source to the destination to resume. This causes the same behavior as if the connection was restored.

Example

```
SET DR_GROUP "\Data Replication\Transactions" SUSPEND
```

This example suspends replication for the DR Group *Transactions*.

SET DISK

The SET DISK command changes the properties of a disk.

Syntax

```
SET DISK <name>
```

Switches

The following switches are available for the SET DISK command:

- COMMENT=
- NAME=

These switches are described in the following paragraphs.

COMMENT=

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed and the string must be enclosed in double quotes.

NAME=

New name to give to the specified disk.

Example

```
SET DISK "\Disk Groups\Ungrouped Disks\Disk 005" NAME=5Disk
```

This example renames the disk *Disk 005* to *5Disk*.

SET DISK_GROUP

The SET DISK_GROUP command changes the specified disk group properties.

Syntax

```
SET DISK_GROUP <name>
```

Switches

The following switches are available with the SET DISK_GROUP command:

- ADD=
- COMMENT=
- DELETE=
- NAME=
- OCCUPANCY_ALARM=

These switches are described in the following paragraphs.

ADD=

The number of disk drives to add to the specified disk group.

COMMENT=

This is an optional switch that associates a user-defined comment with the disk group. Maximum of 128 characters are allowed and the string must be enclosed in double quotes.

DELETE=

The name of the disk drive to be removed from the disk group. Data is not deleted when the disk is removed from the disk group.

NAME=

New name to give to the specified disk group.

OCCUPANCY_ALARM=

The point when a defined percentage of space is used. When this point is reached, an event log is generated and sent to the Management Appliance (optionally the host system) informing the administrator that the group is becoming full. Do not use the percent sign (%) after the number.

Example

```
SET DISK_GROUP "\Disk Groups\pool" NAME="nuclear secrets"
```

This example renames the disk group *pool* with its new name *nuclear secrets*.

SET FOLDER

The SET FOLDER command changes the specified folder properties.

Syntax

```
SET FOLDER <name>
```

Switches

The following switches are available for the SET FOLDER command:

- COMMENT=
- NAME=

These switches are described in the following paragraphs.

COMMENT=

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed and the string must be enclosed in double quotes.

NAME=

New name to give to the specified folder.

Example

```
SET FOLDER "\Virtual Disks\top_secret\" NAME=everyone_knows
```

This example renames the folder *top_secret* to *everyone_knows*.

SET HOST

The SET HOST command adds or deletes World Wide Names, sets the operating systems for a host, and adds comments to a specified host.

Note: Use the SET HOST command to add additional FCAs to hosts added to the storage system with the ADD HOST command.

Syntax

```
SET HOST <name>
```

Switches

The following switches are available with the SET HOST command.

- **ADD_WORLD_WIDE_NAME=**
- **DELETE_WORLD_WIDE_NAME=**
- **COMMENT=**
- **OPERATING_SYSTEM=**

These switches are described in the following paragraphs.

ADD_WORLD_WIDE_NAME=

Adds the World Wide Name of the host port.

DELETE_WORLD_WIDE_NAME=

Deletes the World Wide Name of the host port.

COMMENT=

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed and the string must be enclosed in double quotes.

OPERATING_SYSTEM=

The type of operating system for the specified host. Choose one of the following operating systems. (they are spelled here as SSSU expects):

- HPUX
- IBMAIX
- OPEN_VMS
- SOLARIS
- TRU64
- WINDOWS
- CUSTOM = (You must include the equal sign after CUSTOM and the value must be 16 digit hex characters.)

If unspecified, the operating system type for the host defaults to Windows.

Examples

```
SET HOST \Hosts\install OPERATING_SYSTEM=OPEN_VMS
SET HOST \Hosts\install ADD_WORLD_WIDE_NAME=1000-0000-C922-36CA
```

The first example assigns OpenVMS as the operating system type for the host *install*.

The second example adds a WWN to the host *install*.

SET MONITOR

The SET MONITOR command changes the specified monitor properties.

Syntax

```
SET MONITOR <name>
```

Switches

The following switches are available with the SET MONITOR command:

- COMMENT=
- NAME=

These switches are described in the following paragraphs.

COMMENT=

This is an optional switch that associates a user-defined comment with the monitor. Maximum of 128 characters (including spaces) are allowed and the string must be enclosed in double quotes.

NAME=

New name to give to the specified monitor.

Example

```
SET MONITOR "\Hardware\Rack 1\Enclosure 7\Controller A" COMMENT="cabinet near the door"
```

This example adds the comment *cabinet near the door* to the monitor *Controller A*.

Note: Remember to include full paths for objects and enclose any paths that contain spaces in double quotes.

SET OPTIONS

The SET OPTIONS command changes the SSSU characteristics.

Note: The options you set with the SET OPTIONS command are in effect for the current session only. Each time you start SSSU, the default options are reinstated.

Syntax

```
SET OPTIONS
```

Switches

The following switches are available with the SET OPTIONS command:

- **COMMAND_DELAY=**
- **NOCOMMAND_DELAY**
- **DISPLAY_WIDTH=**
- **ON_ERROR=**
 - CONTINUE
 - EXIT_ON_ERROR
 - HALT_ON_ERROR
- **RETRIES=**
- **NORETRIES**
- **SAFE_DELETE**
- **NOSAFE_DELETE**

These switches are described in the following paragraphs.

COMMAND_DELAY=

The number of seconds (0 - 300 seconds) to wait between issuing commands when running a script using the FILE command. The default is 10 seconds. This switch has no effect when you are typing commands.

NOCOMMAND_DELAY

Specifies no wait time between commands issued from within a FILE command.

DISPLAY_WIDTH=

Sets the amount of characters displayed on a line for SHOW commands. The default is 80. If the SSSU output is to be parsed, it is useful to set a high line width. A long line keeps the lines from wrapping, making the output easier to cut, paste, and parse. Line widths from 70 to 500 may be specified.

ON_ERROR=

- **CONTINUE**—In this mode, only the EXIT command causes the scripting utility to halt. This is the default and is recommended when you are manually typing commands.

- **EXIT_ON_ERROR**—Any error causes the scripting utility to exit with an error code. This mode is useful when requesting the entire script to halt immediately if errors occur while executing a script. This causes the scripting utility to halt on *any* kind of error—failed command, syntax error, or ambiguous command.
- **HALT_ON_ERROR**—This is similar to EXIT_ON_ERROR. Any error causes the scripting utility to halt, but not exit until any key is pressed. Then the scripting utility exits with an error code. This allows you to see the error before the window closes on exit.

Note: When manually typing commands, HP does not recommend using the EXIT_ON_ERROR or HALT_ON_ERROR modes. You should avoid these modes because SSSU exits on *any* kind of error, including a typographical one.

RETRIES=

Number of minutes to attempt “long” period retries—such as when the command view eva service is either busy or restarting. Specify between one and ten minutes (inclusive). The default is four minutes.

NORETRIES

Specify that you do not want the scripting utility to retry commands.

SAFE_DELETE

Specify if you want to delete all dependent or related objects before the specified object can be deleted. For example, if you specified a deletion of the VDisk which had LUNs presented, the DELETE VDISK command would be rejected with a message telling you that all the LUNs presented from this storage must be deleted before the storage can be deleted. This is the default.

You must always delete related objects if SAFE_DELETE is specified.

NOSAFE_DELETE

Allows the deletion of an object even if related objects are present. The behavior of the NOSAFE_DELETE command depends on the mode of operation (entering commands manually or running commands through a script) that SSSU is running in).

If you are entering commands at a terminal, the **Are you sure?** prompt displays and you must enter **Yes** for the deletion to proceed. Note that you must enter capital “**Y**” followed by lowercase “**es**” for the delete to proceed.

If the input is from a script (using the **FILE** command) or the from command line, the deletion proceeds without warning.



Caution: Using the **NOSAFE_DELETE** switch can cause loss of data.

Example

```
SET OPTION ON_ERROR=HALT_ON_ERROR NORETRIES
```

The example sets the options for SSSU to stop upon encountering an error and not to retry the commands.

SET SYSTEM

Use the SET SYSTEM command to modify system properties.

Syntax

```
SET SYSTEM <name>
```

Switches

The following switches are available for the SET SYSTEM command:

- COMMENT=
- CONSOLE_LUN_ID=
- NAME=

These switches are described in the following paragraphs.

COMMENT=

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed and the string must be enclosed in double quotes.

CONSOLE_LUN_ID=

The LUN used for console communication. If set to zero, a console LUN is not presented to the host.

Note: This value is used for IBM AIX (set to zero), OpenVMS (required), and Tru64 UNIX (recommended). Other host operating systems ignore the value. See the host operating system installation guides for more information.

NAME=

New name to give to the specified system.

Example

```
SET SYSTEM engineering NAME=accounting
```

This example renames the system *engineering* to *accounting*.

SET VDISK

Use the SET VDISK command to change the properties of specific virtual disks.

Syntax

```
SET VDISK <storage_name>
```

Switches

The following switches are available for the SET VDISK command:

- COMMENT=
- MIRRORED_WRITEBACK
- NOMIRRORED_WRITEBACK
- NAME=
- OS_UNIT_ID=
- PREFERRED_PATH=
- NOPREFERRED_PATH
- READ_CACHE
- NOREAD_CACHE
- SIZE=
- WORLD_WIDE_LUN_NAME=
- WRITE_PROTECT
- NOWRITE_PROTECT

These switches are described in the following paragraphs.

COMMENT=

This is an optional switch that associates a user-defined comment with the new system. A maximum of 128 characters is allowed and the string must be enclosed in double quotes.

MIRRORED_WRITEBACK

All writes are stored in mirrored caches. After the data is stored in both caches, the write is complete. If one mirrored cache is unusable, the write is not complete until the data is in both caches.

NOMIRRORED_WRITEBACK

No writes are stored in mirrored caches. After the data is stored in one cache, the write is complete. The write is complete even if a mirror copy of cache is not available.

NAME=

New name to give to the specified virtual disk.

OS_UNIT_ID=

The ID that is presented to the host operating system. If set to zero, an ID is not presented to the host.

Note: This is a required field for OpenVMS and is recommended for Tru64 UNIX. It is optional for all other operating systems.

PREFERRED_PATH=

The preferred controller path that is specified to handle all I/O for the virtual disks. If a controller fails, the path always reverts to the working controller.

- **PATH_A_BOTH**—Controller path A fails over to controller B. When controller A restarts, the virtual disks fail back to controller A. This is failover/failback mode.
- **PATH_A_FAILOVER**—Controller path A fails over to controller B. When controller A restarts, the virtual disks do not fail back over to controller A. This is failover only mode.
- **PATH_B_BOTH**—Controller path B fails over to controller A. When controller B restarts, the virtual disks fail back to controller B. This is failover/failback mode.
- **PATH_B_FAILOVER**—Controller path B fails over to controller A. When controller B restarts, the virtual disks do not fail back over to controller B. This is failover only mode.

NOPREFERRED_PATH

Allows the I/O to be handled by any controller.

READ_CACHE

Reads are satisfied from the controller's cache.

NOREAD_CACHE

Reads are always satisfied from media, not the controller's cache.

SIZE=

Indicates the new larger size (you cannot change to a size smaller than the original) of the storage in gigabytes. You can specify size in whole gigabytes only. Fractions are not allowed. The value of SIZE can be 1 GB through 2000GB, depending on the disk group's available space.

WORLD_WIDE_LUN_NAME=

Sets the World Wide LUN Name on unrepresented storage (either ACTIVE or snapshot). This switch is rejected when issued against storage that is presented to a host.

Note: This switch is commonly used to allow a host to point to a new version of a snapshot by giving the new snapshot the same WWN as the old snapshot.

WRITE_PROTECT

Does not allow writing to the virtual disk for all presented LUNs.

NOWRITE_PROTECT

Allows writing to the virtual disk for all presented LUNs.

Example

```
SET VDISK "\\Virtual Disks\archive\ACTIVE"  
WORLD_WIDE_LUN_NAME=6000-1fe1-ff00-0000
```

The example assigns the world wide lun name to the *ACTIVE* virtual disk of the *archive* family.

SHOW

Use the SHOW commands to display information about various elements in the currently selected system.

Note: All of the SHOW commands require the full path from the root directory to the object you want to display.

SHOW DR_GROUP

Displays a list of data replication groups for the currently selected system. If you specify the name of the DR Group, it displays detailed information about the specified DR Group.

Syntax

```
SHOW DR_GROUP <dr_group_name>
```

Switches

FULL

FULL shows information about all dr_groups on the currently selected system.

Example

```
SHOW DR_GROUP "\\Data Replication\Transactions"
```

This example shows the DR_Group *Transactions*.

SHOW SYSTEM

Use the SHOW SYSTEM command to display the systems currently managed by a command view eva. If <system_name> is specified, detailed information about that specified system is displayed.

Syntax

```
SHOW SYSTEM <system_name>
```

Switch

Only the FULL switch is available with the SHOW SYSTEM command. NOFULL is the default.

FULL

FULL shows information about all systems currently managed by the command view eva.

Examples

```
SHOW SYSTEM
```

```
SHOW SYSTEM payroll
```

The first example displays a list of available systems on this manager. The second example displays specific information about the system *payroll*.

SHOW DISK

Use the SHOW DISK command to display disk configuration information for physical disks connected to the currently selected system.

Syntax

```
SHOW DISK <disk_name>
```

Switch

Only the FULL switch is available with the SHOW DISK command. NOFULL is the default.

FULL

FULL shows information about all objects on the currently selected system.

Examples

```
SHOW DISK
```

```
SHOW DISK "\Disk Groups\Ungrouped Disks\Disk 039"
```

The first example displays a list of all the physical disks in the currently selected system. The second example displays specific information about the physical “Disk 039.”

Note: Remember to include full paths for objects and enclose any paths that contain spaces in double quotes.

SHOW DISK_GROUP

The SHOW DISK_GROUP command displays the disk groups currently configured by this system. If <disk_group_name> is specified, detailed information about the specified disk group is displayed.

Syntax

```
SHOW DISK_GROUP <disk_group_name>
```

Switch

Only the FULL switch is available with the SHOW DISK_GROUP command. NOFULL is the default.

FULL

FULL shows information about all disk groups on the currently selected system.

Examples

```
SHOW DISK_GROUP
```

```
SHOW DISK_GROUP "\Disk Groups\Default Disk Group"
```

The first example displays a list of the available disk groups. The second example displays information about the default disk group.

Note: Remember to include full paths for objects and enclose any paths that contain spaces in double quotes.

SHOW HOST

The SHOW HOST command displays the hosts currently configured on this system. If *<host_name>* is specified, information about the specified host is displayed.

Syntax

```
SHOW HOST <host_name>
```

Switch

Only the FULL switch is available with the SHOW HOST command.

FULL

FULL shows information about all hosts on the currently selected system.

Examples

```
SHOW HOST
```

```
SHOW HOST \Hosts\med_lab
```

The first example displays the list of available hosts on the currently selected system. The second example displays specific information about the host *med_lab*.

Note: Remember to include full paths for objects and enclose any paths that contain spaces in double quotes.

SHOW LUN

The SHOW LUN command displays the LUNs currently configured on this system.

Syntax

```
SHOW LUN <LUN_name>
```

Switch

Only the FULL switch is available with the SHOW LUN command. NOFULL is the default.

FULL

FULL shows information about all LUNs on the currently selected system.

Examples

```
SHOW LUN
```

```
SHOW LUN \Hosts\SVT165\13
```

The first example displays the list of available LUNs on the selected system. The second example displays specific information about the LUN 13 on \Hosts\SVT165.

Note: Remember to include full paths for objects and enclose any paths that contain spaces in double quotes.

SHOW MANAGER

The SHOW MANAGER command displays information about the currently selected manager.

Syntax

```
SHOW MANAGER
```

Switch

The switches FULL and NOFULL can be specified, but are ignored. Specifying a specific manager name is also ignored. The SHOW MANAGER command always displays the information for the currently selected manager.

Example

```
SHOW MANAGER
```

The example displays information about the currently selected manager.

SHOW MONITOR

The SHOW MONITOR command displays the monitors currently installed on this system. If *<monitor_name>* is specified, detailed information about that specified monitor is displayed.

Syntax

```
SHOW MONITOR <monitor_name>
```

Switch

Only the FULL switch is available for the SHOW MONITOR command. NOFULL is the default.

FULL

Use FULL in the absence of *<monitor_name>* to display detailed information about all monitors.

Examples

```
SHOW MONITOR
```

```
SHOW MONITOR "\Hardware\Rack 1\blue_cab"
```

The first example lists monitors available on the currently selected system.

The second example displays information about the monitor *blue_cab*.

Note: Remember to include full paths for objects and enclose any paths that contain spaces in double quotes.

SHOW OPTIONS

The SHOW OPTIONS command displays the current SSSU options.

Syntax

```
SHOW OPTIONS
```

Example

```
SHOW OPTIONS
```

The example shows the following current options set for SSSU:

```
Options:  
NOCOMMAND_DELAY  
DISPLAY_WIDTH = 80  
ON_ERROR = CONTINUE  
RETRIES = 4 (minutes)  
SAFE_DELETE
```

SHOW POWER

The SHOW POWER command displays the controllers currently configured on this system. If you specify *<controller_name>*, detailed information about that specific controller is displayed. Controllers are the only power objects.

Syntax

```
SHOW POWER <controller_name>
```

Switch

Only the FULL switch is available with the SHOW POWER command. NOFULL is the default.

FULL

Use FULL in the absence of *<controller_name>* to display detailed information about all controllers.

Examples

```
SHOW POWER
```

```
SHOW POWER "\Hardware\Rack 1\Enclosure 7\Controller A"
```

The first example displays a list of all power objects within the selected system. The second example displays specific information about *Controller A*.

Note: Remember to include full paths for objects and enclose any paths that contain spaces in double quotes.

SHOW VDISK

The SHOW VDISK command displays the storage currently configured on this system. If *<storage_name>* is specified, detailed information about the specified storage.

Syntax

```
SHOW VDISK <storage_name>
```

Switch

Only the FULL switch is available with the SHOW VDISK command. NOFULL is the default.

FULL

Use FULL in the absence of *<storage_name>* to display detailed information about all virtual disks.

Examples

```
SHOW VDISK
```

```
SHOW VDISK "\Virtual Disks\scratch_disk\ACTIVE"
```

The first example displays a list of all virtual disks in the currently selected system. The second example displays information about the specific active virtual disk *scratch_diskACTIVE*.

Note: Remember to include full paths for objects and enclose any paths that contain spaces in double quotes.

SHOW WORLD_WIDE_NAME

The SHOW WORLD_WIDE_NAME command displays the host World Wide Names (WWNs) that are visible to this system and not already assigned to a host.

Syntax

```
SHOW WORLD_WIDE_NAME
```

Example

```
SHOW WORLD_WIDE_NAME
```

This example displays all of the WWNs visible to this system which have not been assigned to a host.

SHUTDOWN

Use the SHUTDOWN command to shut down any controller displayed by the [SHOW POWER](#) command.

Syntax

```
SHUTDOWN <controller_name>
```

Switches

The following switches are available with the SHUTDOWN command.

- ALL_PEERS
- NOALL_PEERS

These switches are described in the following paragraphs.

ALL_PEERS

Shuts down all peer controllers (both controllers) on this system.

NOALL_PEERS

Default. Shuts down only the specified controller.

Examples

```
SHUTDOWN "\Hardware\Rack 1\Enclosure 7\Controller A" NOALL_PEERS
```

```
SHUTDOWN "\Hardware\Rack 1\Enclosure 7\Controller B" ALL_PEERS
```

The first example only shuts down *Controller A*.

The second example shuts down *Controller B* and the peer.

Configuration Examples



This appendix provides examples that describe how to use commands to create a simple configuration and to view specific system information. The following topics are covered:

- [Creating a Simple Configuration](#), page 88
- [Using the SHOW Commands](#), page 92

Creating a Simple Configuration

This section presents a simple configuration example in two ways:

- Annotated with steps (page 88 through page 91)
- Sample output as it displays after executing all of the commands (page 91)

Creating the Storage System

- Select the manager with the `SELECT MANAGER` command.
 - View the systems available on the manager (Uninitialized Storage System1, if this is an uninitialized storage system), with the `SHOW SYSTEM` command.
 - Add (create) the storage system with the `ADD SYSTEM` command.
1. Start SSSU from anywhere on the same network as your management appliance.

The `NoSystemSelected>` prompt displays.

2. Enter the `SELECT MANAGER` command with the name of the manager and the logon information (a sample username, password, and manager are used in the following example).

```
NoSystemSelected> SELECT MANAGER swma31k008 USERNAME=XXX  
PASSWORD=XXX  
NoSystemSelected>
```

Note: Remember to define the username and password through the management appliance—not in SSSU.

3. At the prompt, enter the `SHOW SYSTEM` command to display all the available systems on this manager.

```
NoSystemSelected> SHOW SYSTEM  
Systems available on this Manager:  
Uninitialized Storage System1  
NoSystemSelected>
```

The display includes systems that are available on this manager. In the example, the default for an uninitialized storage system is one system named *Uninitialized Storage System1*.

4. Select the Uninitialized Storage System1 system with the SELECT SYSTEM command.

```
NoSystemSelected> SELECT SYSTEM "Uninitialized Storage System1"  
Uninitialized Storage System1>
```

The prompt changes to reflect the selected system (*Uninitialized Storage System1*).

5. Initialize the storage system with the ADD SYSTEM command, providing a name for the system (such as *Yekao*). This creates one system using 8 disks in the default Disk Group. The default number of disks is 8.

```
Uninitialized Storage System1> ADD SYSTEM Yekao  
NoSystemSelected>
```

Note: The prompt returns to `NoSystemSelected>` because the system's name has changed. You must reselect the system using the name you just gave it (`SELECT SYSTEM Yekao`).

Attempting to add (create) a system without the basic license returns an error message. This message indicates that you need to enter your licensing information in the command view `eva`. You must have a license and follow the registration process.

Creating and Presenting a Virtual Disk

Here are the steps to follow to create a virtual disk and present the system to a host.

- Select the newly created system
 - Add a virtual disk (ADD VDISK) to the storage system
 - Add a host to the system (ADD HOST)
 - Add a LUN using a host and a VDISK
1. Use the SELECT SYSTEM command to select the system you just created.

```
NoSystemSelected> SELECT SYSTEM Yekao  
Yekao>
```

The prompt changes to reflect the object selected. In this case, the system *Yekao*.

2. Create a virtual disk using the ADD VDISK command, supplying virtual disk name and size in whole GBs.

```
Yekao> ADD VDISK Enicar SIZE=4  
Yekao>
```

This example creates a 4 GB virtual disk named *Enicar* is created with a Vraid0 redundancy.

Note: The virtual disk (storage) can be between 1 GB (minimum) and 2000GBs (maximum). The increments must be in whole GBs.

To present this newly created virtual disk, you need to have at least one host within the storage system. This host must be added to the storage system configuration.

You can use the SHOW WORLD_WIDE_NAME command to see the available Fibre Channel Adapter World Wide Names.

3. Use the ADD HOST command to add a host to the storage system, supplying the World Wide Name (WWN) for the host port (FCA).

Use caution with WWN assignments as these are not verified when assigned.

```
Yekao> ADD HOST \Hosts\SVT165  
WORLD_WIDE_NAME=1000-0000-c923-6735  
Yekao> SET HOST \Hosts\SVT165  
ADD_WORLD_WIDE_NAME=1000-0000-c923-6736
```

4. Add the LUN to the host. Use the ADD LUN command, supplying the full path to the virtual disk and the host machine.

```
Yekao> ADD LUN 13 VDISK="Virtual Disks\Enicar\ACTIVE"  
HOST=\Hosts\SVT165
```

Remember, for any path or name that contains spaces, you must enclose the entire string in double quotes, as in the example above.

Now LUN 13 is presenting the VDISK to the host.

Sample Configuration Command Output

The following example shows the commands that were entered during the previous procedures (page 88 through page 91), without the interruption of the instructions.

```
NoSystemSelected> SELECT MANAGER swma31k008 USERNAME=XXX  
PASSWORD=XXX  
NoSystemSelected> SHOW SYSTEM  
Systems available on this Manager:  
Uninitialized Storage System1  
NoSystemSelected> SELECT SYSTEM "Uninitialized Storage System1"  
Uninitialized Storage System1> ADD SYSTEM Yekao  
NoSystemSelected> SELECT SYSTEM Yekao  
Yekao> ADD VDISK Enicar SIZE=4  
Yekao> ADD HOST SVT165 WORLD_WIDE_NAME=1000-0000-c923-6735  
Yekao> SET HOST \Hosts\SVT165  
ADD_WORLD_WIDE_NAME=1000-0000-c923-6736  
Yekao> ADD LUN 13 VDISK="Virtual Disks\Enicar\ACTIVE"  
HOST=\Hosts\SVT165
```

Using the SHOW Commands

This section provides examples of using the SHOW command, based on the storage system, host, and virtual disk created in the previous section, “[Creating a Simple Configuration](#)” on page 88. The purpose of this presentation is to provide a sample of SSSU output.

To view information about various system attributes, you first select the system and then use the SHOW commands.

SHOW SYSTEM

Use the SHOW SYSTEM command to view information about the selected system.

1. At the `NoSystemSelected>` prompt, enter SHOW SYSTEM.

```
NoSystemSelected> SHOW SYSTEM
Systems available on this Manager:
Yekao
Makawao
```

```
NoSystemSelected>
```

The available systems on this manager are displayed.

2. Select the system you want with the SELECT command.

```
NoSystemSelected>SELECT SYSTEM Yekao
```

3. Use the SHOW SYSTEM command to display information about the system.

```
Yekao> show system Yekao
Yekao information:
Initialized Enterprise System Properties:
      Name : Yekao
      Licenses : BASIC SNAPSHOT DRM
      Operational_State : Initialized - Good
      Total_storage_space : 947.78 GB
      Storage_space_used : 194.51 GB
      Available_storage_space : 753.26 GB
      Device_addition_policy : Manual
      Console_LUN_ID : 0
      Controller_time : 08-Jan-1947 15:01:00
      Node_world_wide_name : 5000-1FE1-5000-5C20
      Comments :
      ID : 08000710b4080560bc49010000d0000000000200
      ElementRefreshStatus : 0,0,0,0,0,0,0,0,3,0,0,0
Yekao>
```

SHOW VDISK

Use the SHOW VDISK command to display information about the virtual disks (storage) within the selected system.

1. Enter the SHOW VDISK command (with a system selected) to display a list of available storage.

```
Yekao> show vdisk
Vdisks available on this System:
  \Virtual Disks\maui6228_1\ACTIVE
  \Virtual Disks\maui_1\ACTIVE
  \Virtual Disks\lanai_1\ACTIVE
Yekao>
```

Note: Remember that the ADD VDISK command creates a virtual disk family and you must include “\ACTIVE” when referring to the actual virtual disk.

2. Use the SHOW VDISK command to display information about the virtual disk.

```
Yekao> show vdisk "\Virtual Disks\maui6228_1\ACTIVE"
\Virtual Disks\maui6228_1\ACTIVE information:
Identification:
      Name : \Virtual Disks\maui6228_1\ACTIVE
      Family_Name : maui6228_1
      Hex_UUID : 6005-08b4-0001-49bc-0000-e000-000d-0000
      World_Wide_Lun_Name : 6005-08b4-0001-49bc-0000-e000-000d-0000
Attributes:
      Disk_Type : Original
      Disk_Group : \Disk Groups\Default Disk Group
      Capacity_Req : 25
      Capacity_Used : 25
      Sharing_Relationship : none
      Redundancy : vRaid5
      Write_Cache_Policy : Mirrored write-back
      Read_Cache_Policy : On
      Comm_Protocol :
      Write_Protect : No
```

```
Enable_Disable : Enable
OS_Unit_ID : 0
Preferred_path : No preference
Comments :
ID : 01200710b4080560bc49010000e000000000d00
Condition_State:
Operational_State : 1
Date Time:
Created : 28-Jun-1946 18:28:34
Presentation [0]:
StorageClientName : \Hosts\maui_6228
Lun : 1
DR_Attributes:
Remote_VDisk:
Yekao>
```

SHOW HOST

Use the SHOW HOST command to display information about the hosts available on the selected storage system.

1. With the system selected, enter SHOW HOST to display the list of available hosts.

```
Yekao> show host
Hosts available on this System:
\Hosts\maui
\Hosts\maui_6228
\Hosts\lanai
Yekao>
```

2. Use the SHOW HOST command, followed by the specific host name, to display information about the host.

Note: Remember that you need to type the full path to the host object, and that if there are any spaces within the path or host name, the entire string needs to be enclosed within double quotes.

```
Yekao> show host "\Hosts\maui"
\Hosts\maui information:
Identification:
    Name : \Hosts\maui
    IPAddress : Dynamic IP Assignment
    Status : Initialized - Good
    Operating_System : IBMAIX
Fibre_Channel_adapter_ports:
Fibre_Channel_adapter_ports [0]:
    Port : 5005-02e0-1000-09fe
Fibre_Channel_adapter_ports [1]:
    Port : 5005-02e0-1000-018b
Presentation:
Presentation [0]:
    Lun : 1
    Virtual_disk : \Virtual Disks\maui_1\ACTIVE
    Comments :
    ID : 00800710b4080560bc49010000d0000000001b00
Yekao>
```

SHOW LUN

Use the **SHOW LUN** command to display information about specific LUNs.

1. With the system selected, enter the **SHOW LUN** command to display the list of available LUNs.

```
Yekao> show lun
LUNs available on this Cell:
\Hosts\maui_6228\1
\Hosts\maui\1
\Hosts\lanai\1
```

2. Use the **SHOW LUN** command, with the specific LUN named, to display the specific LUN information.

```
Yekao> show lun "\Hosts\maui\1"  
  \Hosts\maui\1 information:  
      Name : \Hosts\maui\1  
      StorageName : \Virtual Disks\maui_1\ACTIVE  
      StorageclientName : \Hosts\maui  
      ID : 00880710b4080560bc49010000e000000000b00  
Yekao>
```


This glossary defines terms used in this guide or related to this product and is not a comprehensive glossary of computer terms.

active virtual disk

A virtual disk (VD) is a simulated disk drive created by the controllers as storage for one or more hosts. An active virtual disk is accessible by one or more hosts for normal storage. An active virtual disk and its snapshot, if one exists, constitute a virtual disk family. An active virtual disk is the only necessary member of a virtual disk family.

See also **virtual disk, virtual disk copy, virtual disk family, and snapshot.**

array

All the physical disk drives in a storage system that are known to and under the control of a controller pair.

array controller

See **controller.**

block

Also called a sector. The smallest collection of consecutive bytes addressable on a disk drive. In integrated storage elements, a block contains 512 bytes of data, error codes, flags, and the block address header.

cache

High-speed memory that sets aside data as an intermediate data buffer between a host and the storage media. The purpose of cache is to improve performance.

See also **read cache, write cache, and mirrored cache.**

command view eva

A graphical user interface through which users control and monitor a storage system. The software can be installed on one or more management appliances in a fabric. Each command view eva installation is a management agent.

communication logical unit number (LUN)

See **console LUN.**

console LUN

A SCSI-3 virtual object that makes a controller pair accessible by the host before any virtual disks are created. *Also* called a communication LUN.

console LUN ID

The ID that can be assigned when a host operating system requires a unique ID. The console LUN ID is assigned by the user, usually when the storage system is initialized.

See also **console LUN**.

controller

A hardware/firmware device that manages communications between host systems and other devices. Controllers typically differ by the type of interface to the host and provide functions beyond those the devices support.

controller pair

Two interconnected controller modules which together control a physical disk array. A controller pair and the disk array together constitute a storage system.

default disk group

The first disk group created at the time the system is initialized. The default disk group can contain the entire set of physical disks in the array or just a few of the disks.

See also **disk group**.

disk

A physical storage device (disk drive) connected to the system.

disk failure protection

A method by which a controller pair reserves drive capacity to take over the functionality of a failed or failing physical disk.

DISK_GROUP

A collection of disks from which storage is created.

DR_GROUP

One or more data replication groups that preserve write ordering and share a common log.

EMU

Environmental Monitoring Unit. An element that monitors the status of an enclosure, including the power, air temperature, and blower status. The EMU detects problems and displays and reports these conditions to a user and the controller. In some cases, the EMU implements corrective action.

Enterprise Virtual Array

The hp name used to describe the storage system that includes controllers, storage devices, enclosures, cables, and power supplies. Also called the Enterprise Storage System.

environmental monitoring unit

See EMU.

event log

System information sent to the Management Appliance or host operating system.

fabric

A Fibre Channel fabric switch or two or more interconnected Fibre Channel switches allowing data transmission.

fabric port

A port that is capable of supporting an attached arbitrated loop. This port on a loop has the AL_PA hexadecimal address 00 (loop ID 7E), giving the fabric the highest priority access to the loop. A loop port is the gateway to the fabric for the node ports on a loop.

failover

The process that takes place when one controller assumes the workload of a failed companion controller. Failover continues until the failed controller is operational.

FC HBA

Fibre Channel Host Bus Adapter. An interchangeable term for Fibre Channel adapter.

See also FCA.

FCA

Fiber Channel Adapter. An adapter used to connect the host server to the fabric. *Also* called a Host Bus Adapter (HBA) or a Fibre Channel Host Bus Adapter (FC HBA).

See also FC HBA

fiber

The optical media used to implement Fibre Channel.

fibre

The international spelling that refers to the Fibre Channel standards for optical media.

Fibre Channel

A data transfer architecture designed for mass storage devices and other peripheral devices that require very high bandwidth.

Fibre Channel adapter

See FCA.

GB

Gigabyte. A unit of measurement defining either:

- A data transfer rate.
See also GBps
- A storage or memory capacity of 1,073,741,824 (2^{30}) bytes.

GBps

Gigabytes per second. A measurement of the rate at which the transfer of bytes of data occurs. A GBps is a transfer rate of 1,000,000,000 (10^9) bits per second.

See also GB.

Giga (G)

The notation to represent 10^9 or 1 billion (1,000,000,000).

HBA

Host Bus Adapter.

See FCA.

host

A system that has data path access to the virtual disk through a logical unit number (LUN) defined on the system.

Host Bus Adapter

See FCA.

host computer

See host.

host ports

A connection point to one or more hosts through a Fibre Channel fabric. A host is a computer that runs user applications and that uses (or can potentially use) one or more of the virtual disks that are created and presented by the controller pair.

host-side ports

See host ports.

initialization

A process that prepares a storage system for use. Specifically, the system binds controllers together as an operational pair and establishes preliminary data structures on the disk array. Initialization also sets up the first disk group, called the default disk group.

LUN

Logical Unit Number. The host-accessible presentation of a virtual disk on the system.

manager

The object that controls configuration of the system. The StorageWorks command view eva is the manager in the case of the Enterprise Virtual Array.

MB

Megabyte. A term defining either:

- A data transfer rate.
See also MBps.
- A measure of either storage or memory capacity of 1,048,576 (2^{20}) bytes.

MBps

Megabytes per second. A measure of bandwidth or data transfers occurring at a rate of 1,000,000 (10^6) bytes per second.

Mega

A notation denoting a multiplier of 1 million (1,000,000).

mirrored caching

A process in which half of each controller's write cache mirrors the companion controller's write cache.

mirroring

The act of creating an exact copy or image of data.

monitor

An element that monitors the status of an enclosure, including the power, air temperature, and blower status.

parity

A method of checking if binary numbers or characters are correct by counting the ONE bits. In odd parity, the total number of ONE bits must be odd; in even parity, the total number of ONE bits must be even. Parity information can be used to correct corrupted data.

parity bit

A binary digit added to a group of bits that checks to see if there are errors in the transmission.

parity check

A method of detecting errors when data is sent over a communications line. With even parity, the number of ONE bits in a set of binary data should be even. With odd parity, the number of ONE bits should be odd.

physical disk

A disk drive mounted in a disk drive enclosure that communicates with a controller pair through the device-side Fibre Channel loops. A physical disk is hardware with embedded software, as opposed to a virtual disk, which is constructed by the controllers. Only the controllers can communicate directly with the physical disks.

The physical disks, in aggregate, are called the array and constitute the storage pool from which the controllers create virtual disks.

physical disk array

See array.

port

A Fibre Channel connector on a Fibre Channel device.

port_name

A 64-bit unique identifier assigned to each Fibre Channel port. The port_name is communicated during the login and port discovery processes.

preferred path

A preference for which controller of the controller pair manages the virtual disk. This preference is set by the user through the command view eva when creating the virtual disk. A user can change the preferred path of a virtual disk at any time. The main purpose of selecting a preferred path is load balancing.

read caching

A cache method used to decrease subsystem response times to a read request by allowing the controller to satisfy the request from the cache memory rather than from the disk drives. Reading data from cache memory is faster than reading data from a disk. The read cache is specified as either on or off for each virtual disk. The default state is on.

read ahead caching

A cache management method used to decrease the subsystem response time to a read request by allowing the controller to satisfy the request from the cache memory rather than from the disk drives.

redundancy

- **Element Redundancy**—The degree to which logical or physical elements are protected by having another element that can take over in case of failure. For example, each loop of a device-side loop pair normally work independently but can take over for the other in case of failure.
- **Data Redundancy**—The level to which user data is protected. Redundancy is directly proportional to cost in terms of storage usage; the greater the level of data protection, the more storage space is required.

SCSI (small computer system interface)

- An American National Standards Institute (ANSI) interface that defines the physical and electrical parameters of a parallel I/O bus used to connect computers and a maximum of 16 bus elements.
- The communication protocol used between a controller pair and the hosts. Specifically, the protocol is FC-AL or SCSI on a Fibre Channel. SCSI is the higher command-level protocol and Fibre Channel is the low-level transmission protocol. The controllers have full support for SCSI-2; additionally, they support some elements of SCSI-3.

SCSI-3

The ANSI standard that defines the operation and function of Fibre Channel systems.

snapshot

A temporary virtual disk (VD) that reflects the contents of another virtual disk at a particular point in time. A snapshot operation is only done on an active virtual disk. The active disk and its snapshot constitute a virtual disk family.

See also **active virtual disk**, **virtual disk copy**, and **virtual disk family**.

spare policy

This is the disk failure protection level set for the system and disk groups.

See also [disk failure protection](#).

SSN

Storage System Name. A user-defined, unique 20-character name that identifies a specific storage system.

storage pool

The aggregated blocks of available storage in the total physical disk array.

storage system

The controllers, storage devices, enclosures, cables, and power supplies and their software.

Storage System Scripting Utility (SSSU)

A command-line application supplied in the host operating system kits that lets you configure and control HSV controllers.

Storage System Name

See SSN.

switch

An electro-mechanical device that initiates an action or completes a circuit.

system

A controller or controller pair that is treated conceptually as one controller. In the command view *eva*, the system is the name you give the initialized storage system.

TB

Terabyte. A term defining either:

- A data transfer rate.
See also TBps.
- A measure of either storage or memory capacity of 1,099,511,627,776 (2^{40}) bytes.

TBps

Terabytes per second. A data transfer rate of 1,000,000,000,000 (10^{12}) bytes per second.

uninitialized storage system

A state in which the storage system is not ready for use.

See also [initialization](#).

VDisk (virtual disk)

A collection of blocks created on one or more disks that can be used by a host for information storage and retrieval.

The host computer sees the virtual disk as “real,” with the characteristics of an identical physical disk.

See also [active virtual disk](#), [virtual disk copy](#), [virtual disk family](#), and [virtual disk snapshot](#)

virtual disk copy

A clone or exact replica of another virtual disk at a particular point in time. Only an active virtual disk can be copied. A copy immediately becomes the active disk of its own virtual disk family.

See also [active virtual disk](#), [virtual disk family](#), and [virtual disk snapshot](#)

virtual disk family

A virtual disk and its snapshot, if a snapshot exists, constitute a family. The original virtual disk is called the active disk. When you first create a virtual disk family, the only member is the active disk.

See also active virtual disk, virtual disk copy, and virtual disk snapshot.

virtual disk snapshot

See snapshot.

Vraid0

A virtualization technique that provides no data protection. Data host is broken down into chunks and distributed on the disks comprising the disk group from which the virtual disk was created. Reading and writing to a Vraid0 virtual disk is very fast and makes the fullest use of the available storage, but there is no data protection (redundancy) unless there is parity.

Vraid1

A virtualization technique that provides the highest level of data protection. All data blocks are mirrored or written twice on separate physical disks. For read requests, the block can be read from either disk, which can increase performance. Mirroring takes the most storage space because twice the storage capacity must be allocated for a given amount of data.

Vraid5

A virtualization technique that provides moderate data protection using parity striping. Parity is a data protection mechanism for a striped virtual disk. A striped virtual disk is one whose data to and from the host is broken down into chunks and distributed on the physical disks comprising the disk group in which the virtual disk was created. If the striped virtual disk has parity, another chunk (a parity chunk) is calculated from the set of data chunks and written to the physical disks. If one of the data chunks becomes corrupted, the data can be reconstructed from the parity chunk and the remaining data chunks.

World Wide Name

See WWN.

write back caching

A controller process that notifies the host that the write operation is complete when the data is written to the cache. This occurs before transferring the data to the disk. Write back caching improves response time since the write operation completes as soon as the data reaches the cache. As soon as possible after caching the data, the controller then writes the data to the disk drives.

write caching

A process when the host sends a write request to the controller, and the controller places the data in the controller cache module. As soon as possible, the controller transfers the data to the physical disk drives.

WWN

World Wide Name. A unique Fibre Channel identifier consisting of a 16-character hexadecimal number. A WWN is required for each Fibre Channel communication port.

A

ADD commands 29

ADD COPY

description 29

examples 31

switches

DISK_GROUP= 29

OS_UNIT_ID= 30, 39

VDISK= 30

WORLD_WIDE_LUN_NAME= 30

syntax 29

ADD DISK_GROUP

description 32

example 34

switches

DEVICE_COUNT= 33

OCCUPANCY_ALARM= 33

SPARE_POLICY= 34

syntax 33

ADD DR_GROUP

description 31

example 32

Syntax 31

ADD FOLDER

description 34

examples 35

switch, COMMENT= 35

syntax 35

ADD HOST

description 36

example 37

switches

COMMENT= 36

OPERATING_SYSTEM= 36

WORLD_WIDE_NAME= 37

syntax 36

using 90

ADD LUN

description 37

examples 38

switches

HOST= 37

VDISK= 37

syntax 37

using 91

ADD SNAPSHOT

description 38

examples 40

switches

ALLOCATION_POLICY= 39

VDISK= 39

WORLD_WIDE_LUN_NAME= 39

syntax 38

ADD SYSTEM

description 40

example 42

switches

COMMENT= 33, 35, 36, 41, 43, 73

CONSOLE_LUN_ID= 41

DEVICE_COUNT= 41

SPARE_POLICY= 42

syntax 41

using 89

ADD VDISK

- description [42](#)
- examples [47](#)
- switch defaults [46](#)
- switches
 - DISK_GROUP= [43](#)
 - MIRRORED_WRITEBACK [44](#)
 - NOPREFERRED_PATH [44](#)
 - NOREAD_CACHE [45](#)
 - NOWRITE_PROTECT [45](#)
 - OS_UNIT_ID= [44](#)
 - PREFERRED_PATH= [44, 75](#)
 - READ_CACHE [44](#)
 - REDUNDANCY= [45](#)
 - SIZE= [45](#)
 - WORLD_WIDE_LUN_NAME= [45](#)
 - WRITE_PROTECT [45](#)
- syntax [42](#)
- using [90](#)
- audience [20](#)
- authorized reseller, HP [18](#)

C**CAPTURE CONFIGURATION**

- description [48](#)
- example [49](#)
- restoring configurations [49](#)
- syntax [48](#)

CAPTURE VALIDATE

- description [50](#)
- example [50](#)
- syntax [50](#)

command view eva [20](#)**configuration**

- creating [88](#)
- importance of capturing [22](#)

CONSOLE_LUN_ID

- and operating systems [41](#)

conventions

- document [16](#)
- text symbols [16](#)

COPY, ADD [29](#)**creating**

- and presenting a virtual disk [90](#)
- simple configuration [88](#)
- the storage system [88](#)

D**DELETE commands** [51](#)**DELETE DISK_GROUP**

- description [51](#)
- example [52](#)
- syntax [51](#)

DELETE DR_GROUP

- example [51](#)
- syntax [51](#)

DELETE FOLDER

- description [52](#)
- examples [52](#)
- syntax [52](#)

DELETE HOST

- description [52](#)
- example [53](#)
- syntax [52](#)

DELETE LUN

- description [53](#)
- example [53](#)
- syntax [53](#)

DELETE SYSTEM

- description [53](#)
- example [54](#)
- syntax [53](#)

DELETE VDISK

- description [54](#)
- examples [55](#)
- syntax [54](#)

DISK

- SET [64](#)
- SHOW [79](#)

DISK_GROUP

- ADD [32](#)
- DELETE [51](#)
- SET [65](#)
- SHOW [79](#)

document
conventions 16

E

Enterprise Virtual Array, and SSSU 20

EXIT

description 56
example 56
syntax 56

F

FILE

description 57
examples 57
syntax 57

FOLDER

ADD 34
DELETE 52
SET 65

formatting and output of returned data 20

G

getting help 18
with commands 25

H

HELP

description 58
example 58
syntax 58

help, obtaining 18

HOST

ADD 36
DELETE 52
SET 67
SHOW 80

HP

authorized reseller 18
storage website 18
technical support 18

HP-UX 36, 68

HSV controllers 20

I

IBM AIX 36, 68
and the CONSOLE_LUN_ID 41, 73
installing SSSU 21

L

LUN

ADD 37
DELETE 53
SHOW 81

M

MANAGER

SELECT 60
SHOW 82

MONITOR

SET 68
SHOW 82

N

naming

disk group example 23
host example 23
paths and 23
virtual disk example 23

O

OpenVMS 36, 68

and the CONSOLE_LUN_ID 41, 73

operating systems

how named in SSSU 36
setting with ADD HOST 68

OPTIONS

SET 69
SHOW 83

options, setting for SSSU 69

output of returned data, formatting 20

overview, of SSSU 20

P

paths and naming 23

POWER, SHOW 83

presenting a virtual disk, creating and 90

R

RESTART 59

description 59

examples 59

switches

ALL_PEERS 59, 86

NOALL_PEERS 59, 86

syntax 59

restoring configurations 49

returned data, formatting and output of 20

S

SELECT 60

SELECT MANAGER

description 60

examples 61

switches

PASSWORD= 61

USER= 61

syntax 60

SELECT SYSTEM

description 60

examples 60

syntax 60

using 89, 90

SET commands 62

SET DISK

description 64

example 65

switches

COMMENT= 65

NAME= 65

syntax 64

SET DISK_GROUP

description 65

example 66

switches

ADD= 65

COMMENT= 65

DELETE= 66

NAME= 66

OCCUPANCY_ALARM= 66

syntax 65

SET DR_GROUP

description 62

example 64

syntax 62

SET FOLDER

description 65

example 67

switches

COMMENT= 66

NAME= 67

syntax 66

SET HOST

description 67

examples 68

switches

ADD_WORLD_WIDE_NAME= 67

COMMENT= 68

DELETE_WORLD_WIDE_NAME= 67

OPERATING_SYSTEM= 68

syntax 67

SET MONITOR

description 68

example 69

switches

COMMENT= 69

NAME= 69

syntax 68

SET OPTIONS

description 69

examples 72

switches

COMMAND_DELAY= 70

DISPLAY_WIDTH= 70

NOCOMMAND_DELAY 70

NORETRIES= 71

NOSAFE_DELETE 71

ON_ERROR= 70

- RETRIES= 71
- SAFE_DELETE 71
- syntax 69
- SET SYSTEM
 - description 73
 - example 73
 - switches
 - CONSOLE_LUN_ID= 73
 - NAME= 73
 - syntax 73
- SET VDISK
 - description 74
 - examples 76
 - switches
 - COMMENT= 74
 - MIRRORED_WRITEBACK 44, 74
 - NAME= 75
 - NOMIRRORED_WRITEBACK 44, 75
 - NOPREFERRED_PATH 75
 - NOREAD_CACHE 76
 - NOWRITE_PROTECT 76
 - OS_UNIT_ID= 75
 - READ_CACHE 75
 - SIZE= 76
 - WORLD_WIDE_LUN_NAME= 30, 76
 - WRITE_PROTECT 76
 - syntax 74
- SHOW
 - commands 77
 - output samples 92
- SHOW DISK
 - description 79
 - examples 79
 - syntax 79
- SHOW DISK FULL switch 79
- SHOW DISK_GROUP
 - description 79
 - examples 80
 - FULL switch 80
 - syntax 79
- SHOW DR_GROUP
 - description 77
 - example 77
 - syntax 77
- SHOW HOST
 - description 80
 - examples 80
 - FULL switch 80
 - output samples 94
 - syntax 80
- SHOW LUN
 - description 81
 - examples 81
 - FULL switch 81
 - output samples 95
 - syntax 81
- SHOW MANAGER
 - description 82
 - examples 82
 - syntax 82
- SHOW MONITOR
 - description 82
 - examples 82
 - FULL switch 82
 - syntax 82
- SHOW OPTIONS
 - description 83
 - example 83
 - syntax 83
- SHOW POWER
 - description 83
 - examples 84
 - FULL switch 84
 - syntax 83
- SHOW SYSTEM
 - description 77
 - examples 78
 - FULL switch 78, 80
 - output samples 92
 - syntax 77
 - using 88

SHOW VDISK

- description [84](#)
- examples [84](#)
- FULL switch [84](#)
- output samples [93](#)
- syntax [84](#)

SHOW WORLD_WIDE_NAME

- description [85](#)
- example [85](#)
- syntax [85](#)

SHUTDOWN

- description [86](#)
- examples [86](#)
- syntax [86](#)

simple configuration, creating a [88](#)

SNAPSHOT, ADD [38](#)

SSSU

- installing [21](#)
- naming operating systems [36](#)
- overview [20](#)
- setting interface options [69](#)

storage system, creating [88](#)

Sun Solaris [36](#), [68](#)

switch defaults, ADD VDISK [46](#)

symbols in text [16](#)

SYSTEM

- ADD [40](#)
- DELETE [53](#)
- SELECT [60](#)

SET [73](#)

SHOW [77](#)

T

technical support, HP [18](#)

text symbols [16](#)

Tru64 UNIX [36](#), [68](#)

and the CONSOLE_LUN_ID [41](#), [73](#)

V

VDISK

ADD [42](#)

DELETE [54](#)

SET [74](#)

SHOW [84](#)

virtual disk, creating and presenting a [90](#)

Vraid0 [45](#)

Vraid1 [34](#), [45](#)

Vraid5 [34](#), [45](#)

W

websites, HP storage [18](#)

Windows 2000 [36](#), [68](#)

Windows NT [36](#), [68](#)

WORLD_WIDE_LUN_NAME, use with
snapshots [30](#)

WORLD_WIDE_NAME, SHOW [85](#)