PART NUMBER	VERSION NUMBER	EDITION NUMBER
313457304	3.1.0	E

SVA™ ADMINISTRATOR
SOFTWARE

USER'S GUIDE
For Windows

PRODUCT TYPE
SOFTWARE

Shared Virtual Array
Administrator

Version 3.1.0

for Windows

User's Guide

Part Number: 313457304

Information contained in this publication is subject to change without notice.

We welcome your feedback. Please contact the Global Learning Solutions Feedback System at:

GLSFS@Stortek.com

or

Global Learning Solutions
Storage Technology Corporation
One StorageTek Drive
Louisville, CO 80028-3256
USA

Please include the publication name, part number, and edition number in your correspondence if they are available.

Export Destination Control Statement

These commodities, technology or software were exported from the United States in accordance with the Export Administration Regulations. Diversion contrary to U.S. law is prohibited.

Restricted Rights

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) and (2) of the Commercial Computer Software - Restricted Rights at FAR 52.227-19 (June 1987), as applicable.

Limitations on Warranties and Liability

Storage Technology Corporation cannot accept any responsibility for your use of the information in this document or for your use in any associated software program. You are responsible for backing up your data. You should be careful to ensure that your use of the information complies with all applicable laws, rules, and regulations of the jurisdictions in which it is used.

Proprietary Information Statement

The information in this document, including any associated software program, may not be reproduced, disclosed or distributed in any manner without the written consent of Storage Technology Corporation.

Should this publication be found, please return it to StorageTek, One StorageTek Drive, Louisville, CO 80028-5214, USA. Postage is guaranteed.

Edition E, May 2005. EC 132282.

Contents

Tables

Preface

This book describes how to use the Shared Virtual Array Administrator for Windows (referred to as "SVAA"). SVAA provides configuration and administration functions for the StorageTek Shared Virtual Array (SVA).

Who Should Read This Book

This book is for data administrators, capacity planners, performance specialists, and system administrators. This book assumes that you are familiar with disk array operations and Windows system administration.

Note: In this book, Windows 2000 Server and Windows 2003 Server are referred to collectively as Windows. In the few instances where there are differences in SVAA function, the specific operating system is identified.

Conventions

The following typographic conventions are used in this book for showing command syntax and examples.

Table 0-1 Typographic conventions for commands and examples.

Typeface or symbol	Meaning	Example
`AaBb123`	Name of command, reserved keywords, mandatory punctuation.	`querysubsystem`
AaBb123	Variable placeholder, to be replaced with a real name or value.	`formary -subsys` *subsys_name*
`[ ]`	Square brackets contain arguments that are optional.	`dropconnection [-force]`
`|`	The pipe separates arguments, only one of which may be used at a time.	`-ckdrw yes|no`
`...`	The ellipsis indicates that a parameter supports either a list or a range of values.	`-dmod` *u.t.s* `...`
,	The comma indicates that the parameter supports only a list of values (no ranges).	`-aryid` *array_id*`[,`*array_id*`]`
:	The colon indicates that the parameter supports only a range of values (no lists).	`-dates` *startdate*`[:`*enddate*`]`
`C:\>`	Indicates the Windows Command Prompt.	`C:\> sibadmin`
`SIB>`	Indicates the SVAA CLI shell prompt.	`SIB> formaray -subsys ABC1`

Shared Virtual Array Documentation

This section lists software and hardware documentation for the Shared Virtual Array products.

How to Obtain Software Documentation

All of the Shared Virtual Array software publications are available from the following sources:

- On the "Software Publications" CD-ROM (part number 3134524*nn*). To order a copy, contact StorageTek Publication Sales and Service at 800-436-5554 or send a fax to 303-661-7367.
- Online (for viewing and printing), at the StorageTek Customer Resource Center (CRC) website at: `www.support.storagetek.com`. To access the SVAA publications, use the following steps:

  a. Log in.

  Note: Logging in requires a customer login ID and password which can be obtained by calling StorageTek Customer Support at 800-678-4430.

  b. In the navigation bar on the left side of the screen, click **Software** under **Current Products**.

  c. Scroll down to **Virtual Disk (SVA) Software**, and click on the product and platform you want.

  d. The SVAA publications are available under the "Manuals and Guides" heading. You may need to click **View More** to see the complete list.

SVA Administrator for Windows Library

- *Shared Virtual Array Administrator for Windows Command Quick Reference*
- *Shared Virtual Array Administrator for Windows Installation Guide*
- *Shared Virtual Array Administrator for Windows Messages*
- *Shared Virtual Array Administrator for Windows Quick Start Guide*
- *Shared Virtual Array Administrator for Windows User's Guide*

Related SVA Software Publications

SVA Console for Windows NT (SVAC):

- *Shared Virtual Array Console for Windows Quick Start Guide*

For any StorageTek software:

- *Requesting Help from Software Support*

SVA Hardware Publications

Shared Virtual Array hardware publications are available from the following sources:

- On the "SVA Hardware Publications" CD-ROM (part number 3118447*nn*). To order a copy, contact StorageTek Publication Sales and Service at 800-436-5554 or send a fax to 303-661-7367.
- Online (for viewing and printing), at the StorageTek Customer Resource Center (CRC) website at: `www.support.storagetek.com`. To access the Shared Virtual Array publications, use the following steps:

  a. Log in.

  Note: Logging in requires a customer login ID and password which can be obtained by calling StorageTek Customer Support at 800-678-4430.

  b. In the navigation bar on the left side of the screen, click **Disk** under **Current Products**.

  c. Scroll down to **Virtual Disk Storage**, and click on the product you want.

  d. The SVA publications are available under the "Manuals and Guides" heading. You may need to click **View More** to see the complete list.

The V2X Shared Virtual Array (SVA) library consists of:

- V2X *Shared Virtual Array General Information*
- V2X *Shared Virtual Array Installation and Maintenance*
- V2X *Shared Virtual Array Introduction*
- V2X *Shared Virtual Array Operations and Recovery*
- V2X *Shared Virtual Array Planning*
- V2X *Shared Virtual Array Reference*
- V2X *Shared Virtual Array System Assurance*
- *Peer to Peer Remote Copy Configuration Guide*

The V960 Shared Virtual Array (SVA) library consists of:

- V960 *Shared Virtual Array General Information*
- V960 *Shared Virtual Array Installation and Maintenance*

- V960 *Shared Virtual Array Introduction*
- V960 *Shared Virtual Array Operations and Recovery*
- V960 *Shared Virtual Array Planning*
- V960 *Shared Virtual Array Reference*
- V960 *Shared Virtual Array System Assurance*
- *Peer to Peer Remote Copy Configuration Guide*

The 9500 Shared Virtual Array (SVA) library consists of:

- *9500 Shared Virtual Array General Information*
- *9500 Shared Virtual Array Installation*
- *9500 Shared Virtual Array Introduction*
- *9500 Shared Virtual Array Maintenance*
- *9500 Shared Virtual Array Operation and Recovery*
- *9500 Shared Virtual Array Planning, Implementation, and Usage*
- *9500 Shared Virtual Array Reference*
- *9500 Shared Virtual Array System Assurance*
- *Peer to Peer Remote Copy Configuration*

Trademarks

The following terms are trademarks or registered trademarks of Storage Technology Corporation:

- Iceberg
- Shared Virtual Array
- StorageTek
- SVA

The following terms are trademarks of Microsoft Corporation:

- Windows
- Windows NT
- Windows 2000
- Windows 2003

Note: Other terms and product names used herein may be trademarks of their respective owners.

Summary of Changes

Edition E May 2005

This edition includes the following changes:

- Miscellaneous corrections.
- Support for Windows 2003 and discontinuance of Windows NT support.

Fourth Edition (Rev D) February 2003

This edition includes the following changes:

- New point-in-time reports available through the new SVAA server Web-based interface. See the following sections:
  - Chapter 6., “Point-in-Time Reports” on page 67
  - “SVAA Manager Program,” on page 10—enables the Web-based interface.
- New functions allowing user control of SVAA server cache. See the following sections:
  - “SVAA Server Cache,” on page 17
  - `querycache`—Display SVAA server cache settings.
  - `refreshcache`—Force an immediate refresh of all or selected elements of SVAA server cache.
  - `setcacherefreshrate`—Set refresh rates for all or selected elements of SVAA server cache.
- New capability allowing SnapShot copy of all types of Windows 2000 dynamic disk volumes. See Chapter 4., “SnapShot” on page 45.
- New PPRC SnapShot feature. See “PPRC SnapShot,” on page 63.
- New capability allowing SnapShot copy to a PPRC primary volume. See “Snap to Primary,” on page 64.
- New chapter describing PPRC (peer-to-peer remote copy). See Chapter 5., “Peer-to-Peer Remote Copy (PPRC)” on page 51. Includes the following new information:
- New appendix providing definitions and calculations for all SVAA display and query fields. See Appendix C., “Interpreting SVAA Displays and Queries” on page 221.

Third Edition (Rev C) September 2002

This edition includes the following changes:

- V2X Shared Virtual Array support, including the following features (see *V2X Shared Virtual Array General Information* for a full list of V2X features):
  - 8, 16, and 32GB cache
  - 36GB drive modules
  - Support for up to 16 SSIDs (subsystem identifiers)
  - Support for up to 4096 3390-3 devices (ESCON and fibre attach)
  - Support for up to 1365 3390-9 devices (ESCON and fibre attach)
  - Added V2X documents to "SVA Hardware Publications" section.
- `alterdevice`—Added `-devpath` parameter. Allows you to specify a functional device by its host-specific device path, as an alternative to specifying its FDID (functional device ID).
- Deprecation of 5+2 array. The 5+2 array size is not valid for V2X nor V960 subsystems. Made changes to the following commands:
  - `altersubsystem`—5+2 array size not valid for V2X nor V960 subsystems.
  - `formarray`—5+2 array size not valid for V2X nor V960 subsystems. `-arysi` parameter eliminated.
- PPRC open support. Made changes to the following commands:

  Note: Only certain SVAA platforms and SVA subsystem configurations support the use of PPRC commands and functions. Contact StorageTek Software Support for details.
  - `alterdevice`—New parameters for establishing, modifying, and deleting PPRC pairs.
  - `alteriointerface`—New parameters for establishing PPRC paths.
  - `definedevice`—New parameters for establishing bridge volumes and PPRC pairs.
  - `deletedevice`—Delete fails if device is part of a PPRC pair.
  - `displaydevice`—Display PPRC data.
  - `displayiointerface`—New parameter for displaying PPRC path information.
  - `dropadministratorgroup`—New parameter to remove the PPRC administrator group.
  - `queryadministratorgroup`—Displays PPRC administrator group.
  - `queryconnections`—Displays PPRC administrator group information.

  - querypermissions—Displays PPRC administrator group information.
  - release—New release restrictions for PPRC devices.
  - setadministratorgroup—New parameter to specify a PPRC administrator group.
  - snap—New snap restrictions based on PPRC devices.
- Added ability to modify some characteristics of a SCSI larger LUN device. See the alterdevice command for details.
- Support for SnapShot of Windows spanned and striped disks. See the snap command for details.
- Updated documentation to explicitly show that the domain is part of a user name (this is a clarification to the documentation only).

Chapter 1. Overview

What is SVAA?

Shared Virtual Array Administrator (SVAA) for Windows is a standard Windows service that helps you manage Shared Virtual Array (SVA) subsystems from a Windows computer. Under Windows, SVAA supplements other disk utility programs to provide extended facilities for configuring and administering SVA subsystems.

Access to SVAA is through the CLI (command line interface) and the separately provided Shared Virtual Array Console (SVAC) for Windows.

Note: In this book, Windows 2000 Server and Windows 2003 Server are referred to collectively as Windows. In the instances where there are differences in SVAA function, the specific operating system is identified.

Related Products

SVAA works in association with the following products:

- Shared Virtual Array (SVA) subsystem—An SVA subsystem is a StorageTek RAID storage device that uses a large number of small disks operating under a virtual storage architecture. UNIX, Windows, or S/390 host computers can store and retrieve data on the SVA subsystem via fibre and/or ESCON connections.
- Shared Virtual Array Console for Windows—A GUI interface providing SVA subsystem configuration, administration, and reporting functions.
- Shared Virtual Array Administrator products for Solaris, HP-UX, AIX, Linux Intel, and S/390—Software products providing additional platform support for SVAA. See the associated product manuals for details.

Figure 1-1, on page 2 illustrates how these products work together to provide a complete data storage solution.

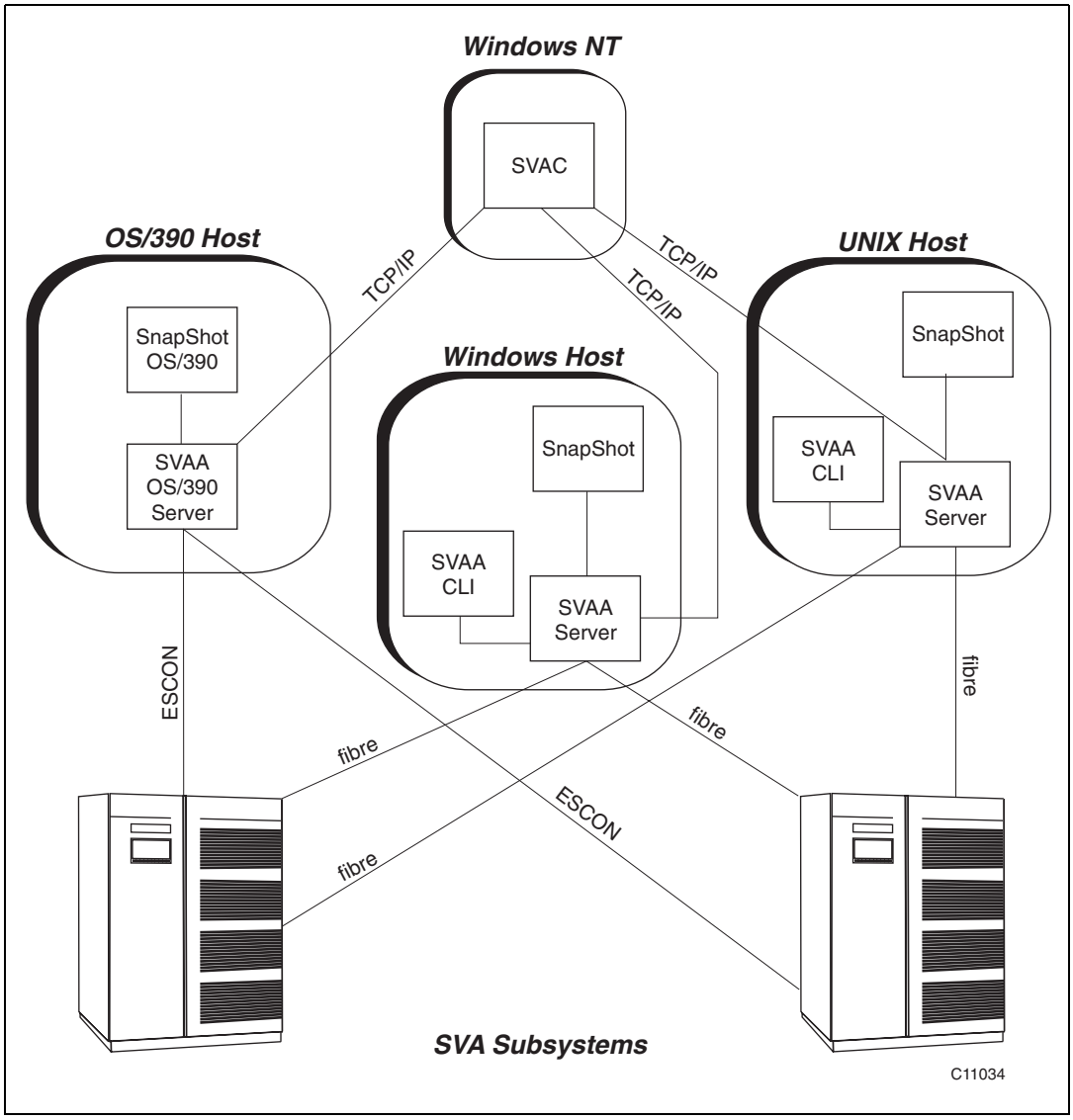


Figure 1-1 Shared Virtual Array hardware and software products

SVAA also works with the following products:

- SVA NMP Agent for AIX, HP-UX, Solaris, and Windows and SVA NMP Monitor for Windows—Software products providing extensive options for monitoring the status of the SVA subsystem and SVAA server.
- SVA Path for AIX, HP-UX, Solaris, and Windows—Software products providing failover, failback, and load-balancing capabilities for the SVA subsystem.

SVAA Features

SVAA for Windows provides the following features:

- Single-point storage administration control of virtual disks in the SVA subsystem
- SVA subsystem configuration
- ShapShot™ copy of SCSI partitions
- Configuration of the peer-to-peer remote copy utility
- SCSI partition release
- Ability to view information about SVAA server connections and tasks
- Command line interface (CLI)

SVA Subsystem Configuration

Using SVAA, you can define the functional devices in an SVA subsystem by specifying their type, size, and other characteristics. You can configure an SVA subsystem to support a co-located environment, which includes both SCSI and CKD (count-key-date) device emulation. This allows a single SVA subsystem to be accessible from UNIX, Windows, and S/390 hosts.

You can use SVAA to reconfigure the SVA subsystem, adding functional devices "at will" to satisfy any application requirement. See Chapter 3., "Storage Subsystem Management" for details.

Functional Differences Between Subsystem Models

Earlier SVA subsystem models supported some functions that are not available on the V2X Shared Virtual Array. SVAA 3.1 is compatible with these earlier products, therefore the functions are documented in this manual for compatibility. Table 1-1, "Functional differences between SVA subsystems" summarizes these differences.

Table 1-1 Functional differences between SVA subsystems

Function	9500 SVA	V960 SVA	V2X SVA
Array sizes available:	5+2, 13+2	13+2	13+2
Maximum number of FDIDs:	1024 3390-3s or 341 3390-9s	1024 3390-3s or 341 3390-9s	4096 3390-3s or 1365 3390-9s
Maximum number of SSIDs:	4	4	16

SnapShot of SCSI Partitions

SnapShot offers high-speed data duplication without the need for increased disk capacity. SnapShot duplicates SCSI partitions by creating a second set of pointers to the data; it does not physically move any data. The two independent sets of pointers treat the data as if it were two physically separate images. Updates can occur simultaneously both to the original data and to the duplicate, with no concern for compromising the data in either.

The speed of SnapShot makes the data available to an application in a much shorter time than traditional duplication methods. Moreover, SnapShot operations are

performed entirely within the SVA subsystem; they do not consume resources in the host operating system, bus, or host bus attachment (HBA).

SCSI Partition Release

The SCSI partition release facility informs an SVA subsystem that a given SCSI partition no longer contains any useful data, so the subsystem can release the space the partition is using. Thus, partition release enables you to take better advantage of an SVA's extended capacity, to achieve lower overall storage costs, and at the same time to securely erase unwanted data.

See "Space Release," on page 43 for details.

Peer-to-Peer Remote Copy (PPRC)

Peer-to-peer remote copy (PPRC) is a hardware solution supported by SVAA that enables any data (system or application) to be mirrored between pairs of disks. Once a PPRC pair is established, all data from the device to be mirrored (the primary volume) is copied to the secondary volume. From that point on, all updates to the primary volume are copied synchronously to the secondary volume, so that at any point in time the secondary volume is an exact mirror of the primary volume.

See Chapter 5., "Peer-to-Peer Remote Copy (PPRC)".

CLI (Command Line Interface)

The SVAA CLI (command line interface) enables you to issue SVA subsystem management commands from a Windows server.

See Chapter 7., "SVAA CLI Commands" for details about the CLI and its commands.

SVAA Concepts

The following discussion of certain SVA and SVAA concepts may be useful as you use succeeding chapters of this book. For complete definitions of terms, see the "Glossary" at the back of this book.

Virtual Storage

SVA's virtual storage architecture extends the capabilities of traditional online storage. It allows you to define the disk storage system configuration in terms of traditional SCSI or CKD disks, but these disks are virtual and do not exist as physical devices.The data stored on an SVA subsystem is dynamically mapped across a collection of physical devices organized into a logical group called an "array." Mapping between virtual devices and the SVA subsystem is accomplished with tables of pointers located in the SVA subsystem control unit.

SVA's virtual storage architecture allows you to define up to 1024 virtual devices on a V960 subsystem, 4096 on a V2X, independent of subsystem capacity.

Drive Module

Within the SVA subsystem, the individual physical disk drives are called drive modules. See "Drive Modules," on page 35 for details.

Functional Device

A functional device is a virtual disk, as viewed by the host operating system, the applications, and the users. A functional device can emulate one of a variety of SCSI and CKD (count-key-data) disk devices.

See "Functional Devices," on page 38 for details.

SCSI Larger LUN Device

A SCSI larger LUN device consists of one or more functional devices that are viewed by the host system as a single logical device. A larger LUN device is composed of one parent device (the index 0 device), and one or more child devices (devices 1–n).

See “SCSI Larger LUN Devices,” on page 39 for details.

ECAM Device

An ECAM device is a functional device over which SVAA command and control messages are sent to the SVA subsystem. The protocol used by SVAA and the SVA subsystem is called extended control and monitoring (ECAM), hence the term “ECAM device.”

A “privileged” ECAM device, a special type of ECAM device, is used to control changes in the SVA subsystem’s state.

See “ECAM Devices,” on page 33 for additional details.

Domain

A domain can be thought of as a logical SCSI bus. Each controller in the Windows computer is connected to a domain, giving it access to the devices that have been configured within that domain. There can be up to 16 SCSI domains per SVA subsystem (domain IDs 0–15).

Only one host type can be connected to any one domain; that is, two hosts with different operating systems cannot share the same domain. When defining a SCSI functional device, you must specify its domain, target, and LUN. Each domain can have up to 256 devices (although on V960 and earlier subsystems, not all domains can have the maximum 256 devices because the total number of devices per SVA subsystem is 1024).

Figure 1-2, “SCSI domains” illustrates a possible multiple-domain configuration.

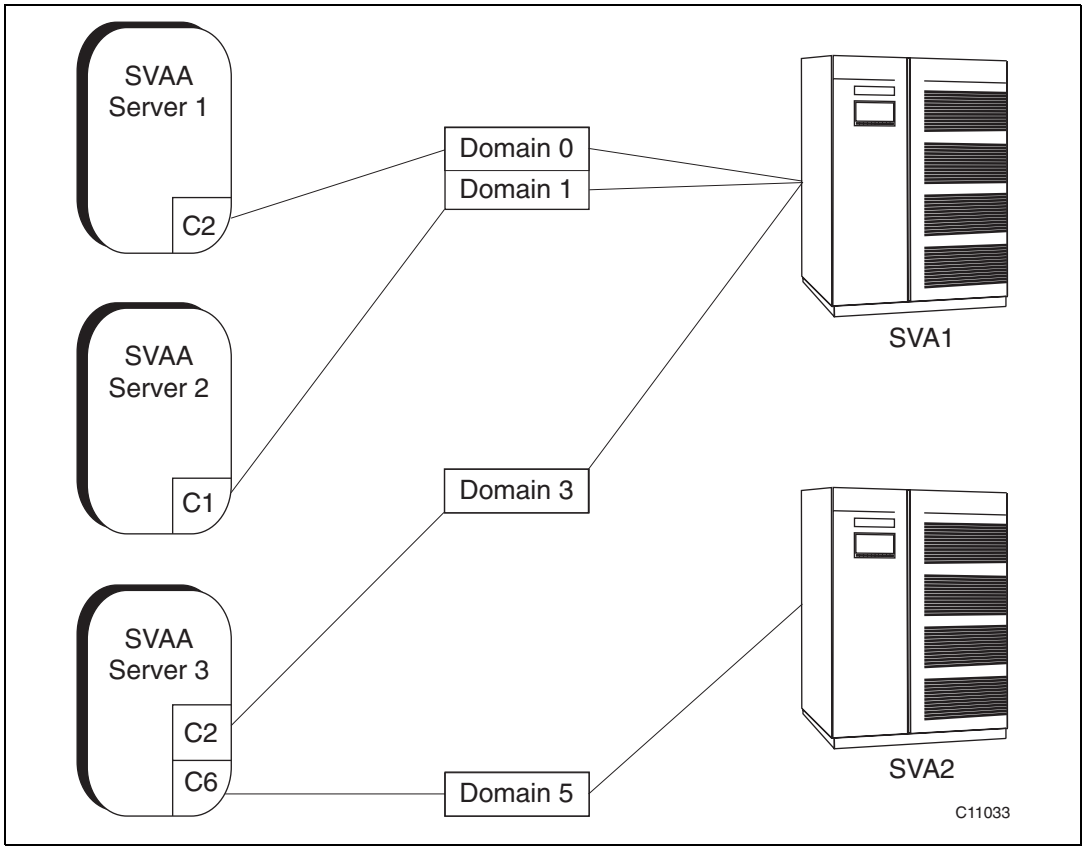


Figure 1-2 SCSI domains

Subsystem Partitions

Within the SVA subsystem, the term "partition" indicates the current use or availability of a drive module—not a physical location. It should not be confused with the concept of a "SCSI partition."

Depending upon its status, each drive module belongs to one of the following SVA subsystem partitions:

- Production—Drive modules assigned to production arrays for storing user data.
- Spares—Drive modules not logically associated with an array. These drive modules are available for forming new arrays.
- MAT—Drive modules not yet available for storing user data.
- Unavailable—Drive modules not available for use in an array, including failed and uninstalled drive modules.

See Appendix A., "Drive Module Status Codes" for additional details.

Minimum Configuration Steps You Must Perform at the LOP

Before you begin defining and updating your SVA subsystem using SVAA, you must use the subsystem LOP (local operator panel) to define a minimum subsystem configuration. The minimum configuration requires the following steps:

1. Configure and enable at least one I/O interface accessible to the host operating system.
2. Assign a subsystem name to the SVA subsystem.
3. Form at least one array in the Production partition.
4. Define at least one privileged ECAM device.

Note: If your installation includes S/390 hosts, you must also define up to four (in the case of V960) or 16 (in the case of V2X) SVA subsystem identifiers (SSIDs) that match what has been defined on the S/390 hosts.

Chapter 2. Server Management

Overview

The SVAA server runs as a standard Windows service under the local System Account. It can be configured to start up automatically during Windows startup or only on-demand. SVAA configuration information is stored in the Windows Registry.

As a service running under the local System Account, SVAA does not have access to system resources. The SVAA software must be installed on a local disk, and all files it accesses must also be on a local disk.

SVAA does not have any special privileges beyond those assigned to it as a member of the System Account.

SVAA Configuration

To configure SVAA, you use the SVAA Manager which is installed during SVAA installation. You are required to define an initial server configuration during SVAA installation. You can also reconfigure an SVAA server at any time with the SVAA Manager. See "SVAA Manager Program," on page 10 for details.

Multiple SVAA Server Configurations

You can use the SVAA Manager to define multiple SVAA configurations. Each configuration is automatically assigned a unique service name. The service name is used to register the service when it is started up. This allows you to run multiple copies of the SVAA service on the same computer. Each server configuration should have a unique name, TCP/IP port number, and path to the server configuration file.

Windows Registry

The server configuration information you specify in the SVAA Manager is stored in the Windows Registry. Each SVAA server configuration is assigned a unique Registry key name. When you start an SVAA server, the SVAA Manager reads the Registry key and uses the values defined there to configure the server. Optionally, you can use runtime parameters on the SVAA Manager to override some of the values defined in the Registry, such as server port number and log file name. (See "Service Tab," on page 12 for details).

Caution: It is recommended that you not edit the Windows Registry, except under the instruction of a StorageTek Software Support representative. Unauthorized modifications could have unpredictable results.

Server Configuration File (config.dat)

A copy of the server configuration data in the Windows Registry is maintained in a file named `config.dat`. The `config.dat` file serves strictly as a backup to the Windows Registry; the data in the Registry always takes precedence. You specify the location of the `config.dat` file when you set up the server configuration with the SVAA Manager. A typical location would be `C:\Program Files\StorageTek\Shared Virtual Array Administrator 3.1.0\Server31`). If multiple SVAA servers are installed on the same computer, there is one `config.dat` file per server, each one located in its own folder.

Caution: Any user who can read the `config.dat` file can potentially bypass SVAA server security. SVAA server startup parameters allow you to bypass Registry entries and use the `config.dat` file for startup options. It is recommended that you not edit the file directly, except under the instruction of a StorageTek Software Support representative. Unauthorized modifications could have unpredictable results.

SVAA Manager Program

The SVAA Manager program is used to configure, start, and stop the SVAA server. To run the SVAA Manager, you must be logged on as a user with Windows administrator privileges.

Starting the SVAA Manager

During SVAA installation, the SVAA Manager program is added to the Windows **Start** menu. To start the SVAA Manager click **Start**, and then click **Programs>StorageTek>Shared Virtual Array Administrator 3.1.0>SVAA Manager**.

SVAA Manager Dialog Box

The SVAA Manager dialog box has two tabs:

- **Server**—Used to create and maintain SVAA server configurations.
- **Service**—Used to set startup options for each defined SVAA server configuration.

Server Tab

Following are descriptions of the fields on the **Server** tab dialog box.

Name:

Each SVAA server must have a unique name. This name is displayed during server execution.

Enter any string up to 256 characters in length. The following characters are accepted, even as the first one: a to z, A to Z, 0 to 9, $, @, #, -, _, +, &, ., and /.

Caution: You must not have two SVAA servers with the same name, even if they are running on different hardware platforms, as this can have unpredictable results.

Port:

The TCP/IP port number for this SVAA server configuration. The default is 41248.

Enter any integer between 1024 and 64567.

Note: This value should be unique for each SVAA server configuration.

Install Path:

The folder where the SVAA server software is installed. Typically, this would be `C:\Program Files\StorageTek\Shared Virtual Array Administrator 3.1.0\`.

Enter the full pathname of the folder, such as `C:\Program Files\StorageTek\Shared Virtual Array Administrator 3.1.0\`. The folder must already exist. The path must be on a local drive, as the SVAA server cannot access any network resources.

Config Path:

The folder where the `config.dat` file is located. This file contains the values you define for this SVAA server configuration, and it serves as a backup to the Windows Registry.

Enter a full pathname of the folder, such as `C:\Program Files\StorageTek\Shared Virtual Array Administrator 3.1.0\Server31\`. The folder must already exist. The path must be on a local drive, as the SVAA server cannot access any network resources.

Note: This value should be unique for each SVAA server configuration.

Log File:

The name of the file where the SVAA server will write error messages.

Enter the full pathname, including the file name, such as `C:\Program Files\StorageTek\Shared Virtual Array Administrator 3.1.0\Server31\Server31log`. The folder where the file will be located must already exist. The path must be on a local drive, as the SVAA server cannot access any network resources.

Note: This value should be unique for each SVAA server configuration.

Enable Web Interface

Indicates whether you want to enable the SVAA server WBI. The WBI is used to produce point-in-time reports for one or more SVA subsystems.

A check mark enables the WBI; a blank check box disables it.

Web Interface Port:

The TCP/IP port number to be used for communication between the WBI and the SVAA server.

Enter any integer between 1024 and 57602.

Note: This value must be unique for all SVAA server configurations on the host system. It also must be different from the **Port** value.

All Devices:

Lists all fixed disks on the host. This includes local hard drives as well as SVA subsystem devices. Network drives and other devices (such as floppy drives and CD-ROM drives) are not included in the list.

ECAM Devices:

Lists devices that have been designated as ECAM devices for the server. ECAM devices are used by the SVAA server to communicate with the SVA subsystem.

Note: To move a drive between the All Devices and ECAM Devices scroll lists, select the drive letter and double-click it or click an arrow (<< or >>). To move all drive letters from one scroll list to the other, Shift-click an arrow.

Note: See "Installing SVA Path with SVA Administrator" in the *SVA Path for Windows User's Guide* for instructions on updating the ECAM device configuration in SVA Path.

Service Tab

Following are descriptions of the fields on the **Service** tab dialog box.

Name:

Displays the name of the selected SVAA server configuration Registry key. This value is generated automatically. It has the form `SVAA_StorageTek_`***nn***, where *nn* is a two-digit number.

This field is display only.

Description:

A description of the SVAA server configuration. This name will be displayed in the Services dialog. It is also displayed in the SVAA server list box.

Enter any string up to 256 characters in length. The following characters are accepted, even as the first one: a to z, A to Z, 0 to 9, $, @, #, -, _, +, &, ., and /.

Startup Type:

Specifies whether this SVAA server service will be started manually or automatically.

Click **Manual** to allow the service to be started manually on demand. Click **Automatic** to have the service started automatically during the Windows startup process.

Status:

Displays the current status of the SVAA service. Normal values are `Stopped` or `Running`. For short periods of time the service can be in an intermediate state, such as `Starting` or `Stopping`. Abnormal states include `Pausing`, `Paused`, `Continuing`, or `Unknown`. Also, an abnormal condition exists if the service is in an intermediate state for more than a minute or so.

This field is display only.

Parameters:

Provides a place to enter optional parameters to be passed to the SVAA server as it starts up. The parameters are described below.

Note: Parameter values containing spaces must be enclosed in quotes.

`-config` *pathname*

Optional parameter. Specifies the location of the `config.dat` file to be used by the SVAA server. When the server starts up, it will use the values specified in this file instead of those in the Windows Registry which were defined during SVAA configuration.

Enter the full pathname where the `config.dat` file is located.

`-debug` *format*

Optional parameter. Activates trace logging on the server side for diagnostic purposes.

format (an integer from 1 to 7) specifies the format for the trace entries. You can enter only one value at a time. Ranges, lists, and wildcards are not supported.

Note: Normally, you would use this parameter only under the direction of StorageTek Software Support. Your support representative will give you the appropriate *format* value to use.

`-ecamdev` *device_path*

Optional parameter. Specifies the ECAM device to be used by the SVAA server to access the SVA subsystem. This entry overrides the default ECAM device specified in the Windows Registry.

Entry must be a valid ECAM device previously defined with the `addsubsystempath` command. Enter the full pathname of the device.

`-file`

Optional parameter. Used for Windows-based SVAA servers only. Specifies to repopulate the Windows Registry with values from the SVAA configuration file (`config.dat`). This is useful for restoring the SVAA configuration information in the Windows Registry in the event that the Registry data has been corrupted in some way.

The file to use is determined as follows:

– If you are using the `-config` parameter on this command to explicitly specify a `config.dat` file, the values will be taken from there.

– If you are not using the `-config` parameter, the values will be taken from the default `config.dat` file for this server configuration, as specified in the **Config Path** field on the **Server** tab dialog box.

`-log` *file_name*

Optional parameter. Specifies the log file to be used by the SVAA server (where SVAA error messages are sent). This entry overrides the log file specified in the Windows Registry.

Enter the full pathname, including the file name.

`-nowebport`

Optional parameter. Disables the WBI for this SVAA server. This entry overwrites the WBI status in the Windows Registry, disabling the WBI for this SVAA server.

Once this parameter is used, the WBI remains disabled for this SVAA server until the interface is explicitly re-enabled.

`-port` *port_number*

Optional parameter. Specifies the port number to be used by the SVAA server's TCP/IP connections. This entry overrides the port number in the Windows Registry.

Enter any integer between 1024 and 64567.

Note: If you have multiple SVAA servers running on a single host machine, each server must have a unique port number.

`-serveridletime` *minutes*

Optional parameter. Specifies that the SVAA cache refresh cycle is to be suspended if there is no cache activity for the specified number of minutes. The cache refresh cycle will be restarted as soon as cache activity is resumed. "Cache activity" is defined as any request that queries or modifies data in the SVAA cache, such as `displaydevice`, `startdrain`, or `refreshcache`; commands such as `queryversion` and `setadministratorgroup` do not affect SVAA cache and therefore do not affect this option.

Enter the number of minutes; entry can be any integer between 1 and 10080 (seven days). You must specify a value; there is no default.

If you do not specify this parameter, SVAA server cache refreshes will never be suspended for this instance of the server.

`-webport` *port_number*

Optional parameter. Specifies the TCP/IP port number to be used for communications between the SVAA server and the WBI. This entry overrides the port number in the Windows Registry, making it the new default WBI port number for this SVAA server..

Note: This entry must not be the same as the port number of any active SVAA servers (SVAA server port numbers are defined in the Windows Registry and can be overridden by the `-port` parameter).

Enter any integer between 1024 and 64567.

Note: If the WBI was previously disabled for this SVAA server, specifying this parameter re-enables it. The WBI will remain enabled until explicitly disabled.

SVAA Server Administration

This section describes how to use the SVAA Manager program and the CLI to perform typical SVAA server administration tasks.

Defining a New SVAA Server Configuration

Use the following steps to create a new SVAA server configuration.

1. Start the SVAA Manager.
2. Click the **Server** tab.
3. Click **New**.
4. Define the new configuration by filling in the fields on the dialog box. See “Server Tab,” on page 10 for details.
5. Click the **Service** tab.
6. Enter a value in the Description field. See “Service Tab,” on page 12 for details.
7. Click **Save**.

The new SVAA server configuration will appear in the SVAA server list box, on the right-hand side of the dialog box. You can start the server by following the steps in “Starting the SVAA Server”.

Modifying an SVAA Server Configuration

Use the following steps to make changes to an existing SVAA server configuration.

1. Start the SVAA Manager.
2. Click the **Server** tab.
3. In the SVAA server list box, select the configuration you want to modify.
4. Make the changes by editing the fields on the dialog box. See “Server Tab,” on page 10 for details.
5. Click **Save**.

Deleting an SVAA Server Configuration

Use the following steps to delete a server configuration from the SVAA server list box. This does not in any way delete or de-install the SVAA server software on your computer.

1. Start the SVAA Manager.
2. Click the **Server** tab.
3. In the SVAA server list box, select the configuration you want to delete.
4. Click **Delete**.

Starting the SVAA Server

Use the following steps to start the SVAA server.

1. Start the SVAA Manager.
2. Click the **Service** tab.
3. In the SVAA server list box, select the SVAA server configuration you want to start.
4. Click **Manual** or **Automatic** to indicate whether or not you want the SVAA server to start up automatically during Windows startup.
5. Optionally, you can pass startup parameters to the SVAA server by typing the parameters in the **Parameters** box. The parameters will apply to this instance of the SVAA server only.
6. Click **Start**.

It is possible to have multiple SVAA servers running on one host. Typically, you would do this when installing a new version of the SVAA server software. You could keep the old version running in a production mode while installing, configuring, and verifying the new version. Each server has a unique TCP/IP port number.

Stopping the SVAA Server

Use the following steps to stop the SVAA server gracefully.

1. Start the SVAA Manager.
2. Click the **Service** tab.
3. In the SVAA server list box, select the SVAA server configuration you want to stop.
4. Click **Stop**.

In-process requests are completed, and pending requests are dropped. All users currently connected to the server are dropped, and the SVAA service is stopped.

Caution: Do not use any other method to stop the SVAA server, as this can have unpredictable results.

Displaying Server Information

The `displayserver` command displays general information about the SVAA server, including server name, software version, installed features, host name and operating system level, and TCP/IP port number.

The `queryversion` command displays just the SVAA software version.

Displaying Users Connected to the Server

You can use the `queryconnections` command to display a list of userids currently connected to the SVAA server.

Disconnecting a User From the Server

You can use the `dropconnection` command to disconnect an individual user or all users from the SVAA server. To drop an individual user, you must specify that user's connection identifier, which is displayed with the `queryconnections` command.

Related CLI Commands

Table 2-1, “Server administration commands” lists commands you can use to administer the SVAA server. See “SVAA CLI Command Reference (Help Files),” on page 89 for detailed descriptions of the commands and their parameters.

Table 2-1 Server administration commands

Command	Function
`displayserver`	Displays general information about the SVAA server.
`dropconnection`	Disconnects specified users from the SVAA server.
`queryconnections`	Displays userids currently connected to the SVAA server.
`queryrequests`	Displays pending SVAA server requests.
`queryversion`	Displays the SVAA server software version.

SVAA Server Cache

The SVAA server maintains a cache of information about the overall SVA subsystem, I/O interfaces, physical devices, and functional devices. The server keeps a separate cache for each SVA subsystem to which it has access.

Note: The SVAA server cache should not be confused with SVA subsystem cache. The latter is used in the transfer of user data to and from the SVA subsystem.

The SVAA server cache includes the following elements:

Cache Element	Type of Information Included [a]
Array status	Array drain status.
Functional device performance	Device capacity, number of requests, amount of data transferred, time available.
Functional device properties	Device characteristics which vary depending on the type of device. May include device DTL, device type, read/write enabled status, ECAM privileged status.
Functional device space	Device tracks available and used.
I/O interface performance	Speed, number of I/Os, amount of time busy, etc.
I/O interface properties	Name, type, status.
Physical device performance	Busy time, number of bytes transferred for read/write operations.
Physical device status	Location, status, array assignment, capacity.

Cache Element	Type of Information Included [a]
Subsystem performance	Free space, NCL, back-end capacity, SVA cache sizes, number of ECAM messages processed and bypassed.
Subsystem properties	Installed features, general information, such as number of devices, number of I/O interfaces supported.

a. Representative of the types of information, but not all-inclusive.

Note: PPRC (peer-to-peer remote copy) properties are not included in SVAA server cache.

Why the SVAA Uses a Cache

The SVAAcache helps to optimize overall system performance. By keeping some information in its cache, the SVAA reduces the number of times it must access the SVA subsystem to obtain information about subsystem components. This improves request response time, particularly for display requests, such as `displaydevice`, `displayiointerface`, `displaysubsystem`, etc., and the point-in-time reports (see Chapter 6., "Point-in-Time Reports" for details).

How the Cache is Updated

The SVAA server cache can be updated in any of the following ways:

- As part of a periodic cache refresh. By default, the SVAA server automatically refreshes all its cache elements every 300 seconds (five minutes). You can use the `setcacherefreshrate` command to override this rate for some or all cache elements to better meet the needs of your installation. See "Changing Cache Refresh Rates," on page 19 for details.
- By user intervention. You can use the `refreshcache` command to force the immediate refresh of some or all cache elements. This can be done at any time. See "Forcing an Immediate Cache Refresh," on page 19 for details.
- Through an SVA subsystem update initiated by the SVAA server. Anytime an SVAA server causes a change to an SVA subsystem component, that server's cache is immediately updated with the new information. For example, if a user on Server1 initiates an array drain, Server1's cache is immediately updated as the array status changes from "Pending," if applicable, to "Draining," to "Active." The cache of any other SVAA servers is not updated, however, until one of the two other update methods occurs. See "Cache Update Issues" for additional information.

Cache Update Issues

An SVAA server's cache is not immediately updated when an SVA subsystem update is initiated by another server or the LOP (local operator panel). Therefore, the cache of all servers may not always reflect the exact status of the SVA subsystem at any given point in time.

In situations where it is critical that an SVAA server's cache be current, you can do either of the following:

- Turn caching off for the elements in question. See "Turning Off Cache" for details.

- Perform an immediate cache refresh. See “Forcing an Immediate Cache Refresh” for details.

Changing Cache Refresh Rates

By default, all cache elements for all SVA subsystems are refreshed every 300 seconds (five minutes). The `setcacherefreshrate` command allows you to change specified refresh rates to any frequency, ranging from one minute to 24 hours.

You can set different cache rates for different elements and/or different SVA subsystems. You can also turn some or all caching off. See “Turning Off Cache” for details.

Turning Off Cache

The `-off` option on the `setcacherefreshrate` command allows you to turn off the SVAA server cache for specified cache elements and/or specified SVA subsystems. If a cache element is turned off, the SVAA will always request that information directly from the SVA subsystem.

Note: Turning off cache may significantly affect request response times; therefore, it should be done with care.

You can turn caching on again by using the `setcacherefreshrate` command to re-assign a refresh rate.

Note: Turning off SVAA cache is not the same as suspending cache refreshes. See “Suspending Cache Refreshes” for a description of that activity.

Displaying Cache Refresh Rates

The `querycache` command displays the current cache refresh rate settings. You can choose which elements and/or which SVA subsystems you want to display.

Forcing an Immediate Cache Refresh

The `refreshcache` command immediately refreshes selected cache elements. You can use this command when it is critical that the SVAA server cache be up-to-date. For example, you can use this command before performing a display command to ensure that the display will reflect the most current SVA subsystem information.

Suspending Cache Refreshes

When the SVAA server refreshes its cache it performs I/O to the SVA subsystem, thus consuming system resources. You can potentially use fewer system resources by optionally instructing the server to suspend cache refreshes during periods when there is no cache activity (“cache activity” is defined as any request that queries or modifies data in the SVAA cache).

To control SVAA cache suspensions you start the server with the `-serveridletime` option on the `sibserv` command. This option instructs the SVAA server to suspend refreshing its cache if it receives no cache-related requests for a specified period of time. If this time threshold is reached, the SVAA server stops refreshing its cache until the next time it receives a cache-related request, at which time the server performs I/O to the subsystem to refresh its cache and resume the cache refresh cycle.

Note: Suspending SVAA cache refreshes is not the same as turning off the cache. See “Turning Off Cache” for a description of that activity.

Related CLI Commands

Table 2-2, "Server cache commands" lists commands you can use to modify SVAA server cache refresh rates. See "SVAA CLI Commands," on page 85 for detailed descriptions of the commands and their parameters.

Table 2-2 Server cache commands

Command	Function
`querycache`	Displays SVAA server refresh rates.
`refreshcache`	Forces an immediate refresh of selected cache elements.
`setcacherefreshrate`	Changes the refresh rate of selected cache elements.
`sibserv`	Instruct the SVAA server (with the `-serveridletime` option) to suspend cache refreshes if there is no cache activity for a specified period of time.

SVAA Server Log

What It Is

The SVAA server log is a record of important server events (such as startup and shutdown) and abnormal conditions (errors), as well as a time-stamped "heartbeat" entry approximately every 15 seconds while the server is running. The log is an important tool for monitoring the status of the SVAA server and the SVA subsystem.

Server log entries are written to a disk file (see "Server Log File Name and Location" for details). New entries are added to the end of the file so that existing ones are not overwritten. In addition to the disk file, the 1,000 most recent server entries are stored in the server's memory.

The messages written in the SVAA server log are documented in the *SVAA for Windows Messages* manual. You can refer to that book for a detailed explanation of each message and possible actions you should take.

Interface With the Windows Event Log

All SVAA server error and informational messages are included in the server log. In addition, the most critical messages (message categories E, S, F, and T) are also sent to the Windows Applications Event Log. This allows you to use the Event Viewer as well as the server log to monitor critical SVAA server events.

Server Log File Name and Location

The default SVAA server log file location is identified in the Windows Registry. You define this with the SVAA Manager program.

If, for some reason, no log file name is found in the Windows Registry, the log will be created as `bhlog.txt` in the current folder where the SVAA server is started.

Note: Optionally, you can direct messages to a different file by using the `-log` parameter in the SVAA Manager when you start the server. The log file must always be on a local drive, as the SVAA server cannot access any files on network drives.

See “SVAA Manager Program,” on page 10 for details.

Viewing the Server Log

It is important that you periodically check the SVAA server log for error messages so you can take appropriate action.

You can use the `queryserverlogentries` command to display the 1,000 most recent server log entries which are stored in memory. The entries can be displayed by log level, search string, and/or time of occurrence.

Note: To view the log file stored on disk (which may contain more than 1,000 entries), you can use a tool such as `Quick View` to display the file at your terminal. You can display the file while the SVAA server is running. Although it is possible to use an editing tool (such as Notepad) to view the log file, this is not recommended, as you could inadvertently modify the file while you are looking at it. Modifying the file can cause some messages to be lost if the server tries to write to the file while you have it open.

Managing Server Log Files

Log entries are always appended to the log file without overwriting existing ones. So, for example, if you shut down and later restart the SVAA server, messages from the second server session will be appended to the log file that already existed from the first session.

The SVAA server does not monitor the size of the log file. Therefore, you will need to monitor the file size yourself and use some sort of aging strategy to keep the file from becoming too large. One strategy is to periodically move the existing log file to a new location. You can move the file while the SVAA server is running. The server will then create a new log file in the appropriate location when it logs the next message.

Caution: You must not let the log file fill up, as this would fill up the file system and possibly stop the SVAA server.

Multiple Servers on a Computer

If you are running more than one SVAA server on a single computer, it is recommended that you define separate log files for each. It is possible to have more than one SVAA server writing to the same log file, but this is not recommended because there is no built-in mechanism for associating a particular message to a particular server.

Related CLI Commands

Table 2-3, “Server log commands” lists commands you can use to manage the SVAA server log. See “SVAA CLI Command Reference (Help Files),” on page 89 for detailed descriptions of the commands and their parameters.

Table 2-3 Server log commands

Command	Function
`queryserverlogentries`	Displays the SVAA server log.

Client Log

What It Is

The SVAA client log contains a record of important CLI (command line interface) events (such as startup and exit) and abnormal conditions (errors). The log is an important tool for monitoring the status of CLI sessions. Up to 1,000 entries are contained in the log, which is stored in the client's memory, not written to disk.

The messages written in the SVAA client log are documented in the *SVAA for Windows Messages* manual. You can refer to that book for a detailed explanation of each message and possible actions you should take.

Interface With the Windows Event Log

All CLI error and informational messages are included in the client log. In addition, the most critical messages (message categories E, S, F, and T) are also sent to the Windows Applications Event Log. This allows you to use the Event Viewer as well as the SVAA client log to monitor critical CLI events.

Viewing the CLI Log

It is important that you periodically check the SVAA client log for error messages so you can take appropriate action.

You can use the `queryclientlogentries` command to display the 1,000 most recent CLI log entries which are stored in memory. The entries can be displayed by log level, search string, and/or time of occurrence.

Related CLI Commands

Table 2-4, "Client log commands" lists commands you can use to manage the SVAA client log. See "SVAA CLI Command Reference (Help Files)," on page 89 for detailed descriptions of the commands and their parameters.

Table 2-4 Client log commands

Command	Function
`queryclientlogentries`	Displays the SVAA client log.
`sibadmin`	Starts the CLI and begins logging to the client log.

Security

The SVAA security system controls the authentication of users logging on to the SVAA server, as well as which requests a user can submit to the SVAA server from either the CLI or the SVA Console (SVAC).

When SVAA security is in full force, the system works as follows:

- Whenever a user submits a request through the CLI or the SVA Console, the system compares the user's privileges to the authorization criteria for the request.
- If the user meets the criteria, the request is processed; if the user does not meet the criteria, the request fails with an error message.

The major components of the security system are:

- User privileges—defines what privileges a user holds.

- Request authorization criteria—defines what privileges a user must have in order to be able to run a particular command.
- Security modes—defines the level of authorization checking performed by the SVAA server.

These components are described in the following sections.

User Privileges

There are the following types of SVAA users:

- General users
- SVAA administrators
- PPRC administrators

In most situations, you would run SVAA with an SVAA administrator group, although it is possible to run without one. If you are using PPRC, you would also run with a PPRC administrator group defined, although it is possible to run without one.

What General Users Can and Cannot Do

Anyone with a valid userid and password on the computer where the SVAA server is running is considered at least a general user. General users cannot run SVAA administrator commands. Additionally, if a PPRC administrator group has been defined (see "PPRC Administrators," on page 25 for details), general users cannot run PPRC administrator commands. General users can run commands for which they have sufficient Windows read/write file permissions. See "Request Authorization Criteria," on page 26 for more details.

For example, in order for any user (including members of an administrator group and `Administrator`) to be able to perform SVA subsystem administration functions (such as defining devices or forming arrays), the user must have write-access to a privileged ECAM device. Also, in order for any user (including members of an administrator group and `Administrator`) to be able to snap a partition, the user must have read-access to the source partition and write-access to the target partition.

What SVAA Administrators Can Do

SVAA administrators are the only users who can submit SVAA administration requests, such as defining a new ECAM path to the SVA subsystem and disconnecting users from the SVAA server.

See "Request Authorization Criteria," on page 26 for complete details.

What PPRC Administrators Can and Cannot Do

If a PPRC administrator group has been defined, PPRC modification requests, such as defining PPRC paths or PPRC pairs, can be submitted only by users meeting both of the following criteria:

- They must be a member of the PPRC administrator group, and
- They must have write access to one of the SVA subsystem's privileged ECAM devices.

If a member of the PPRC administrator group does not have write access to one of the SVA subsystem's privileged ECAM devices, any PPRC modification requests they submit will not be processed.

See “PPRC Administrators,” on page 25 for details.

Displaying Privileges

Any user can use the `querypermissions` command to display privileges for a user or a Windows group. Each user or group is identified as either an SVAA Administrator, a PPRC Administrator, or a User.

Note: When SVAA security is in “none” mode, all users are identified as SVAA Administrators since all users can perform all commands. See “Security Modes,” on page 29 for details.

SVAA Administrators

This section describes how to define, change, and drop the SVAA administrator group.

How SVAA Administrators Are Identified

To be an SVAA administrator, a user must be logged in as one of the following:

- Userid `Administrator`.
- A member of the SVAA administrator group, a special group that has been defined to have SVAA administrator privileges. The SVAA administrator group must be a valid Windows group, as defined in the User Manager. It can be a global group. Only one SVAA administrator group can be in effect at a time.

Initial SVAA Administrator Group

When the SVAA server is first installed, the default administrator groups are `Administrators` and `Domain Admins`. If you do not have a Windows group called `Administrators` or `Domain Admins`, you may want to assign administrator privileges to a different group.

To change to a different group later, you must first drop the existing SVAA administrator group, then assign a new Windows group. See “Changing the SVAA Administrator Group” for the steps. It is possible to run SVAA without an SVAA administrator group (see “Dropping the SVAA Administrator Group,” on page 25 for details).

Changing the SVAA Administrator Group

Use the following steps to change the SVAA administrator group.

1. Log in either as a member of the current SVAA administrator group or as `Administrator`.
2. Use the `dropadministratorgroup` command to remove the existing SVAA administrator group from consideration for administrative privileges. As a result, `Administrator` will be the only user that can perform SVAA administrator-level commands.
3. Log in as `Administrator` if you are not already.
4. Use the `setadministratorgroup` command to define a new SVAA administrator group. The new SVAA administrator group must already be a valid Windows group.

These changes take effect immediately; you do not need to restart the SVAA server.

Dropping the SVAA Administrator Group

In some situations you may want to restrict SVAA administrator privileges to the `Administrator` user only. To do this, a user with SVAA administrator privileges can use the `dropadministratorgroup` command to remove the current SVAA administrator group from consideration for administrator privileges. As a result, `Administrator` will be the only user that can perform SVAA administrator-level commands. (Of course, the dropped group still exists as a valid Windows group.)

Note: If you shut down the SVAA server after dropping the administrator group, when the server is restarted, the administrator groups will be reset automatically to `Administrators` and `Domain Admins`.

PPRC Administrators

This section describes how to define, change, and drop the PPRC administrator group.

How Security is Handled for PPRC Modification Requests

PPRC modification requests are requests that modify PPRC paths or PPRC devices. PPRC modification requests are classified as requests that use the PPRC-related parameters on the following commands: `alterdevice`, `alteriointerface`, or `definedevice`.

When a user submits a PPRC modification request, the request is processed in one of two ways, depending on whether a PPRC administrator group has been defined:

- If there is a PPRC administrator group, the user must both be a member of that group and also have write access to one of the SVA subsystem's privileged ECAM devices in order for the request to be processed. If the user is a member of the PPRC administrator group but does not have write access to a privileged ECAM device, the request will not be processed. Likewise, if the user has write access to a privileged ECAM device but is not a member of the PPRC administrator group, the request will not be processed.
- If no PPRC administrator group is defined, the user must simply have write access to one of the SVA subsystem's privileged ECAM devices in order for the request to be processed.

How PPRC Administrators Are Identified

To be a PPRC administrator, a user must be a member of the PPRC administrator group. The PPRC administrator group must be a valid Windows group, as defined in the User Manager. It can be a global group. Only one PPRC administrator group can be in effect at a time.

There is no initial PPRC administrator group, so until you define one explicitly with the `setadministratorgroup` command, the system will not check for PPRC administrator privileges.

Changing the PPRC Administrator Group

Use the following steps to change the PPRC administrator group.

1. Log in either as a member of the current SVAA administrator group or as `Administrator`.
2. Use the `dropadministratorgroup -pprcadmin` command to remove the existing PPRC administrator group. At this point, all PPRC modification requests will only require write access to a privileged ECAM device.
3. Use the `setadministratorgroup -pprcadmin` command to define a new PPRC administrator group. The new PPRC administrator group must

already be a valid Windows group. As a result, all PPRC modification requests will require membership in the PPRC administrator group as well as write access to a privileged ECAM device.

These changes take effect immediately; you do not need to restart the SVAA server.

Dropping the PPRC Administrator Group

The current PPRC administrator group can be removed at any time. To do this, a user with SVAA administrator privileges must use the `dropadministratorgroup` command. As a result, all PPRC modification requests will only require write access to a privileged ECAM device. (Of course, the dropped group still exists as a valid Windows group.)

Request Authorization Criteria

Each CLI and SVA Console request has a set of authorization criteria that a user must meet in order to be able to perform the command. The authorization criteria are determined based on the request's type and action.

Each request is assigned one of the following types, based on the types of resources it modifies or queries:

- Server
- Subsystem
- System resource

In addition, each request is assigned one of the following actions, based on how it accesses the system:

- Alter
- Copy
- Query

Table 2-5, "Request Authorization Criteria" shows, for the CLI, how the request type and action are used together to determine each request's authorization criteria. It also identifies which CLI commands use which criteria.

Note: See the SVA Console Online Help for specific authorization criteria for each SVA Console request.

Table 2-5 Request Authorization Criteria

Request Type	Action	Successful Authorization Criteria	Applicable Commands

Server	Alter	User must be an SVAA administrator.	`addsubsystempath`
			`dropadministratorgroup`
			`dropconnection`
			`refreshcache`
			`removesubsystempath`
			`setcacherefreshrate`
			`setsecuritymode`
			`startservertrace`
			`stopservertrace`

Subsystem	Alter	User must have write access[a] to a privileged ECAM device.	`alterdevice`
			`alteriointerface`
			`altersubsystem`
			`attndevice`
			`definedevice`
			`deletedevice`
			`formarray`
			`refreshcache`
			`setcacherefreshrate`
			`startdrain`

Subsystem	Copy	If a PPRC administrator group has been defined, user must be a member of that group. If no PPRC administrator group has been defined, user must have write access[a] to a privileged ECAM device.	PPRC modification parameters in the following commands:
			`alterdevice`
			`alteriointerface`
			`definedevice`

System Resource	Alter	User must have write access[a] to the resource named in the request.	`release`
System Resource	Copy	User must have read access to the source resource, and write access[a] to the target resource.	`snap`

Table 2-5 Request Authorization Criteria

Request Type	Action	Successful Authorization Criteria	Applicable Commands

Server	Query	All users allowed.	`displayserver`
			`queryadministratorgroup`
			`querycache`
			`queryconnections`
			`querypermissions`
			`queryrequests`
			`querysecuritymode`
			`queryserverlogentries`
			`queryservertrace`
			`querysubsystem`
			`querysubsystempath`
			`queryversion`

Subsystem	Query	All users allowed.	`displayarray`
			`displaydevice`
			`displaydrivemodule`
			`displayiointerface`
			`displayncaplodpct`
			`displaysubsystem`
			`reportevents`

a. Read and write access to a resource is established through standard Windows file permissions.

Special Cases

Some commands are exempt from SVAA security because they are "client-side" commands, not processed by the SVAA server. These commands can be performed by any user. The commands are:

- `exit`
- `get`
- `help`
- `queryclientlogentries`
- `queryclienttrace`
- `set`
- `startclienttrace`
- `stopclienttrace`

The `setadministratorgroup` command is another special case; it can only be performed by the `Administrator` userid.

Although included in the CLI command reference, `sibadmin` is not actually an SVAA command. Therefore it is not subject to the SVAA security system, and access to it is based solely on Windows file permissions.

Security Modes

SVAA security can be set to one of the following modes:

- Active—SVAA security is in full force. Authorization checks are performed on all commands. If a user is not authorized to perform a command, an error message is displayed and the command fails. This is the default mode.
- Warning—Authorization checks are performed on all commands. If a user is not authorized to perform a command, a warning message is printed to the SVAA server log, but the command continues to completion. Essentially allows all users to run all commands, but with warning messages logged to the SVAA server log; no error message is displayed at the CLI.
- None—Authorization checking is not performed. All users can initiate all CLI commands. It is not recommended that you use this mode.

Note: Even when the SVAA security system is running in "warn" or "none" mode, the underlying operating system security is still in effect.

Initial Mode Setting

The first time you start the SVAA server, it will come up in "active" mode which is the default. SVAA will continue to come up in this mode until you explicitly change it.

Changing the Security Mode

You can change the security mode at any time with the `setsecuritymode` command. This command requires SVAA administrator privileges. The new mode takes effect immediately; you do not need to shut down and restart the SVAA server.

The mode you specify is stored in the Windows Registry, so whenever you start the SVAA server, it comes up in this mode.

Displaying the Security Mode

Any user can use the `querysecuritymode` command to display the current security mode setting.

Scenarios

Following are some scenarios which illustrate the functioning of the SVAA security system.

"Active" Mode Scenarios

- A general user with write access to a privileged ECAM device submits a `definedevice` command; the command is processed.
- A general user without write access to a privileged ECAM device submits a `querysubsystempath` command; the command is processed.
- A general user submits an `addsubsystempath` command; the command fails, as it requires SVAA administrator privileges.
- A general user with read, but not write, access to both the source and target objects submits a `snap` request; the command fails, as it requires the user to have write access to the target object.

- A member of the SVAA administrator group without write access to the requested object submits a `release` command; the command fails, as write permission is required, even for SVAA administrators.
- A PPRC administrator group is defined. A user who is a member of the SVAA administrator group but not the PPRC administrator group submits an `alteriointerface -addpath` command; the command fails, as membership in the PPRC administrator group is required.
- A PPRC administrator group is defined. A user who is a member of the PPRC administrator group but does not have write access to one of the SVA subsystem's privileged ECAM devices submits an `alterdevice -addrcpairs` command; the command fails, as write access to a privileged ECAM device is required.
- No PPRC administrator group is defined. A user with write access to a privileged ECAM device submits an `alterdevice -modifyrcpairs` command; the command is processed.

"Warn" Mode Scenarios

In the following scenarios, the security mode is set to "warn."

- A member of the SVAA administrator group submits a `dropadministratorgroup` command; the command is processed.
- The same user submits a `setadministratorgroup` command; the command is processed with a warning logged to the SVAA server log (in active mode this command can only be performed by `Administrator`).
- A general user without write access to a privileged ECAM device, but with write access to the requested resource, submits a `release` command; the command is processed.
- A general user without write access to a privileged ECAM device submits an `alterdevice` command; the command is processed with a warning logged to the SVAA server log (in active mode this command requires the user to have write access to a privileged ECAM device).

"None" Mode Scenarios

- A general user submits a `removesubsystempath` command; the command is processed.
- A member of the SVAA administrator group submits a `setadministratorgroup` command; the command is processed.

Related CLI Commands

Table 2-6, “Security commands” lists commands you can use to manage user security. See “SVAA CLI Command Reference (Help Files),” on page 89 for detailed descriptions of the commands and their parameters.

Table 2-6 Security commands

Command	Function
`dropadministratorgroup`	Drop or remove the current SVAA administrator or PPRC administrator group.
`queryadministratorgroup`	Display the current SVAA and PPRC administrator groups.
`queryconnections`	Display permissions, including SVAA and PPRC administration, for all users connected to the SVAA server.
`querypermissions`	Display privileges, including SVAA and PPRC administration, for specified users or groups.
`querysecuritymode`	Display the current SVAA security mode.
`setadministratorgroup`	Define an SVAA and/or PPRC administrator group.
`setsecuritymode`	Set the SVAA security mode.

Multiple Servers Attached to an SVA Subsystem

If you have more than one server connected to an SVA subsystem, and you make a subsystem configuration change from one server, you should allow at least two minutes for the change to show up on the other server(s). For example, if you define a device on one server, allow at least two minutes for that device to show up on any other servers connected to the same SVA subsystem.

Chapter 3. Storage Subsystem Management

ECAM Devices

An ECAM (extended control and monitoring) device is essentially a communications path between the SVAA server and an SVA subsystem. In order for the SVAA server to send commands to an SVA subsystem, the subsystem must have at least one ECAM device.

A privileged ECAM device is an ECAM device that is eligible to send requests that alter the SVA subsystem configuration (such as `alterdevice` and `altersubsystem`). At least one ECAM device in an SVA subsystem must be privileged.

Note: The following restrictions apply to ECAM devices:

- Message traffic on an ECAM device is generally low. You can use ECAM devices for customer data, but for performance reasons it is not recommended that they be used for data that is accessed frequently.
- In cases where the host operating system reserves devices (for example, AIX and Windows), it is not recommended that ECAM devices be used for filesystems.
- Sharing ECAM devices among multiple hosts is not supported. Unpredictable results may occur.

Defining the First Privileged ECAM Device

The first privileged ECAM device for an SVA subsystem must be defined at the LOP (local operator panel), as part of the initial minimum subsystem configuration. (See "Minimum Configuration Steps You Must Perform at the LOP," on page 6 for additional information.)

Defining Additional Privileged ECAM Devices

After the first privileged ECAM device has been defined at the LOP, you can use the SVAA server to define additional ones.

Use the following steps to define additional privileged ECAM devices for an SVA subsystem:

1. Use the `alterdevice` or `definedevice` command to create a privileged ECAM functional device; that is, eligible to communicate requests that alter the SVA subsystem configuration.
2. Use the `addsubsystempath` command to identify the device as an ECAM path to the SVA subsystem.

3. Although not required, it is recommended that you use the `displaydevice` command to verify that the device has been successfully assigned privileged ECAM status.

Deleting an ECAM Device

Use the following steps to delete an ECAM device (privileged or unprivileged):

1. Use the `removesubsystempath` command to remove the path through the device to the SVA subsystem.
2. Use the `deletedevice` command to delete the device.

If you delete an ECAM device without first using the `removesubsystempath` command, the device path will remain in the Windows Registry as a valid path to the SVA subsystem. Consequently, the next time you restart the SVAA server, the server will unsuccessfully attempt to re-establish communications with the SVA subsystem through the device and error messages will be entered into the SVAA server log. To resolve this condition, you must re-run the SVAA Manager and redefine the valid ECAM devices. (See "SVAA Manager Program," on page 10 for details.)

Related CLI Commands

Table 3-1, "ECAM device commands" lists commands you can use to maintain ECAM devices. See "SVAA CLI Command Reference (Help Files)," on page 89 for detailed descriptions of the commands and their parameters.

Table 3-1 ECAM device commands

Command	Function
`addsubsystempath`	Enable a new path to an ECAM device.
`querysubsystempath`	Display ECAM device paths.
`removesubsystempath`	Delete ECAM device paths.
`alterdevice`	Modify the privileged ECAM status of a functional device.
`definedevice`	Define the privileged ECAM status of a functional device.

I/O Interfaces

An I/O interface is a channel interface between the SVA subsystem and the computer on which the SVAA server is running. An I/O interface can be a serial (ESCON) or fibre-channel interface.

I/O interfaces are defined at the SVA subsystem LOP (local operator panel). Using the SVAA server, you can change only the name of an I/O interface, as well as add or delete a PPRC (peer-to-peer remote copy) path between a primary and secondary SVA subsystem. To change any other characteristic, you must use the LOP.

Related CLI Commands

Table 3-1, “I/O interface commands” lists commands you can use to maintain I/O interfaces. See “SVAA CLI Command Reference (Help Files),” on page 89 for detailed descriptions of the commands and their parameters.

Table 3-1 I/O interface commands

Command	Function
`alteriointerface`	Change the name of an I/O interface.
`displayiointerface`	Display I/O interface characteristics.

SVA Subsystems

You must define at the LOP (local operator panel) the name of an SVA subsystem and a subsystem identifier (SSID) for at least one logical 3990 controller emulated within the subsystem.

Note: The SSID is used by S/390 hosts only.

Altering Subsystem Characteristics

After the SVA subsystem name, SSID, and privileged ECAM device have been defined at the LOP, you can use the SVAA server to change any of the following characteristics:

- SVA subsystem name
- SSIDs
- Site name
- Default array size for the subsystem (for V960 and V2X subsystems, the only valid default array size is 13+2)

Related CLI Commands

Table 3-2, “SVA subsystem commands” lists commands you can use to maintain I/O interfaces. See “SVAA CLI Command Reference (Help Files),” on page 89 for detailed descriptions of the commands and their parameters.

Table 3-2 SVA subsystem commands

Command	Function
`altersubsystem`	Modify SVA subsystem characteristics.
`displaysubsystem`	Produce a report showing subsystem characteristics
`querysubsystem`	Display a list of current subsystems.

Drive Modules

Drive modules are the individual physical devices in the SVA subsystem.

Related CLI Commands

Table 3-3, “Drive module commands” lists commands you can use to maintain drive modules. See “SVAA CLI Command Reference (Help Files),” on page 89 for detailed descriptions of the commands and their parameters.

Table 3-3 Drive module commands

Command	Function
`displaydrivemodule`	Display information about specified drive modules.
`startdrain`	Drain specified drive modules

Arrays

An array is a group of drive modules that is used collectively to achieve data redundancy and/or improved performance. There can be up to eight arrays in an SVA subsystem.

Note: For V960 and V2X subsystems, the only valid default array size is 13+2.

For V960 and V2X subsystems, all arrays consist of 15 drive modules; for earlier SVA subsystem models, arrays consist of either 7 or 15 drive modules. An array with 15 drive modules is described as a “13+2” configuration, while a 7-drive array has a “5+2” configuration. In either case, the first value (“13” or “5”) refers to the portion (in terms of a number of drive modules) of the array's capacity that is available for user data. The second value (“2”) refers to the portion that is used for redundancy data. Thus a 13+2 array provides storage capacity equivalent to:

- 13 drive modules of user data, and
- 2 drive modules of redundancy data.

The size of any array you form (13+2 or 5+2) is determined by the current array size value for the SVA subsystem, which is set with the `altersubsystem` command. For V960 and V2X subsystems, this value must be 13+2.

Forming the First Array

The first array (a production array) for an SVA subsystem is formed at the LOP (Local Operator Panel) as part of the initial minimum subsystem configuration. (See “Minimum Configuration Steps You Must Perform at the LOP,” on page 6 for additional information.)

To form the first array in an SVA subsystem, 16 spare drive modules are required for a 13+2 array, and 8 are required for a 5+2 array (not valid for V960 nor V2X subsystems). In either case, the extra drive is reserved as a spare for the entire subsystem.

Forming Additional Arrays

After the first array has been defined at the LOP, you can use the SVAA server to form additional arrays. On SVA subsystem models earlier than V960, the number of spare drive modules required to form an additional array depends upon the current SVA subsystem array size or, if you override it, the array size that you specify when you form the array. To form *n* arrays (after the first array), the number of available spare drive modules required is:

(n * 7) + 1 to form “5+2” arrays, or

(n * 15) + 1 to form “13+2” arrays.

Use the following steps to form additional arrays in the Production partition:

Note: For SVA subsystems earlier than V960, before beginning this procedure you must decide upon the size of the arrays that you want to form (5+2 or 13+2). Here are some general rules to help you decide on array size:

- The larger array size (13+2) provides greater physical capacity (because there are fewer total spares) and better performance (because more drives are written to simultaneously).
- The smaller array size (5+2) provides more total spares and faster recovery times.

1. Set the SVA subsystem array size value (13+2 for V960 and V2X subsystems), if it hasn’t been set already, using the `altersubsystem` command.
2. Determine the total number of spare drives necessary to form the arrays: multiply the array size (7 or 15) by the number of arrays to be formed, and add 1 (the minimum number of available spares required).
3. Use the `displaydrivemodule` command to ensure that there are enough drive modules in the Spares partition to form the arrays.
4. Use the `formarray` command to create the additional Production arrays.
5. Use the `displayncaplodpct` command to verify the increased disk array capacity. The “Disk Array Capacity” field should show an increase, while the “NCL %” should show a decrease.

Related CLI Commands

Table 3-4, “Array commands” lists commands you can use to maintain arrays. See “SVAA CLI Command Reference (Help Files),” on page 89 for detailed descriptions of the commands and their parameters.

Table 3-4 Array commands

Command	Function
`formarray`	Create a new array.
`altersubsystem`	Modify the current array size for an SVA subsystem.
`displayarray`	Display array information.
`displaydrivemodule`	Display drive modules in an array. Also displays spares, so you can verify you have enough of them on-hand before creating arrays.
`startdrain`	Drain an array.

Functional Devices

Functional devices are the virtual SCSI or CKD (3990) devices you use to store data on the SVA subsystem. Functional devices can be used individually for storing data, or several of them together can be used to make up a SCSI larger LUN device (see "SCSI Larger LUN Devices," on page 39).

SCSI Devices

When defining a SCSI device, you must specify its domain, target, and LUN. A domain can be thought of as a logical SCSI bus. Each controller in the Windows computer is connected to a domain, giving it access to the devices that have been configured within that domain.

The SCSI target must be defined as "0" for all devices on a fibre-channel interface.

There can be up to 16 SCSI domains per SVA subsystem. Each domain can have up to 256 devices (although on V960 and earlier subsystems, not all domains can have the maximum 256 devices because the total number of devices per SVA subsystem is 1024).

You can choose between the following device types for SCSI functional devices:

- SCSI-A—3390-3 devices with SCSI format.
- SCSI-B—3390-9 devices with SCSI format.
- SCSI—SCSI larger LUN devices.

CKD Devices

You can use the SVAA for Windows server to manage devices that are used by an S/390 host for data storage. In order for an S/390 host to be able to access a device, the device must be defined as enabled for SCSI read access, as well as enabled for CKD read and write access.

Related CLI Commands

Table 3-5, "Functional device commands" lists commands you can use to maintain functional devices. See "SVAA CLI Command Reference (Help Files)," on page 89 for detailed descriptions of the commands and their parameters.

Table 3-5 Functional device commands

Command	Function
`alterdevice`	Modify functional device characteristics. **Note:** You can modify only some characteristics of a SCSI larger LUN. **Note:** You cannot alter a device's FDID or device type. Instead, you must delete the device and then redefine it with the new characteristics. If you want to retain the data on the device, you must first back it up, then after redefining the device, you can restore the data.
`attndevice`	Initiate a state-change interrupt on a CKD device.

Table 3-5 Functional device commands (Continued)

Command	Function
`definedevice`	Add new functional devices or SCSI larger LUN devices to an SVA subsystem.
`deletedevice`	Delete functional devices or SCSI larger LUN devices from an SVA subsystem. **Note:** When you delete a device, all data on the device is effectively erased. Therefore, prior to deleting the device, you must either back up any data that must be preserved or relocate it to another functional device.
`displaydevice`	Display functional device characteristics.

Configuring New Functional Devices on the Windows Computer

New functional devices on the SVA subsystem must be configured on the Windows computer. Use the following steps.

Note: To have drive letters assigned permanently, you should log on as userid `Administrator` before performing these steps. If you log on as any other userid, the drive letters you assign may be later reassigned to other disks by the `Administrator` user.

1. Access the disk management tool for your operating system.
2. Create a primary partition. Be sure to assign a partition size, a drive letter, and a format. You can give the drive a volume label if you want.

   Note: SVAA does not support extended partitions.

Note: If you are using dynamic disks, you should be aware that you cannot snap a dynamic disk and you cannot use a dynamic disk as an ECAM device.

SCSI Larger LUN Devices

SCSI larger LUN (LLUN) devices are logical devices made up of one or more physical devices. A SCSI larger LUN device is composed of one parent device (the index 0 device) and one or more child devices (indexes 1–n). Although multiple devices are involved, they are viewed by the host system as a single device for storing data.

SCSI larger LUN devices allow you to create devices of virtually any size. While SCSI functional devices come in only two sizes—2.5GB for SCSI-A and 7.4GB for SCSI-B—larger LUN devices can be considerably larger; the actual maximum size depends on the device type and the SCSI block size assigned.

Larger LUN Device Rules

- All functional devices making up a larger LUN device have the same device type (that is, all SCSI-A or all SCSI-B; no mixed types).
- A functional device can be included in only one larger LUN device.

- The total number of functional devices making up a larger LUN device cannot exceed 1024.
- The maximum size of a larger LUN device depends on its device type and SCSI block size.
- Due to SCSI limitations, the size of a larger LUN device cannot exceed 2TB; anything larger would not be accessible.
- When you create a SCSI larger LUN device, you must specify its capacity (in megabytes or gigabytes). The system actually rounds up the capacity to the next multiple, depending on the device type and block size you assign.
- The attributes of the parent device—such as device type and host computer write access—are propagated to all child devices in the larger LUN.
- Once a SCSI larger LUN device has been defined, you can make only the following modifications:
  - Change the device name
  - Enable/disable the device as a CKD device
  - Enable/disable CKD device read/write capability
  - Change the privileged ECAM status for the device
  - Increase the capacity of the device
  - Create, modify, or delete PPRC pairs

  If you want to make any other modifications, you must: 1) define a new larger LUN device with the desired characteristics; 2) move any data from the old larger LUN device to the new one; 3) delete the old device.

Creating Larger LUN Devices

The child devices within a larger LUN device can be defined either explicitly or implicitly, as explained below:

- To explicitly define child devices, you specify a pool of FDIDs to use. Any available devices within the pool are used to create the larger LUN device. ("Available" devices are those not already defined as a functional device or used in another larger LUN device.) The total capacity of the pool must be greater than or equal to the requested capacity of the larger LUN device. Any extra capacity in the pool is ignored.

  The larger LUN FDIDs must not be included in the FDID pool. For example, if you issued a request to create larger LUN devices C0 and C1 by using FDID pool C1 to D1, the system would first create C0 by using child FDIDs C1, C2, and C3 (assuming the larger LUN requires three child FDIDs); then the system would attempt to create larger LUN C1, but it would be unable to do so because the FDID had already been used.
- To implicitly define child devices, you omit the FDID pool and allow the system to select FDIDs. The system attempts to cluster FDIDs within a larger LUN device, so it selects the first available child devices with FDIDs greater than the parent FDID. For example, to create larger LUN 7, the system uses child devices starting with FDID 8. If 8 is not available it uses 9, and so on.

You cannot create multiple larger LUN devices by specifying a sequential list of FDIDs. For example, if you issued a request to create larger LUN devices 21 and 22, the system would first create larger LUN 21 by using child FDIDs 22, 23, and 24 (assuming the larger LUN requires three child FDIDs); then the system would attempt to create larger LUN 22, but it would be unable to do so because the FDID had already been used.

Table 3-6 SCSI larger LUN device commands

Command	Function
`alterdevice`	Modify some SCSI larger LUN device characteristics. Change a functional device into a SCSI larger LUN device by adding child FDIDs. **Note:** You cannot alter any device's FDID or device type. Instead, you must delete the device and then redefine it with the new characteristics. If you want to retain the data on the device, you must first back it up, then after redefining the device, you can restore the data.
`definedevice`	Define new SCSI larger LUN devices for an SVA subsystem.
`deletedevice`	Delete SCSI larger LUN devices from an SVA subsystem. **Note:** When you delete a device, all data on the device is effectively erased. Therefore, prior to deleting the device, you must either back up any data that must be preserved or relocate it to another functional device.
`displaydevice`	Display SCSI larger LUN device characteristics including its composite child devices.

Drains

A drain is the process of moving data off one or more drive modules to other drive modules in order to free the space on the source drives. When the designated drive modules have been successfully drained, they are placed in the MAT partition for reallocation.

You can select drive modules to be drained in one of three ways:

- By array
- By tray
- By individual drive modules (designated by unit, tray, and slot)

SVA subsystem capacity must be sufficient to receive data to be drained. If available capacity is inadequate, the drain request is rejected.

Draining an Array

In an array drain, all the data in the array is moved to other arrays in the SVA subsystem partition. If FDIDs are defined for the partition there must be another array in the same partition for the drain to be performed. If no FDIDs are defined for the partition, you can drain the array.

Draining a Tray

The method used to drain a tray depends on the number of spares available in addition to the one-spare minimum requirement.

- If, for each drive module in the tray to be drained, a spare drive is available outside that tray, each drive module is drained into one of the spares.
- If the number of available spares is inadequate, the process involves draining arrays. In this case, drain activity may be observed in "associated" drive modules—modules that are part of an array being drained, but that lie outside the "target" tray.

Draining a Drive Module

To be selected for draining, a drive module must be in the Production or Spares partition and in the active state; that is, its status must be PA or SA. (Explanations of all drive module status codes appear in Appendix A., "Drive Module Status Codes".)

In a single drive module drain, all the data on the drive is moved to a spare drive module. Thus, for each Production partition drive module to be drained, a spare drive module must be available.

No spare drive is required to drain a spare. Draining a spare is instantaneous.

Reporting Drain Events

You can use the `reportevents` command to generate a detailed report, by SVA subsystem, of drain events that have occurred within the last 60 days. The report shows detailed information for an event, such as the time and date of the request, percentage of the drain complete, and a list of the individual drive modules involved.

Related CLI Commands

Table 3-7, "Drain commands" lists commands you can use to manage drains. See "SVAA CLI Command Reference (Help Files)," on page 89 for detailed descriptions of the commands and their parameters.

Table 3-7 Drain commands

Command	Function
`startdrain`	Initiate a drain.
`reportevents`	Display information about drains.

Space Management

Net Capacity Load

Net capacity load (NCL) is the amount of back-end physical capacity used in an SVA subsystem. This includes user data and the system areas needed to maintain

the arrays. NCL does not include data in cache until the data is written to the back-end.

Space Release

The purpose of the partition release facility is to inform the SVA subsystem that a given SCSI partition no longer contains any useful data and the subsystem may release any space the partition is using. Proper use of this facility enables the SVA subsystem to manage its storage more efficiently, as it increases the available physical capacity of the SVA subsystem and decreases the NCL.

Restrictions

There are significant data loss considerations you should be aware of when releasing a partition. You must observe the following restrictions:

- In this release of the SVAA server, only entire SCSI partitions can be released. There is no support for releasing a file within a file system. The SVAA server cannot distinguish the contents of a SCSI partition; it simply releases the partition as a whole.
- Partition release can be performed on an SVA subsystem only.
- To release a partition, you must have read/write access to it.
- Before attempting to release a partition, you should make sure it is not being accessed by any application and that it has no open files. If there are open files on the partition, SVAA will attempt to close them; if it is unable to close them, the request will fail.
- After releasing a partition, you must re-format it before you can use it.

Related CLI Commands

Table 3-8, “Space management commands” lists commands you can use to manage SVA subsystem space. See “SVAA CLI Command Reference (Help Files),” on page 89 for detailed descriptions of the commands and their parameters.

Table 3-8 Space management commands

Command	Function
`displaydevice`	Display functional capacity information for specified devices or entire SVA subsystems.
`displayncaplodpct`	Display NCL percentages for specified SVA subsystems.
`release`	Release space on a SCSI device.

Chapter 4. SnapShot

How SnapShot Works

Traditionally, the task of duplicating data in the computer room has consisted of copying the data, bit by bit from one place to another, as in copying data from one location on a disk storage device to another, or to a different disk storage device.

StorageTek's duplication method, SnapShot, duplicates SCSI devices without physically moving any data. SnapShot, instead, creates a second set of pointers to the same data, thus creating a second SCSI device. This dramatically reduces the time required before the data (both original and duplicated) is available to an application.

SnapShot accomplishes this duplication quickly because:

- No data is moved
- Only new pointers to the original SCSI device are created
- No additional physical disk or cache space is used in making the duplicate.

Furthermore, SnapShot operations are performed entirely within the SVA subsystem; they do not consume resources in the host operating system, bus, or host bus attachment (HBA).

Snaps of a SCSI device require a minimum of host resources. The actual snapping is performed within the SVA subsystem. The only physical back-end storage allocated by the subsystem is for:

- The original SCSI device (shared between the source and target)
- Any changes made to the source device
- Any changes made to the target device

The two independent sets of pointers treat the data as if it were two physically separate images. Updates can occur simultaneously both to the original data and to the duplicate, with no concern for compromising the data in either device. You can use a snapped device as the source of a transfer to tape while applications continue to use the original device.

Terminology

StorageTek uses the term “snap” for the duplication function provided by SnapShot. We also use this term for the result of a snap operation.

The object being duplicated is called the “source” and the object of the duplication is called the “target.” That is, with SnapShot, we snap the source onto the target.

These terms suggest that this new form of data duplication is quick, economical of resources, and easy to use.

Supported Objects

SnapShot for Windows can duplicate any of the following objects (which are referred to as "volumes" or "devices" in this chapter):

- Simple, contiguous SCSI partitions contained within a single drive
- Entire SCSI physical drives
- Striped and spanned volumes created on Windows NT 4
- All types of Windows dynamic disk volumes (simple, spanned, mirrored, striped, and RAID-5)

SnapShot currently does not support Veritas Logical Volume Manager (LVM) on Windows.

Note: SnapShot on Windows 2003 is supported by SVAA Patch 20 and higher.

Technical Considerations

General Restrictions

There are significant data management considerations when running the `snap` command. The following restrictions must be observed when performing a SnapShot operation.

- A snap to a target object containing data will irrevocably overlay that data.
- Snaps can only be performed when the source and target devices reside within the same SVA subsystem.
- The target device cannot be the same as the source device. That is, you cannot snap a device onto itself.
- In order to snap a device, the user must have read-access to the source device, and read/write-access to the target device.

  Note: Normally `Administrator` has implied access to all source or target devices. However, if another user has set permissions that exclude `Administrator`, the snap will fail.
- The source and target devices must be of equal size. See "Disk Layout Restrictions," on page 47 for details.
- The source and target devices must actually exist. That is, they cannot have a size of 0.
- SVA LUNs used for the source and target devices should have the same SCSI logical block size. SCSI logical block size is specified with the `-scsiblks` parameter on the `definedevice` command.

Object Type Restrictions

Following are restrictions on the types of objects that can be snapped:

- In this release of SnapShot for Windows, you can snap only entire SCSI devices. SnapShot cannot distinguish the contents of a device; it simply copies the device as a whole.
- There is no support for snapping a file within a file system. Veritas LVM on Windows is not supported.

Caution: Attempting to use SnapShot on unsupported volumes may have unpredictable results.

- The source and target devices must be of the same type. That is, you can only snap a drive letter to a drive letter, or a physical drive to a physical drive. It is not possible to snap a drive letter to a physical drive, or vice versa.

Disk Layout Restrictions

To ensure the success of a SnapShot operation, before a snap is initiated the target device must have exactly the same layout as the source. This is due to the specifics of Windows internal disk representation. Specifics for each device type are given below.

Basic Disk Partitions

Basic disk partitions must meet all the following requirements:

- Source and target devices must be on different physical drives (SVA LUNs).
- Source and target devices must be the same size.
- Source and target devices must be positioned at the same relative offset from the beginning of their corresponding physical drives.

Following is an example:

- The snap source is drive D:, a 500MB partition, at offset 200MB from the beginning of physical drive 2.
- The snap target should be a different drive letter (such as E:), also a 500MB partition, at offset 200MB from the beginning of a different physical drive (such as 3).

Striped and Spanned Volumes

Dynamic spanned volumes composed of two or more extents must meet all the requirements listed in “Basic Disk Partitions” above, plus the following requirement:

- The extents making up the source must be exactly the same size. And since the snap target must always have the same layout as the source, the extents making up the target must also be exactly the same size.

Following is an example:

- The snap source is drive F:, a 700MB volume consisting of the following two extents:
  - A 350MB extent on physical drive 5, at offset 300MB from the beginning of the drive.

  - A 350MB extent on physical drive 6, at offset 0MB from the beginning of the drive (that is, at the very beginning of the drive).

- The snap target should be a different drive letter (such as G:), also a 700MB volume consisting of the following two extents:
  - A 350MB extent on a physical drive other than 5 (such as 7), at offset 300MB from the beginning of the drive.
  - A 350MB extent on a physical drive other than 6 (such as 8), at offset 0MB from the beginning of the drive.

Entire Physical Drives:

Entire physical drives must meet all the following requirements:

- Source and target drives must contain the same number of partitions.
- Target partitions must be at the same positions as those on the source.
- Target partitions must be the same size as the corresponding partitions on the source.
- Source and target partitions must have drive letters assigned.

After the snap, the target partitions will contain the same information as the corresponding source partitions (that is, the partitions on the source physical drive at the same positions and of the same size as on the target).

Ensuring Data Integrity

It is strongly recommended that you close all files and quiesce all databases on the source device before attempting a snap. If a user were to update source data while a snap was in progress, SnapShot would complete the copy but the data could be corrupted.

After performing a snap, always run `chkdsk` on the target device to ensure the data integrity of the copy.

Special Considerations

This section describes important issues to remember when snapping Windows devices.

Volume Labels

When snapping basic disk partitions or striped and spanned volumes, the source volume label is not copied to the target; instead, the target volume label is preserved.

When snapping entire physical drives, however, the source volume label is copied to the target.

Partition Sizes

In order for SnapShot to work on Windows systems, both the source and target partitions must be exactly the same size (to within a byte). To view partition sizes, open Windows Explorer, right-click on a drive letter, and select **Properties**. On the **General** tab, the Capacity figure displayed in bytes must be exactly the same for both the source and target partitions.

There are several reasons why partitions created by Windows on identical devices may have different sizes; some, though not all, of these are:

- Partitions created on different versions of Windows may have different sizes due to the way Windows manages disk. So a 100MB partition created on Windows NT 4 may not be exactly the same size as a 100MB partition created on Windows 2000.
- Primary partitions and logical drives in extended partitions may not be of exactly the same size. That is, a 100MB primary partition may not be exactly the same size as a 100MB logical drive partition created in an extended partition.

To avoid mismatched partition sizes, create the source and target partitions in exactly the same way. For example, if the source is a logical drive in an extended partition, create the target as a logical drive in an extended partition. If you create the source partition on Windows 2000, create the target partition on Windows 2000 as well.

Caution: Third-party tools, such as PowerQuest® PartitionMagic® and others, may be able to correct some partition size mismatch problems. The use of disk editing tools is very dangerous, however, as serious data loss may result from incorrect use of such tools. Third-party tools are not endorsed nor supported by StorageTek; therefore, use them at your own risk.

Physical Drive Sizes

The source and target physical drives must be exactly the same size to ensure the success of the snap.

Net Capacity Load Considerations

When you are using SnapShot with an SVA subsystem, you should consider the effect of the snapped partition on the net capacity load (NCL) of the subsystem. When a source is first snapped, the snap consumes no back-end storage, because the tracks for the target share the same tracks as the source. No extra space is needed to accommodate the new partition at this point.

As updates are made, however, new tracks are written to the disk arrays. The more updates that occur to the source and target objects, the more tracks are written on behalf of each. These new tracks occupy more back-end storage as the contents of the source and target objects diverge. It is possible for the source and/or target to be completely rewritten. The potential effect on NCL must be considered in this case.

If NCL becomes a concern, you should consider the length of time target objects are kept on the SVA subsystem. The length of time you maintain the target objects, and the rate at which they become unique because of updates should determine when the target objects can be migrated to tape and deleted to make room for other snaps or to lower overall SVA subsystem NCL.

Chapter 5. Peer-to-Peer Remote Copy (PPRC)

Overview

Note: Only certain SVAA platforms and SVA subsystem configurations support the use of PPRC commands and functions. Contact StorageTek Software Support for details.

Peer-to-peer remote copy (PPRC) is a hardware solution supported by SVAA that enables any data (system or application) to be mirrored between pairs of disks on different SVA subsystems.

Once a PPRC pair is established, all data from the device to be mirrored (the primary volume) is copied to the secondary volume; this process is called "synchronization." From that point on, all updates to the primary volume are copied synchronously to the secondary volume, so that at any point in time the secondary volume is an exact mirror of the primary volume.

Note: This chapter explains how to set up PPRC from SVAA. See the *Peer to Peer Remote Copy Configuration Guide* for details on installing and configuring the SVA subsystem hardware for PPRC.

PPRC Terminology

Following are some key terms relating to PPRC:

- PPRC primary volume—A device that is being mirrored under PPRC control. A primary volume can be copied to only one secondary volume. A device can be only a PPRC primary or secondary volume, not both at the same time.
- PPRC secondary volume—A device that is receiving mirrored data from a primary volume. It is SCSI read/write disabled, physically protecting it from non-PPRC updates, and must be offline to all connected hosts. The secondary volume must have the same disk geometry as the primary volume.
- PPRC pair—A primary volume and its associated secondary volume. PPRC pairs are established and de-established using SVAA commands.
- PPRC link—The physical ESCON connection between two SVA subsystems. For PPRC to function properly, there must be ESCON cables between the primary and secondary SVA subsystems; these cables must be dedicated to each SVA subsystem for the sole purpose of processing PPRC copies. The ESCON cables can carry updates one-way only; therefore, each ESCON cable is dedicated to sending data from one specific SVA subsystem to another. To use bidirectional PPRC, you must have at least two links (one

for each direction). See the *Peer to Peer Remote Copy Configuration Guide* for details on unidirectional and bidirectional PPRC.

- PPRC path—The logical connection between two virtual control units (VCUs) used by PPRC. You can define multiple paths across a single PPRC link. In order for PPRC to access all 1024 devices on a V960 subsystem, or 4096 devices on a V2X, you must define at least one PPRC link and four (in the case of V960) or 16 (in the case of V2X) PPRC paths. (There can be up to four VCUs per SVA subsystem for V960, 16 for V2X—one for every 256 devices.)
- CKD enabled and CKD read/write enabled—Any volume that is part of a PPRC pair or PPRC bridge pair must be enabled for CKD access and CKD write access. You do this by setting the `-ckdena` and `-ckdrw` options on the `definedevice` or `alterdevice` command to `yes`.

Note: The following terms are summarized here. See "Bridge Volumes," on page 53 for complete descriptions.

- PPRC data bridge volume—Used by Power PPRC as a staging area for PPRC data.
- PPRC status bridge volume—Used by Power PPRC for acknowledgments that PPRC data has been successfully transmitted and received.
- PPRC bridge pair—A primary bridge volume and its associated secondary bridge volume. There are both PPRC data bridge pairs and PPRC status bridge pairs.

Power PPRC

The V960 and V2X support Power PPRC only; they do not support standard PPRC. Therefore, throughout this book the term "PPRC" implies Power PPRC. Following is a summary of Power PPRC characteristics.

Power PPRC provides better performance than standard PPRC through the use of a synchronization queue. The queue is set up through data bridge volumes on both the primary and secondary subsystems.

The V960 and V2X support two types of Power PPRC:

- Power PPRC direct—Uses an ESCON connection between the primary and secondary SVA subsystems. Requires the establishment of one data bridge volume on the primary subsystem and one on the secondary subsystem, as well as a data bridge path between the two volumes. You can establish up to four data bridge paths between the two subsystems.
- Power PPRC WAN—Enables PPRC over very long distances through the use of WAN facilities. As with Power PPRC direct, requires the establishment of one data bridge volume on the primary subsystem and one on the secondary subsystem, as well as a data bridge pair between the two volumes. Also requires the establishment of one status bridge volume on the primary subsystem and one on the secondary subsystem, as well as a status bridge pair between those two volumes.

Bridge Volumes

Following is a comparison of data bridge and status bridge volumes:

- Data bridge volume—Data bridge volumes serve as staging areas for the continuous transmission of data between the primary and secondary subsystems.

  Together, the data bridge volumes on the two subsystems are referred to as a "data bridge pair."

  – The primary SVA subsystem uses its data bridge volume to stage data for transfer.

  – The secondary SVA subsystem uses its data bridge volume to store arriving data before writing to its PPRC devices.

- Status bridge volume—Status bridge volumes on the primary and secondary subsystems are used to transmit and receive acknowledgments that data was successfully received.

Neither data bridge nor status bridge volumes are intended for data storage; in fact, you can only designate a previously unused functional device as a bridge volume. Bridge volumes are not formatted nor initialized. See "Establishing PPRC Data Bridge Pairs," on page 56 for details.

PPRC Secondary Volumes and SVA Path

Once a PPRC pair is established, the PPRC secondary volume is SCSI read/write disabled. As a result, SVA Path will no longer be able to access the secondary volume, causing it to log errors to the system log. To avoid this problem, before creating PPRC pairs you should instruct SVA Path to ignore any LUNs that will become PPRC secondary volumes.

Conversely, if you delete a PPRC pair (see "Deleting a PPRC Pair," on page 62) or recover a secondary volume (see "Recovering a Secondary Volume," on page 60), you must instruct SVA Path to once again recognize the affected LUN.

See the *SVA Path for Windows User's Guide* for specific instructions.

Establishing Power PPRC Direct

To use Power PPRC Direct, you must establish the PPRC paths and bridge pairs, then define PPRC pairs. Use the following steps:

1. Connect the proper cabling between the primary and secondary SVA subsystems. See the *Peer to Peer Remote Copy Configuration Guide* for details.

2. Define a PPRC path between the primary and secondary SVA subsystems. See "Defining PPRC Paths" for details.

3. Define PPRC data bridge volumes on the primary and secondary SVA subsystems. See "Defining PPRC Data Bridge Volumes," on page 55 for details.

4. Define bridge pairs on the primary and secondary SVA subsystems. See “Establishing PPRC Data Bridge Pairs,” on page 56 for details.

5. Define PPRC pairs for the data volumes. See “Establishing PPRC Pairs,” on page 56 for details.

Defining PPRC Paths

A PPRC path is a logical path between a primary and secondary SVA subsystem SSID (subsystem ID). Each SSID is a manually assigned value corresponding to a VCU (virtual control unit) within the SVA subsystem. Each VCU or SSID can address a range of 256 devices; therefore proper paths must be established between SSIDs on the primary and secondary subsystems before PPRC pairs can be established within those SSIDs. Following are devices addressed by each VCU:

Note: VCUs 4 through F are available on V2X subsystems only.

VCU 0: 000–FFF
VCU 1: 100–1FF
VCU 2: 200–2FF
VCU 3: 300–3FF
VCU 4: 400–4FF
VCU 5: 500–5FF
VCU 6: 600–6FF
VCU 7: 700–7FF
VCU 8: 800–8FF
VCU 9: 900–9FF
VCU A: A00–AFF
VCU B: B00–BFF
VCU C: C00–CFF
VCU D: D00–DFF
VCU E: E00–EFF
VCU F: F00–FFF

Use the `-addpath` option on the `alteriointerface` command to create a PPRC path. You will need to define the following:

- Primary SVA subsystem. To identify the primary SVA subsystem, you can specify either the VCU or the SSID.
- I/O interface on the primary subsystem that can access the secondary subsystem
- Secondary SVA subsystem name or serial number. When you identify the secondary SVA subsystem for the path, the following considerations apply:

– If the secondary SVA subsystem is “known” to the SVAA server, (that is the SVAA server has access to it) you can use the following to identify the secondary subsystem:

  • Either the name or serial number, and

  • Either the VCU (virtual control unit) or SSID (subsystem ID), or the VCU/SSID combination

  Note: See “Displaying Known Subsystems,” on page 61 for instructions on determining whether an SVA subsystem is known to the SVAA server.

– If the secondary SVA subsystem is “unknown” to the SVAA server, (that is, the SVAA server does not have access to it), you must use the following to identify the secondary subsystem:

  • Serial number, and

  • VCU/SSID combination

• Destination link address (applies only when the PPRC path goes through an ESCON director)

Defining PPRC Data Bridge Volumes

Use the `definedevice` command to define one data bridge volume each on the primary and secondary SVA subsystems. Following are considerations:

• The device must be created with `-devtype databridge`. By definition, a data bridge volume has a SCSI-B type, meaning it emulates a 3390-9 CKD device.

• A data bridge volume must be a previously unused device. If you want to use a specific FDID as a data bridge volume but it is already being used for data storage, you must move the data from the functional device, then delete and redefine the desired FDID as a data bridge volume. Bridge volumes are not formatted nor initialized.

• A data bridge volume cannot be a SCSI larger LUN device.

• You must set the `-ckdena` and `-ckdrw` options to `yes` in order to enable CKD access and CKD read/write access.

Establishing PPRC Data Bridge Pairs

Use the `-addrcpairs` option on the `definedevice` or `alterdevice` command to establish a bridge pair between the data bridge volumes you have just defined. Following are considerations.

- By default, the FDID of the secondary volume is the lowest byte of the primary volume's FDID (for example, if the FDID of the primary volume is 1FF, the FDID of the secondary volume will be FF). Optionally, you can specify a different FDID for the secondary volume.
- When you identify the SVA subsystem for the secondary volume, the following considerations apply:
  - If the secondary SVA subsystem is "known" to the SVAA server, (that is the SVAA server has access to it) you can use the following to identify the secondary subsystem:
    - Either the subsystem name or serial number, and
    - Either the VCU (virtual control unit) or SSID (subsystem ID)

    Note: See "Displaying Known Subsystems," on page 61 for instructions on determining whether an SVA subsystem is known to the SVAA server.
  - If the secondary SVA subsystem is "unknown" to the SVAA server, (that is, the SVAA server does not have access to it), you must use the following to identify the secondary subsystem:
    - Serial number, and
    - SSID

Establishing PPRC Pairs

Use the `-addrcpairs` option on the `definedevice` or `alterdevice` command to establish a PPRC pair. See "Establishing PPRC Data Bridge Pairs" for some considerations. Following are additional ones:

- You can use the command to establish multiple PPRC pairs at once.
- The primary and secondary volumes must have the same disk geometry; that is, they must have the same number of tracks on each cylinder and the same number of bytes on each track.
- A PPRC primary or secondary volume cannot be a privileged ECAM device.
- All PPRC secondary volumes must be offline to all host systems.
- For both primary and secondary volumes, you must set the `-ckdena` and `-ckdrw` options set to `yes`.
- By default, PPRC pairs are established in full-copy mode which means all tracks on the PPRC primary volume are copied to the secondary volume as soon as the pair is established. You can optionally create pairs in "nocopy" mode, which means that no data is copied from the primary to the secondary

volume. See “Synchronization of Data Within a PPRC Pair,” on page 60 for more information.

- Once a PPRC pair is established, the secondary volume will not be accessible to SVA Path. See “PPRC Secondary Volumes and SVA Path,” on page 53 for details.

Establishing Power PPRC WAN

Establishing Power PPRC WAN is similar to establishing Power PPRC Direct, except it also includes the added steps of defining bridge paths, volumes, and pairs for the status information. Use the following steps:

1. Connect the proper cabling between the primary and secondary SVA subsystems. See the *Peer to Peer Remote Copy Configuration Guide* for details.

2. Define one **data** bridge volume on the primary SVA subsystem and one on the secondary subsystem.

   Use the `definedevice` command. See “Defining PPRC Data Bridge Volumes,” on page 55 for considerations.

   Note: Steps 2 and 5 can be performed in one step by using the `-addrcpairs` option on the `definedevice` command.

3. Define one **status** bridge volume on the primary SVA subsystem and one on the secondary subsystem.

   Use the `definedevice` command. Following are considerations:

   – The device must be created with `-devtype statusbridge`. By definition, a status bridge volume has a SCSI-A type, meaning it emulates a 3390-3 CKD device.

   – A status bridge volume must be a previously unused device. If you want to use a specific FDID as a status bridge volume but it is already being used for data storage, you must move the data from the functional device, then delete and redefine the desired FDID as a status bridge volume.

   – A status bridge volume cannot be a SCSI larger LUN device.

4. Define two PPRC paths from the **primary** to the **secondary** SVA subsystem. One path will be used for the **data** bridge, the other will be used for the **status** bridge.

   Use the `-addpath` option on the `alteriointerface` command. See “Defining PPRC Paths,” on page 54 for considerations.

5. Establish a data bridge pair between the two **data** bridge volumes on the primary and secondary SVA subsystems.

Use the `-addrcpairs` option on the `alterdevice` or `definedevice` command. See "Establishing PPRC Data Bridge Pairs," on page 56 for considerations.

Note: Once you establish the pair, the SVA subsystem automatically performs the following actions:

– Reverses one of the PPRC paths defined in Step 4 so that it goes from the **secondary** to the **primary** SVA subsystem; this path will be used for the **status** bridge.

– Establishes a status bridge pair between the two **status** bridge volumes on the two SVA subsystems.

6. Establish PPRC pairs for the data volumes.

   Use the `-addrcpairs` option on the `alterdevice` or `definedevice` command to establish PPRC pairs. See "Establishing PPRC Pairs," on page 56 for considerations.

PPRC States

Figure 5-3, on page 59 shows how PPRC pairs can move between possible PPRC states. Each state is represented by a column in the figure, and the boxes in each column represent the actions that can be performed on a PPRC pair in that state. Some examples:

- When a PPRC pair is first created, both volumes are put in the pending state (primary pending/secondary pending). Synchronization then moves the pair to the duplex state (primary duplex/secondary duplex). Sychronization is a system-initiated action, as represented by the dashed arrow.
- For a pair in the duplex state (primary duplex/secondary duplex), whenever data is written to the primary volume the pair moves back to the pending state (primary pending/secondary pending).
- A PPRC pair in any state can be deleted. Deleting a pair moves both volumes into the simplex state (primary simplex/secondary simplex). "Simplex" means the volumes are not under PPRC control.

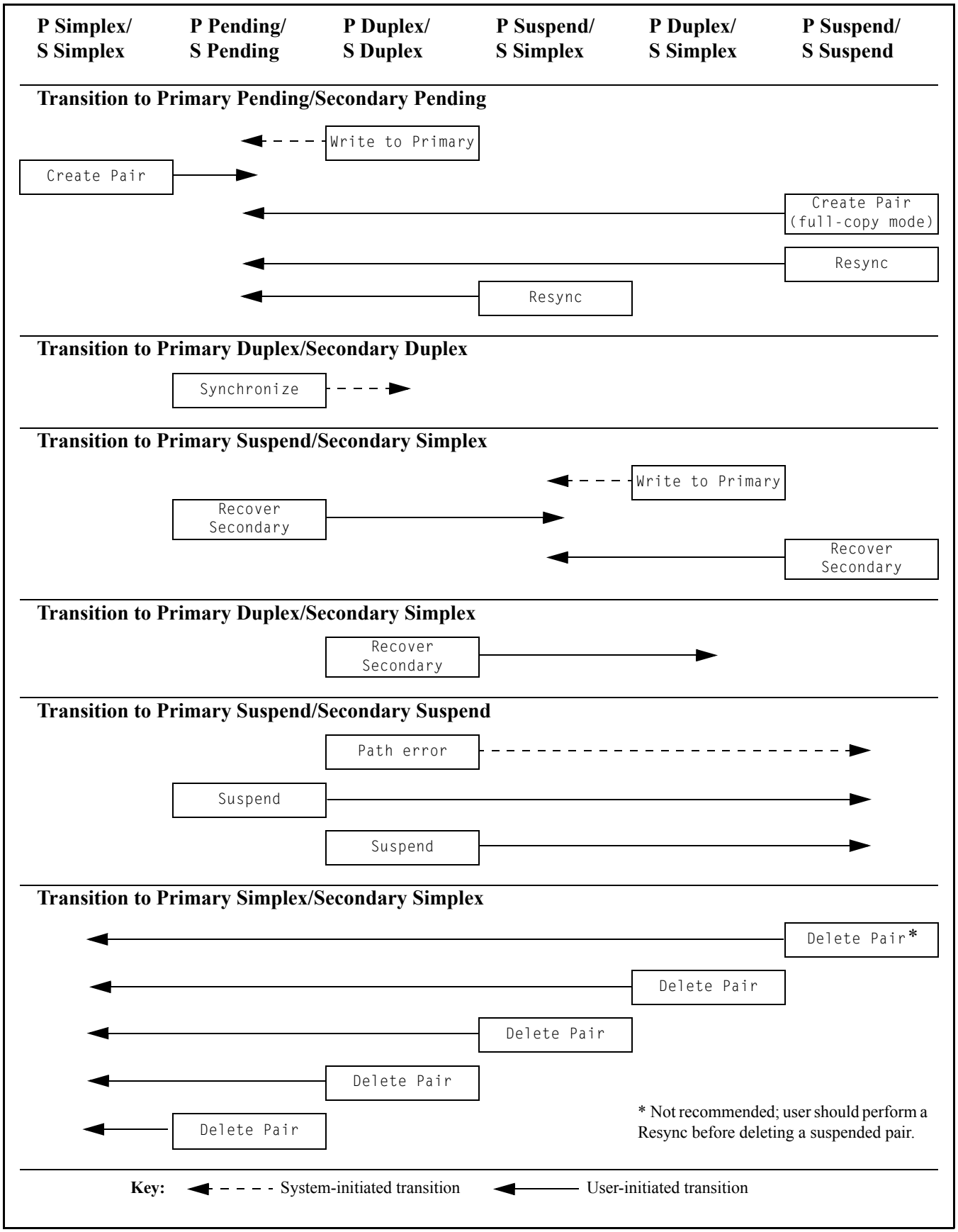


Figure 5-3 PPRC State Transitions

Synchronization of Data Within a PPRC Pair

By default, PPRC pairs are established in full-copy mode, meaning that all tracks on the primary volume are copied to the secondary volume as soon as the pair is established. Once this has occurred, the pair is said to be "synchronized" or "duplex."

Under normal circumstances, a PPRC pair is kept fully synchronized through continuous updates from the primary to the secondary volume. Synchronization can be interrupted, however, if a PPRC pair is suspended or the secondary volume is recovered. See "Suspending a PPRC Pair," on page 60 and "Recovering a Secondary Volume," on page 60 for details on these functions.

"No-copy" Mode

Optionally, you can establish a PPRC pair in "no-copy" mode, meaning that no data is copied from the primary to the secondary volume. This mode is useful if the secondary volume will be overwritten at some time in the future, as in the case of PPRC SnapShot, and you therefore wish to avoid a lengthy synchronization process. It is also useful if you are re-connecting a deleted PPRC pair, but no updates have occurred to the primary volume during the deletion period.

Establishing a pair in "no-copy" mode is done by specifying the `-addrcpairs -mode -nocopy` option on the `definedevice` or `alterdevice` command.

Suspending a PPRC Pair

A PPRC pair can be suspended at any time. Suspending a pair halts the transfer of updates from the primary to the secondary volume. While suspended, the primary volume keeps track of updates made to it, and when the PPRC pair is re-synchronized only the updated tracks are transferred to the secondary volume. See "Re-Synchronizing a PPRC Pair" for details on this process.

A PPRC pair can be suspended in either of the following ways:

- The PPRC pair is explicitly suspended with the `-modifyrcpairs -mode suspend` option on the `alterdevice` command. This command can be issued to either a primary or a secondary volume.
- Either the primary or secondary SVA subsystem detects a communication problem between the two subsystems (a PPRC path error) and suspends the PPRC pair.

Note: You cannot suspend a PPRC bridge pair.

Recovering a Secondary Volume

A secondary volume can be recovered from PPRC control at any time. Recovering a secondary volume has the following effects:

- The secondary volume is placed in the simplex state; all PPRC operations on that volume are stopped, and full host access to it is allowed (under PPRC control, direct access to an active secondary volume is not allowed). No PPRC updates are made to a secondary volume in the simplex state.
- The primary volume is placed in the suspended state when it receives the next write operation from the host. All changes made to the primary volume will be tracked so that when the pair is re-synchronized only the updated tracks

will be transferred to the secondary volume. (See "Re-Synchronizing a PPRC Pair," on page 61.

Recovering a secondary volume is done by specifying the `-modifyrcpairs -mode recover` option on the `alterdevice` command. This option can be issued to a secondary volume only.

To re-activate PPRC control over a recovered secondary volume, you can do either of the following:

- Re-establish (re-synchronize) the PPRC pair with the `-addrcpairs -mode resync` option on the `alterdevice` command. (See "Re-Synchronizing a PPRC Pair," on page 61.
- Pair the secondary volume to a different primary volume (see "Establishing PPRC Pairs," on page 56 for details). Before doing this, you must delete the original PPRC pair with the `-deletercpairs` option on the `alterdevice` command. (See "Deleting a PPRC Pair," on page 62 for details).

Note: If you are running SVA Path, after recovering a secondary volume you must explicitly instruct SVA Path to recognize the volume. See "PPRC Secondary Volumes and SVA Path," on page 53 for details.

Re-Synchronizing a PPRC Pair

Re-synchronizing a PPRC pair effectively re-establishes the pair for PPRC activity. Only tracks that were updated during the suspension period are copied from the primary volume to the secondary. Then the PPRC update process returns to normal (that is the pair is kept synchronized through continuous updates).

Re-synchronizing is done by specifying the `-addrcpairs -mode resync` option on the `alterdevice` command.

Note: You cannot re-synchronize a PPRC bridge pair.

Maintaining PPRC Paths

The following sections describe various maintenance activities that you may wish to perform on PPRC paths.

Displaying Known Subsystems

The `querysubsystem` command displays all SVA subsystems to which the local SVAA server has access. This is useful in helping you determine what information about a secondary subsystem is necessary to establish PPRC paths and pairs.

Displaying PPRC Path Status Information

The `-pprcinfo` option on the `displayiointerface` command displays the current PPRC path configuration information for the specified SVA subsystem. Information such as interface ID, internal IFID, primary SSID, secondary VCU, secondary SSID, and secondary serial number is displayed.

De-establishing a PPRC Path

The `-deletepath` option on the `alteriointerface` command deletes all logical paths associated with a specified primary VCU or SSID. Only active paths on the VCU or SSID are removed; all other paths are unaffected.

Maintaining PPRC Pairs

The following sections describe various maintenance activities you may want to perform on PPRC pairs.

Displaying PPRC Pairs

The `-pprcinfo` option on the `displaydevice` command displays PPRC status information for any device that is part of a PPRC pair. The following entries may appear in the "PPRC State" column:

- Primary—device is a primary PPRC volume
- Secondary—device is a secondary PPRC volume
- Duplex—the PPRC pair is fully synchronized
- Mixed—used for the parent device of a larger LUN only; indicates that one or more child devices have PPRC states that do not match
- Pending—the PPRC pair is currently being synchronized
- Simplex—device is a PPRC data bridge or status bridge volume that is not a member of a PPRC bridge pair
- Suspended—the PPRC pair is currently not being kept in sync

Deleting a PPRC Pair

The `-deletercpairs` option on the `alterdevice` removes the logical connection between the primary and secondary volumes of a PPRC pair. The functional devices remain, but no longer under PPRC control. This option can be issued to either volume in the PPRC pair.

Note: If you are running SVA Path, after deleting a PPRC pair you must explicitly instruct SVA Path to recognize the secondary volume. See "PPRC Secondary Volumes and SVA Path," on page 53 for details.

PPRC Restrictions on SVAA Configuration

Once you have established PPRC pairs, the following restrictions apply on how you can use SVAA to configure and manage the primary and secondary SVA subsystems:

- You cannot alter an I/O interface that is used by a PPRC pair.
- You cannot alter a functional device that is part of a PPRC pair; you must first de-establish the pair.
- You cannot delete a functional device that is part of a PPRC pair; you must first de-establish the pair.
- You cannot alter an SSID (subsystem ID) on a subsystem on which PPRC is enabled. You must first de-establish all PPRC pairs on the subsystem, delete all PPRC paths to/from the subsystem, and have your hardware support representative disable PPRC capability for the subsystem. After you alter the SSID, PPRC can be re-enabled for the subsystem.

PPRC and SnapShot

PPRC Restrictions on SnapShot

The following restrictions apply to PPRC volumes and SnapShot:

- A PPRC primary volume can be either the source or target of a snap. The implications of this depends on whether the source of the snap is under PPRC control:
  - If the source is under PPRC control, the operation is a PPRC SnapShot. See "PPRC SnapShot", below.
  - If the snap source is not under PPRC control, the operation is a "snap to primary." See "Snap to Primary", below.
- A PPRC secondary volume can be neither the source nor target of a snap. This is because a PPRC secondary volume is protected from any non-PPRC data access, except in the case of a PPRC SnapShot (see "PPRC SnapShot," on page 63.)
- A PPRC bridge volume can be neither the source nor target of a snap.

PPRC SnapShot

This section provides an overview of PPRC SnapShot, which is an SVA subsystem feature that must be ordered and installed on each subsystem involved. For complete details, see the *Peer to Peer Remote Copy Configuration Guide*. PPRC SnapShot is supported by most SVA subsystem models. Contact your StorageTek Software Support representative to see if your SVA subsystem is supported.

Note: PPRC SnapShot should not be confused with a "snap to primary" operation. See "Snap to Primary," on page 64.

PPRC SnapShot is an SVA subsystem feature in which data on a primary PPRC subsystem is replicated instantaneously on the secondary subsystem. With PPRC SnapShot, both the source and target of the snap are PPRC primary volumes. When you snap the source PPRC primary volume to a target PPRC primary volume, the corresponding source PPRC secondary volume simultaneously snaps to its corresponding target PPRC secondary volume.

As with any snap, the data replication involved in PPRC SnapShot is instantaneous and uses no additional disk space. It avoids the need to copy data from the snap target across PPRC links to the target's PPRC secondary volume.

How PPRC SnapShot Works

Figure 5-4, "Typical SnapShot Compared With PPRC SnapShot" compares a typical SnapShot operation (in which the source is a PPRC primary volume and the target is a non-PPRC volume) with PPRC SnapShot (in which both the source and target are PPRC primary volumes).

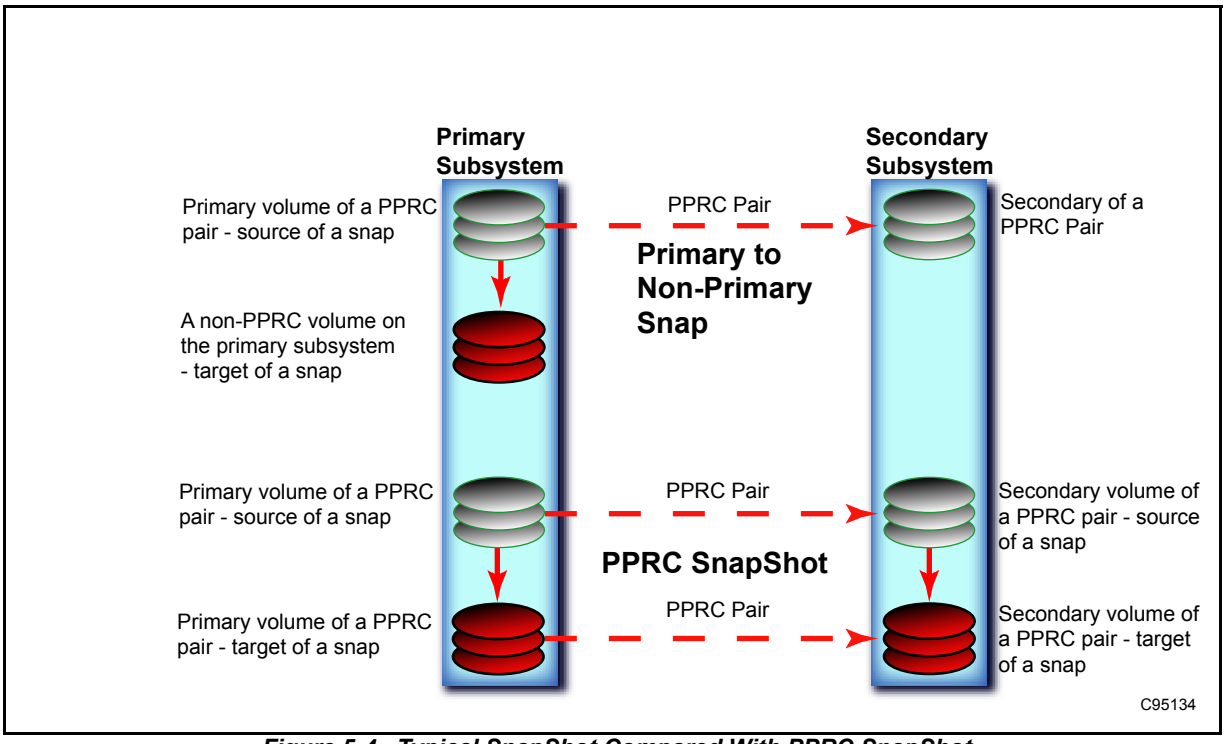


Figure 5-4 Typical SnapShot Compared With PPRC SnapShot

Following are the steps involved in a PPRC SnapShot operation:

1. The user issues a `snap` command for which both the source and target of the snap are PPRC primary volumes on the same primary SVA subsystem.
2. The data is snapped from the source to the target on the primary SVA subsystem.
3. The primary SVA subsystem issues a snap order to the secondary subsystem.
4. The secondary SVA subsystem performs a snap operation from the source PPRC secondary to the target PPRC secondary (that is, from the volume that is the PPRC secondary for the snap source, to the volume that is the PPRC secondary for the snap target).

Special Considerations

Following are special considerations for PPRC SnapShot:

- The target PPRC pair must be in the duplex state in order for the PPRC SnapShot to occur. If the target PPRC primary volume is part of a PPRC pair that is in the pending or suspended states, the primary snap is performed, but the secondary snap is not.
- You cannot use PPRC SnapShot to snap power PPRC bridge volumes
- You cannot use PPRC SnapShot to snap PPRC secondary volumes, since host access is not allowed to secondary volumes.

Snap to Primary

The “snap to primary” capability is available on all SVA subsystems supporting PPRC. It involves snapping data from a source device that is not under PPRC control to a target that is a PPRC primary volume. The tracks on the PPRC primary

volume that are updated by the SnapShot are copied across PPRC links to the PPRC secondary volume, as part of normal PPRC synchronization.

See the *Peer to Peer Remote Copy Configuration Guide* for detailed descriptions of how you can use this capability.

Note: Snap to primary should not be confused with PPRC SnapShot. See "PPRC SnapShot," on page 63.

Related CLI Commands

Table 5-1, "PPRC commands" lists commands you can use to manage PPRC. See "SVAA CLI Command Reference (Help Files)," on page 89 for detailed descriptions of the commands and their parameters.

Note: See "PPRC Administrators," on page 25 for details about defining, administering, and using the PPRC administrator group.

Table 5-1 PPRC commands

Command	Function
`alterdevice`	Establish, modify, or delete PPRC pairs. Suspend or re-synchronize PPRC pairs. Recover PPRC secondary volumes.
`alteriointerface`	Create or delete a PPRC path between primary and secondary SVA subsystems.
`definedevice`	Define bridge volumes. Establish PPRC pairs.
`displaydevice`	Display PPRC device data.
`displayiointerface`	Display PPRC path data.
`dropadministratorgroup`	Drop the PPRC administrator group.
`queryconnections`	Display permissions, including PPRC administration, for all users connected to the SVAA server.
`querypermissions`	Display permissions, including PPRC administration, for a specified user or group.
`queryadministratorgroup`	Display the PPRC administrator group.
`setadministratorgroup`	Define a PPRC administrator group.
`snap`	Perform a PPRC SnapShot operation or a "snap to primary."

Chapter 6. Point-in-Time Reports

Overview

The SVAA point-in-time reports provide detailed statistics of SVA subsystem, functional device, or I/O interface condition at a given point in time. They are produced only through the SVAA Web-based interface (WBI); they cannot be produced through the CLI (command line interface). This chapter explains how to generate and interpret the reports.

Note: The point-in-time reports online Help also contains some of this information.

There are three point-in-time reports:

- Subsystem Summary Report—summarizes the specified SVA subsystem(s), including cache size and efficiencies, read/write efficiencies, and defined device types and compression ratios.
- Device Detail Report—For each specified SVA subsystem, lists all functional devices and their DTL (domain, target, LUN), FDID (functional device ID), address, type, average usage, etc.

  Note: Currently this report offers a functional view of the SVA subsystem only; it does not offer a logical view (showing the functional devices that make up a SCSI larger LUN device).
- I/O Interface Report—For each specified SVA subsystem, lists all I/O interfaces and their interface name, type, activity level, and efficiencies.

Point-in-Time Reports and SVAA Server Cache

The point-in-time reports present a picture of the SVA subsystem as of a specific point in time. The exact point in time depends on when the report is generated and whether your installation is using SVAA server cache. (See "SVAA Server Cache," on page 17 for details.)

If you are using SVAA server cache, the data values that appear on the report come from the cache, not directly from the SVA subsystem. If you are not using SVAA server cache, the report values come from the SVA subsystem at the time the report is generated.

Because of these factors, the values shown on a point-in-time report may not exactly match comparable values displayed in SVAA displays and queries. For example, if I/O interfaces are added or removed between the time a `queryiointerface` command is submitted and when a point-in-time report is generated, the Number of I/O Interfaces shown on the two outputs will vary.

Setting up the Web-Based Interface (WBI)

A user must be able to access the SVAA server through the WBI in order to run the point-in-time reports. The WBI provides a graphical user interface (GUI) to the SVAA server.

This section instructs SVAA administrators on how to set up and maintain the SVAA Web-based interface (WBI). These functions require administrative privileges.

Enabling the WBI

You initially enable the WBI as part of SVAA server installation. For detailed instructions, see "Configure the SVAA Server" in "Chapter 2, Installation" of the *SVAA for Windows Installation Guide*.

When enabling the WBI, you assign a WBI port number to the SVAA server. The WBI port is the TCP/IP port used for communications between the WBI and the SVAA server. Enabling the WBI permits users to access the SVAA server through Web browsers connected to this port. Once enabled, the WBI remains so until explicitly disabled. There can be only one defined WBI port per SVAA server.

Changing the WBI Port Number

Changing the WBI port number for an SVAA server overwrites the entry defined in the Windows Registry. The new port number remains in effect until explicitly changed.

Note: Before changing the WBI port for an SVAA server, you must stop the SVAA server. See "Stopping the SVAA Server," on page 16 for detailed instructions.

You can use either of the following methods in the SVAA Manager program to specify the new WBI port number (see "SVAA Manager Program," on page 10 for details):

- On the **Server** screen, specify the new port number in the **Web Interface Port:** field .
- On the **Service** screen, specify `-webport` *port_number* in the **Parameters:** field, where *port_number* is the new WBI port number.

When you restart the server, the new WBI port number will be in effect.

Note: If the WBI was previously disabled for this SVAA server, specifying this parameter re-enables it on the new port. The WBI will remain enabled until explicitly disabled. See "Disabling the WBI," on page 68 for details.

Disabling the WBI

Disabling the WBI prevents all users from displaying point-in-time reports through a Web browser. The WBI remains disabled until explicitly re-enabled.

Note: Before disabling the WBI for an SVAA server, you must stop the server. See "Stopping the SVAA Server," on page 16 for detailed instructions.

You can use either of the following methods in the SVAA Manager program to disable the WBI (see "SVAA Manager Program," on page 10 for details):

- On the **Server** screen, uncheck the **Enable Web Interface** field.
- On the **Service** screen, specify `-nowebport` in the **Parameters:** field .

When you restart the server, the WBI will be disabled.

Re-enabling the WBI

Re-enabling the WBI makes it available to all users for displaying point-in-time reports. The WBI remains enabled until explicitly disabled.

Note: Before re-enabling the WBI for an SVAA server, you must stop the SVAA server. See "Stopping the SVAA Server," on page 16 for detailed instructions.

You can use either of the following methods in the SVAA Manager program to re-enable the WBI (see "SVAA Manager Program," on page 10 for details):

- On the **Server** screen, check the **Enable Web Interface** field, then specify the new port number in the **Web Interface Port:** field.
- On the **Service** screen, specify `-webport` *port_number* in the **Parameters:** field, where *port_number* is the WBI port number.

When you restart the server, the WBI will be re-enabled on the specified WBI port number.

Security Considerations

The point-in-time reports generate only query-type requests to the SVAA, therefore the reports require no special privileges. All users are allowed to generate all point-in-time reports.

Generating the Reports

This section instructs users on how to generate point-in-time reports through the Web-based interface (WBI).

Supported Browsers

The point-in-time reports are generated from a Web browser connected to the SVAA WBI. The WBI supports all of the following browsers:

- Microsoft Internet Explorer (IE) 5.5 and 6.0
- Netscape R4.7.x, R6.1, and R6.2
- Mozilla 1.0

Accessing the SVAA Server

Use the following procedure to use a Web browser to access the SVAA server.

1. Open a browser window. (See "Supported Browsers" above for browsers you can use.)
2. In the Address/Location field of the browser window, enter the following URL:

   `http://`*host_name*`:`*port_number*

   where:

   – *host_name* is the name of the host running the SVAA server you wish to access.

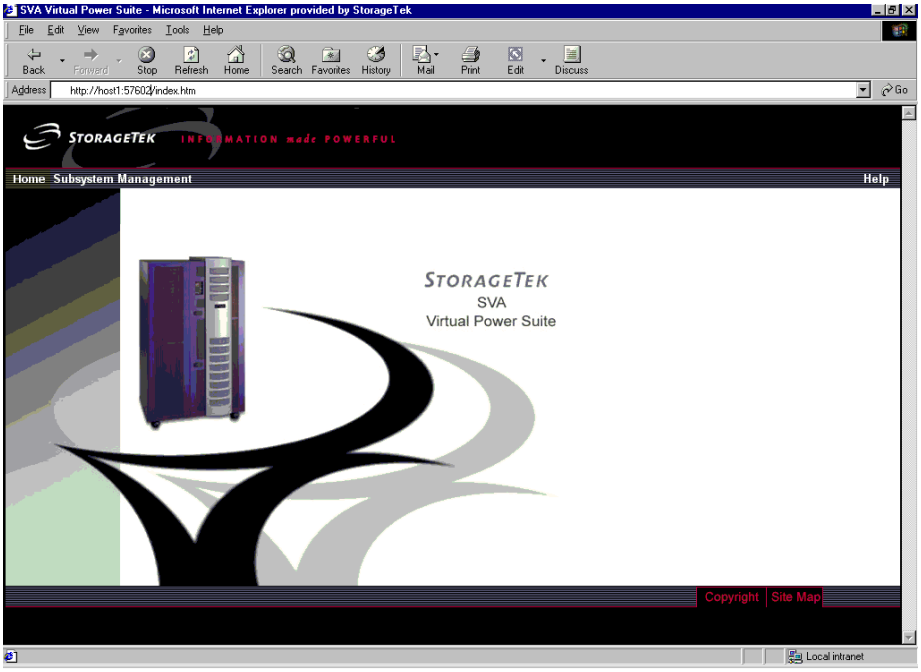
   – *port_number* is the defined WBI port on the SVAA server.

   Note: Your SVAA administrator can provide you with this information.

For example, to access an SVAA server running on a host named “host1” on port 57602, enter the following:

```
http://host1:57602
```

3. The SVA Virtual Power Suite screen appears.

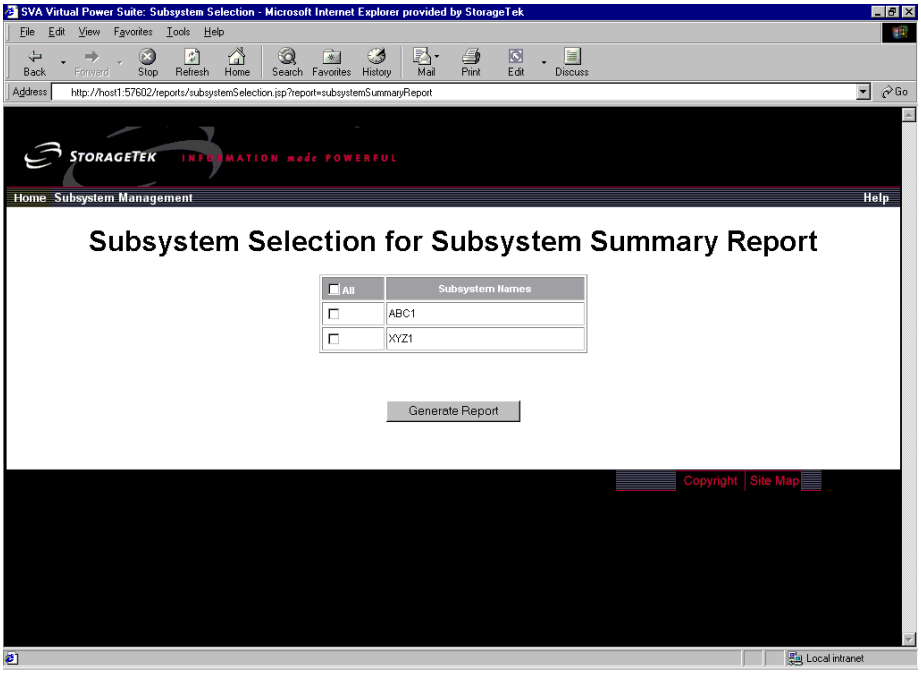


Generating a Report

Use the following procedure to generate a point-in-time report for one or more SVA subsystems.

1. From the SVA Virtual Power Suite screen, click **Subsystem Management**.
2. Click **Reports**.
3. Click the report you wish to generate. Current selections are:
   - Subsystem Summary Report
   - Device Detail Report
   - I/O Interface Report

4. The Subsystem Selection screen appears. The screen displays all SVA subsystems to which the SVAA server currently has access. All point-in-time reports use this screen.



5. Specify the SVA subsystem(s) to be included in the report:

   – Click **All**, to include all SVA subsystems.

   – Click individual subsystem names to only include those subsystems.

   Note: If the SVAA server currently does not have access to any SVA subsystems, you will see "No subsystems available."

6. Click **Generate Report**.

7. The report is generated and displayed in the browser window. See "Interpreting the Reports," on page 72 for detailed descriptions of the report fields.

Printing Reports

You can print a point-in-time report by using the standard file print feature available from your browser. See the documentation for your specific browser for detailed instructions. Typically, you would select **File>Print** from the browser pull-down menu.

Saving Reports

You can save a point-in-time report by using the standard file save feature available from your browser. See the documentation for your specific browser for detailed instructions. Typically, you would select **File>Save As** from the browser pull-down menu and assign a location and name to the report.

Note: Be sure to give the file a `.htm` or `.html` extension to identify it as an HTML file.

Interpreting the Reports

Note: This information is also available in the point-in-time reports online Help.

Note: The point-in-time reports present a picture of the SVA subsystem as of a specific point in time. Therefore, the values shown on a point-in-time report may not exactly match comparable values displayed in SVAA displays and queries. See “Point-in-Time Reports and SVAA Server Cache,” on page 67 for details.

Standard Report Header

All point-in-time reports have the same standard header. Figure 4-3, “Sample Standard Report Header” is a sample.

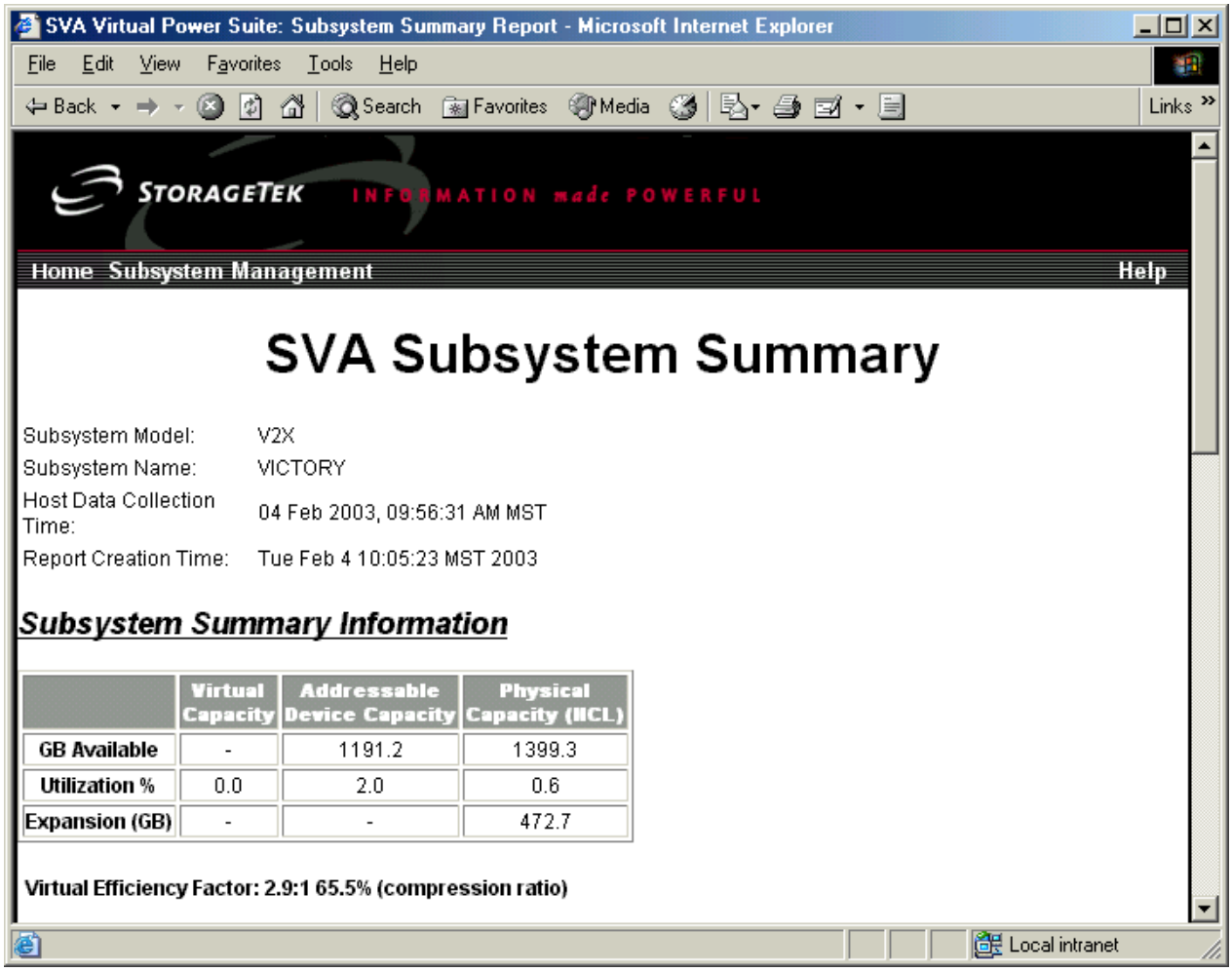


Figure 6-5 Standard Report Header

Table 6-1, “Standard Report Header Fields” provides definitions and calculations for all fields in the report header.

Table 6-1 Standard Report Header Fields

Report Field	Definition	Calculation
Subsystem Model	Storage model of the subsystem	None
Subsystem Name	Name of the subsystem. If multiple subsystems were selected, this field will be a drop down list of all of the subsystems selected.	None
Host Data Collection Time	The time when the host server finished collecting all the data for the reports.	None
Report Creation Time	Current date and time when the report is viewed.	None

GB Available: Addressable Device Capacity	The total functional capacity in GB defined for the subsystem (based on devices defined)	**SCSI devices:**
		`(logical block size * logical blocks per track * (primary capacity - 2) * tracks per cylinder) / 1000000000`
		CKD devices:
		`(bytes per track * primary capacity * tracks per cylinder) / 1000000000`

GB Available: Physical Capacity	Amount of back-end storage available in the subsystem	None
Utilization %: Virtual Capacity	The percentage of physical capacity that should be used	`(physical capacity percent utilized / 75) * 100`

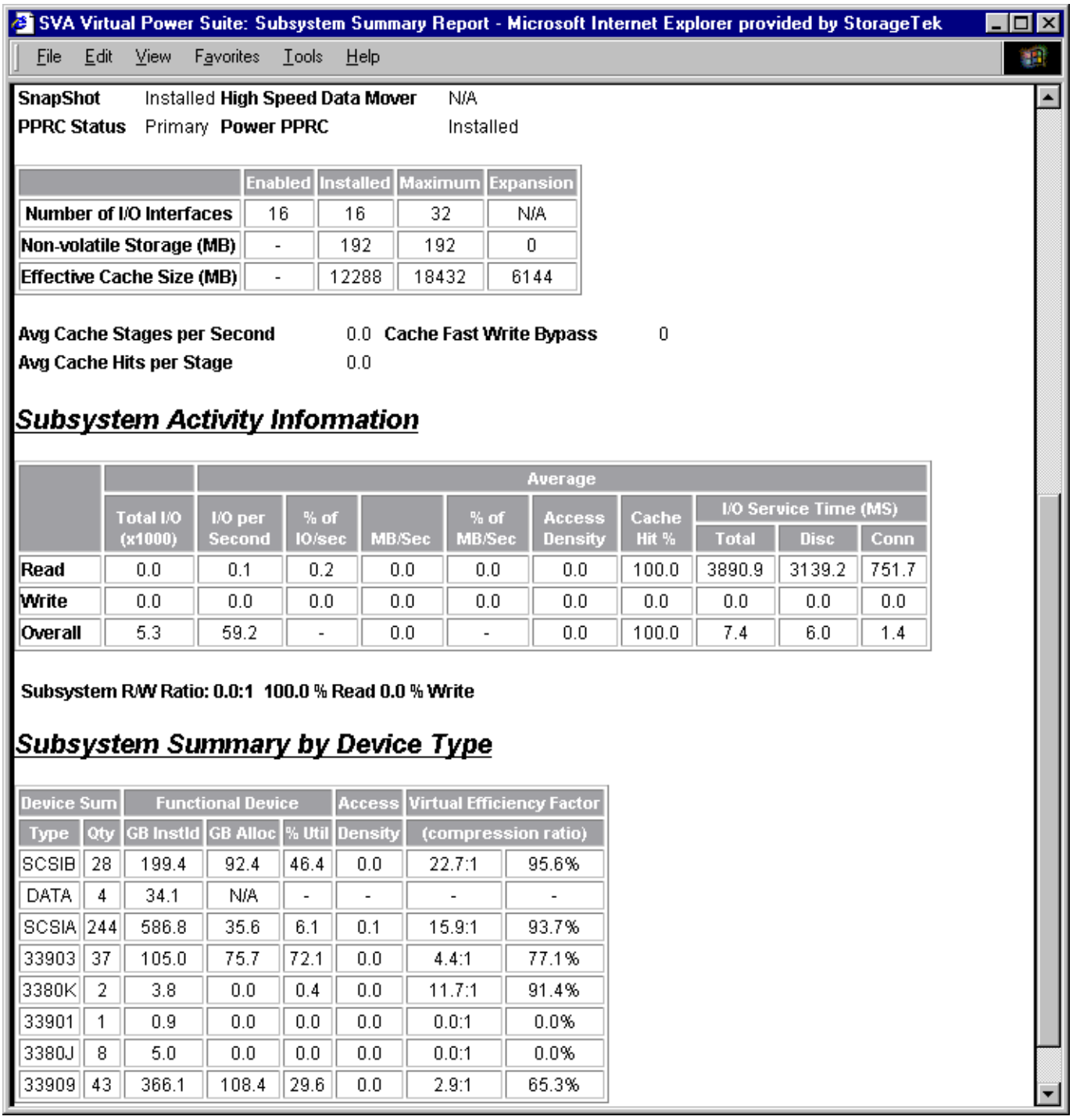
Utilization %: Addressable Device Capacity	The percentage of functional space used	`(functional capacity stored / functional capacity) * 100`
		functional capacity stored (SCSI devices):
		`logical blocks per track * logical block size * number tracks mapped`
		functional capacity stored (CKD devices):
		`bytes per track * number of tracks mapped`

Table 6-1 Standard Report Header Fields (Continued)

Utilization %: Physical Capacity	The percentage of back-end physical capacity that is used	`((total back-end capacity - free back-end capacity) / total back-end capacity) * 100`
Expansion (GB)	Potential total storage capacity if all arrays are installed	`max number of arrays * number of data drives in an array * drive capacity - physical capacity`
Virtual Efficiency Factor (compression ratio)	The approximate ratio of functional capacity stored to the physical capacity used. The percentage is the percentage of saved space.	`functional capacity stored / (total back-end capacity - free back-end capacity)` `(1 - (1 / virtual efficiency ratio)) * 100`

Subsystem Summary Report

For each SVA subsystem selected, the Subsystem Summary Report summarizes subsystem configuration, I/O activity, and functional devices. Figure 4-3, “Sample Subsystem Summary Report” is a sample of the report.



SVA Virtual Power Suite: Subsystem Summary Report - Microsoft Internet Explorer provided by StorageTek

File Edit View Favorites Tools Help

SnapShot Installed **High Speed Data Mover** N/A
PPRC Status Primary **Power PPRC** Installed

	Enabled	Installed	Maximum	Expansion
Number of I/O Interfaces	16	16	32	N/A
Non-volatile Storage (MB)	-	192	192	0
Effective Cache Size (MB)	-	12288	18432	6144

Avg Cache Stages per Second 0.0 **Cache Fast Write Bypass** 0
Avg Cache Hits per Stage 0.0

Subsystem Activity Information

		Average								
	Total I/O (x1000)	I/O per Second	% of IO/sec	MB/Sec	% of MB/Sec	Access Density	Cache Hit %	I/O Service Time (MS)		
								Total	Disc	Conn
Read	0.0	0.1	0.2	0.0	0.0	0.0	100.0	3890.9	3139.2	751.7
Write	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Overall	5.3	59.2	-	0.0	-	0.0	100.0	7.4	6.0	1.4

Subsystem R/W Ratio: 0.0:1 100.0 % Read 0.0 % Write

Subsystem Summary by Device Type

Device Sum		Functional Device			Access	Virtual Efficiency Factor	
Type	Qty	GB Instld	GB Alloc	% Util	Density	(compression ratio)	
SCSIB	28	199.4	92.4	46.4	0.0	22.7:1	95.6%
DATA	4	34.1	N/A	-	-	-	-
SCSIA	244	586.8	35.6	6.1	0.1	15.9:1	93.7%
33903	37	105.0	75.7	72.1	0.0	4.4:1	77.1%
3380K	2	3.8	0.0	0.4	0.0	11.7:1	91.4%
33901	1	0.9	0.0	0.0	0.0	0.0:1	0.0%
3380J	8	5.0	0.0	0.0	0.0	0.0:1	0.0%
33909	43	366.1	108.4	29.6	0.0	2.9:1	65.3%

Figure 6-6 Subsystem Summary Report

Table 6-2, “Subsystem Summary Report Fields” provides definitions and calculations for all fields in the report.

Report Notes:

- The report displays only device types that have been defined.

- A `Cache Fast Write Bypass` value greater than 1 indicates insufficient non-volatile storage (NVS). Call your StorageTek Customer Support Engineer.

Table 6-2 Subsystem Summary Report Fields

Report Field	Definition	Calculation
SnapShot Copy:	Indicates whether SnapShot has been installed on this subsystem.	None
High Speed Data Mover	Indicates whether High-Speed Data Mover has been installed on this subsystem.	None
PPRC Status	Indicates whether PPRC (peer-to-peer remote copy) has been installed and whether this subsystem is a primary or secondary.	None
Power PPRC	Indicates if Power PPRC (peer-to-peer remote copy) has been installed.	None
Number of IO Interfaces: Enabled	Number of IO interfaces that are currently enabled on this subsystem.	None
Number of IO Interfaces: Installed	Number of IO Interfaces that are currently installed on this subsystem.	None
Number of IO Interfaces: Maximum	The Maximum number of IO Interfaces that can be installed and enabled on this subsystem.	None
Number of IO Interfaces: Expansion	Number of interfaces that can be added to the box	`max - number installed` If the box contains both fibre and serial (ESCON) channels then this field will be N/A
Non-Volatile Storage: Installed	Effective non-volatile storage installed on the subsystem	None
Non-Volatile Storage: Maximum	Maximum effective non-volatile storage available on the subsystem	None
Non-Volatile Storage: Expansion	Amount of Non-volatile storage that could be added to the subsystem.	`Non-Volatile Storage Max – Non-Volatile Storage Installed`
Effective Cache Size: Installed	Effective customer cache available on the subsystem	None
Effective Cache Size: Maximum	Maximum effective customer cache available on the subsystem	None
Effective Cache Size: Expansion	Amount of cache that could be added to the subsystem.	`Effective Cache Size Maximum - Effective Cache Size Installed`

Table 6-2 Subsystem Summary Report Fields (Continued)

Average Cache Stages per Sec	The average number of staging operations (transfers of data from DASD storage to cache storage) per second	`(cache transfers (stages) + cache sequential transfers) / interval duration in seconds`
Average Cache Hits per Stage	The ratio of the number of channel segments for which the addressed track was in cache storage to the number of times data was transferred from DASD storage to cache storage (that is, to the number of times the addressed track was not in cache storage). This is the ratio of cache hits to cache misses	`(read hits + write hits) / (cache transfers (stages) + cache sequential transfers)`
Cache Fast Write Bypass (Subsystem Level)	The number of DASD fast write operations that were bypassed (data was written directly to DASD because insufficient non-volatile storage was available).	None
Subsystem Activity Information		
Read: Total I/O	Total number of read requests	`(read normal requests + read sequential requests + read CFW requests) / 1000`
Read: Average I/O per Second	The number of read operations per second	`Total Read I/O / interval duration in seconds`
Read: Average % of I/O per Second	The percentage of total I/O that were read requests	`(average read IO per second / overall average IO per second) * 100`
Read: Average MB per Second	The average megabytes transferred for a read request	`(read front-end bytes transferred / 1000000) / interval duration in seconds`
Read: Average % of MB per Second	The percentage of total megabytes transferred that were from read requests	`(((read front-end bytes transferred / 1000000) / ((read front-end bytes transferred / 1000000) + (write front-end bytes transferred / 1000000))) / interval duration in seconds) * 100`
Read: Average Access Density	The number of read I/O operations per second per gigabyte of functional capacity defined	`(Total Read I/O / (functional capacity / 1000000000)) / interval duration in milliseconds`

Table 6-2 Subsystem Summary Report Fields (Continued)

Read: Average Cache Hit %	The percentage of read operations for which the addressed track was present in cache storage	`(read hits / Total Read I/O) * 100`
Read: Average I/O Service Time Total	The average service time per read I/O operation in milliseconds	`((device utilization time + 500) / 1000) / Total Read I/O`
Read:Average I/O Service Time Connected	The average time the device was connected to the channel (in milliseconds) while processing a read I/O operation	`((device connect time + 500) / 1000) / Total Read I/O`
Read: Average I/O Service Time Disconnected	The average time the device was disconnected from the channel (in milliseconds) while processing a read I/O operation	`(((device utilization time + 500) / 1000) - ((device connect time+500) / 1000)) / Total Read I/O`
Write: Total I/O	Total number of write requests	`write normal requests + write sequential requests + write CFW requests`
Write: Average I/O per Second	The number of write operations per second	`Total Write I/O / interval duration in seconds`
Write: Average % of I/O per Second	The percentage of total I/O that were write requests	`(average write IO per second / overall average IO per second) * 100`
Write: Average M/B per Second	The average megabytes transferred per second for a write request	`(write front-end bytes transferred / 1000000) / interval duration in seconds`
Write: Average % of MB per Second	The percentage of total megabytes transferred that were from write requests	`(((write front-end bytes transferred / 1000000) / ((read front-end bytes transferred / 1000000) + (write front-end bytes transferred / 1000000))) / interval duration in seconds) * 100`
Write: Average Access Density	The number of write I/O operations per second per gigabyte of functional capacity defined	`(Total Write I/O / (functional capacity / 1000000000)) / interval duration in milliseconds`
Write: Average Cache Hit %	The percentage of write operations for which the addressed track was present in cache storage	`(write hits / Total Write I/O) * 100`
Write: Average I/O Service Time Total	The average service time per write I/O operation in milliseconds	`((device utilization time + 500) / 1000) / Total Write I/O`

Table 6-2 Subsystem Summary Report Fields (Continued)

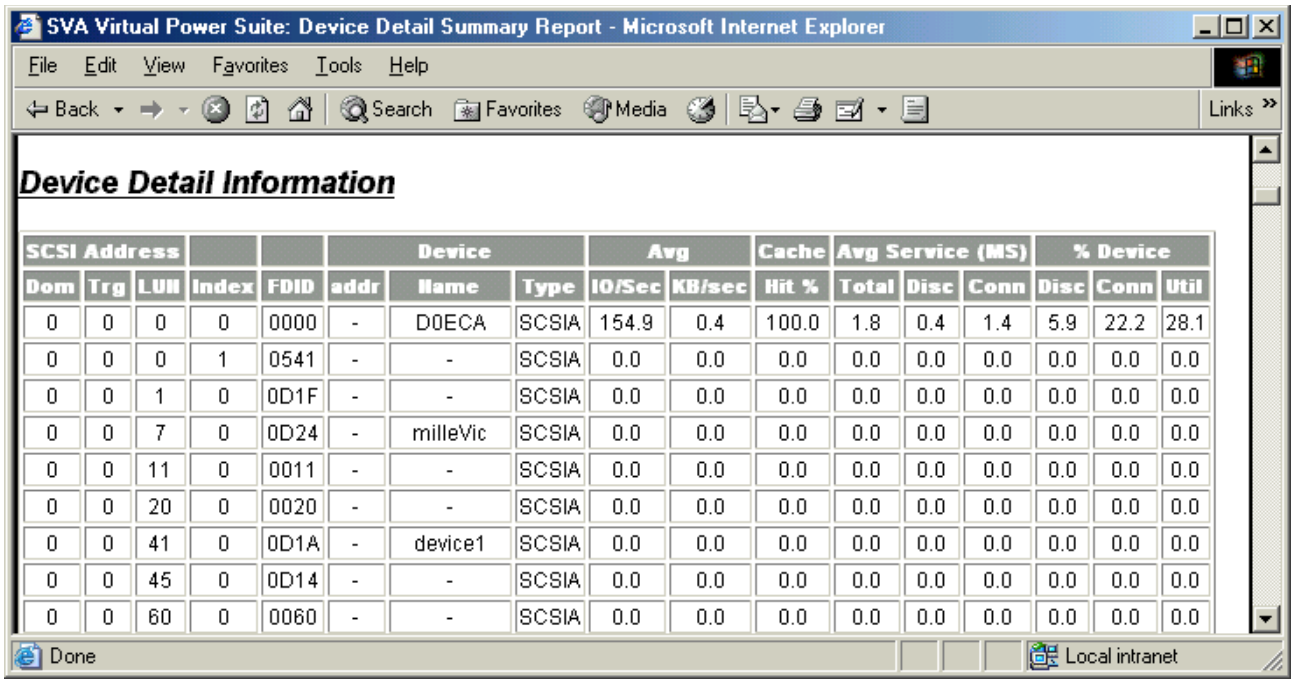
Write:Average I/O Service Time Connected	The average time the device was connected to the channel (in milliseconds) while processing a write I/O operation	`((device connect time + 500) / 1000) / Total Write I/O`
Write: Average I/O Service Time Disconnected	The average time the device was disconnected from the channel (in milliseconds) while processing a write I/O operation	`(((device utilization time + 500) / 1000) - ((device connect time + 500) / 1000)) / Total Write I/O`
Overall: Total I/O	Total device activity	None
Overall: Average I/O per Second	The number of I/O operations per second	`device activity / interval duration in seconds`
Overall: Average M/B per Second	The average megabytes transferred for an I/O request over a time interval	`((read front-end bytes transferred / 1000000) + (write front-end bytes transferred / 1000000)) / interval duration in seconds`
Overall: Average Access Density	The number of I/O operations per second per gigabyte of functional capacity defined	`(device activity / (functional capacity / 1000000000)) / interval duration in milliseconds`
Overall: Average Cache Hit %	The percentage of all I/O operations for which the addressed track was present in cache storage	`((read hits + write hits) / (Total Read I/O + Total Write I/O)) * 100`
Overall: Average I/O Service Time Total	The average service time per I/O operation in milliseconds	`((device utilization time + 500) / 1000) / device activity`
Overall:Average I/O Service Time Connected	The average time the device was connected to the channel (in milliseconds) while processing an I/O operation	`((device connect time + 500) / 1000) / device activity`
Overall: Average I/O Service Time Disconnected	The average time the device was disconnected from the channel (in milliseconds) while processing an I/O operation	`(((device utilization time + 500) / 1000) - ((device connect time + 500) / 1000)) / device activity`
Subsystem Read/Write Ratio	The number of read operations to write operations	`Total Read I/O / Total Write I/O`
Subsystem % Read	Percent of read operations	`(Total Read I/O / (Total Read I/O + Total Write I/O)) * 100`
Subsystem % Write	Percent of write operations	`(Total Write I/O / (Total Read I/O + Total Write I/O)) * 100`

Table 6-2 Subsystem Summary Report Fields (Continued)

Subsystem Summary By Device Type		
Device Type	The type of device	None
Quantity	Total number of specific device type	None
GB Installed	Functional capacity by device type	None
GB Allocated	This value is equal to the capacity stored. This field will be N/A for all PPRC bridge devices. For subsystems that are connected to multiple host platforms there is the possibility of a variance between the reported allocation and the actual value. This could happen if the report is generated while an NCL management utility (for example, DDSR) is actively running, or as the result of an abnormal termination. If the variance is more than two or three percentage points you may want to reschedule the NCL management task. This variance is not cause for undue concern.	`track capacity * number of tracks mapped`
% Utilized	Percentage of utilized space for the specific device type	`(capacity stored / functional capacity) * 100`
Capacity Stored	The functional capacity mapped (occupying back-end storage) for the device type on a subsystem.	`track capacity * number of tracks mapped`
Access Density	The number of I/O operations per second per gigabyte of functional capacity defined	`device activity / (functional capacity / 1000000000)`
Virtual Efficiency Factor (compression ratio)	The approximate ratio of functional capacity stored to the total back-end bytes for that device type The percentage is the percentage of saved space	`(capacity stored / total device type capacity)` `(1 – (1/ratio)) * 100`
total device type capacity	Total back-end bytes for the device	Summation of back-end bytes per extent for all devices of the same type

Device Detail Report

The Device Detail Report provides detail information about all functional devices on the specified SVA subsystem(s). Figure 6-7, "Device Detail Report" is a sample of the report.



SCSI Address					Device			Avg		Cache	Avg Service (MS)			% Device		
Dom	Trg	LUN	Index	FDID	addr	Name	Type	IO/Sec	KB/sec	Hit %	Total	Disc	Conn	Disc	Conn	Util
0	0	0	0	0000	-	D0ECA	SCSIA	154.9	0.4	100.0	1.8	0.4	1.4	5.9	22.2	28.1
0	0	0	1	0541	-	-	SCSIA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0	0	1	0	0D1F	-	-	SCSIA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0	0	7	0	0D24	-	milleVic	SCSIA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0	0	11	0	0011	-	-	SCSIA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0	0	20	0	0020	-	-	SCSIA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0	0	41	0	0D1A	-	device1	SCSIA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0	0	45	0	0D14	-	-	SCSIA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0	0	60	0	0060	-	-	SCSIA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 6-7 Device Detail Report

Table 6-3, "Device Detail Report Fields" provides definitions and calculations for all fields in the report.

Report Notes:

- If the `%Device Util` value is greater than 50.0 for four consecutive hours, call your StorageTek Customer Support Engineer.

Table 6-3 Device Detail Report Fields

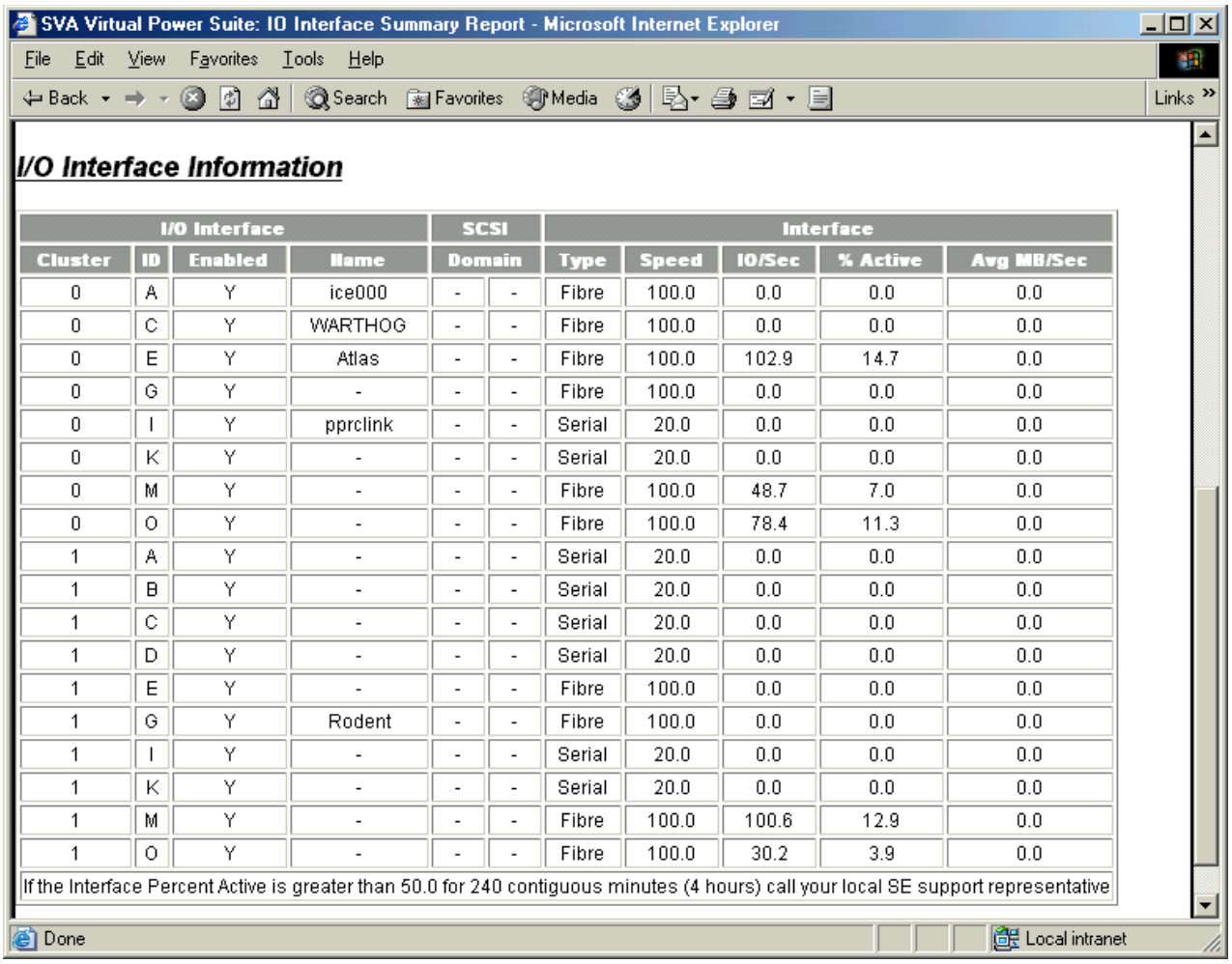
Report Field	Definition	Calculation
SCSI Address: DOM TGT LUN	The domain, target, and LUN (DTL) of this SCSI device.	None
Index	For SCSI Larger LUNs only. The index of the device in the SCSI Larger LUN. Index 0 indicates the parent device.	None
FDID	FDID (functional device ID) of this device.	None
Device Address	OS/390 only. The address of the device as defined in the IOCP	None
Device Name	Name of the device	None
Device Type	The type of DASD of the device	None

Table 6-3 Device Detail Report Fields (Continued)

Average I/O per Second	The average number of I/O operations per second for the device	`device activity / duration interval in seconds`
Average KB per Second	The amount of data (in kilobytes) transferred per second between the host and the subsystem.	`((front-end read bytes transferred + front-end write bytes transferred) / 1000) / duration interval in seconds`
Cache Hit %	The percentage of all I/O operations for which the addressed track was present in cache storage	`((read normal hits + read sequential hits + read CFW hits + DFW sequential hits + DFW normal hits + write CFW hits) / (read normal requests + read sequential requests + read CFW requests + write normal requests + write sequential requests + write CFW requests)) * 100`
Average Service (MS): Total	The average service time per I/O operation in milliseconds	`((Device utilization time + 500) / 1000) / device activity`
Average Service (MS): Disconnect	The average time the device was disconnected from the channel (in milliseconds) while processing an I/O operation.	`(((Device utilization time + 500) / 1000) - ((connect time + 500) / 1000))) / device activity`
Average Service (MS): Connected	The average time the device was connected to the channel (in milliseconds) while processing an I/O operation.	`((connect time + 500) / 1000) / device activity`
% Device Disconnected	The percentage of the interval in which the device was disconnected from the channel while processing an I/O operation	`((((Device utilization time + 500) / 1000) - ((connect time + 500) / 1000))) / duration interval in milliseconds) * 100`
% Device Connected	The percentage of the interval in which the device was connected to the channel while processing an I/O operation	`(((connect time + 500) / 1000) / duration interval in milliseconds) * 100`
% Device Utilized	The percentage of the interval in which the device was utilized (busy).	`(((Device utilization time + 500) / 1000) / duration interval in milliseconds) * 100`

I/O Interface Report

The I/O Interface Report lists detail information about all I/O interfaces on the specified SVA subsystem(s). Figure 6-8, “I/O Interface Report” is a sample of the report.



I/O Interface				SCSI		Interface				
Cluster	ID	Enabled	Name	Domain		Type	Speed	IO/Sec	% Active	Avg MB/Sec
0	A	Y	ice000	-	-	Fibre	100.0	0.0	0.0	0.0
0	C	Y	WARTHOG	-	-	Fibre	100.0	0.0	0.0	0.0
0	E	Y	Atlas	-	-	Fibre	100.0	102.9	14.7	0.0
0	G	Y	-	-	-	Fibre	100.0	0.0	0.0	0.0
0	I	Y	pprclink	-	-	Serial	20.0	0.0	0.0	0.0
0	K	Y	-	-	-	Serial	20.0	0.0	0.0	0.0
0	M	Y	-	-	-	Fibre	100.0	48.7	7.0	0.0
0	O	Y	-	-	-	Fibre	100.0	78.4	11.3	0.0
1	A	Y	-	-	-	Serial	20.0	0.0	0.0	0.0
1	B	Y	-	-	-	Serial	20.0	0.0	0.0	0.0
1	C	Y	-	-	-	Serial	20.0	0.0	0.0	0.0
1	D	Y	-	-	-	Serial	20.0	0.0	0.0	0.0
1	E	Y	-	-	-	Fibre	100.0	0.0	0.0	0.0
1	G	Y	Rodent	-	-	Fibre	100.0	0.0	0.0	0.0
1	I	Y	-	-	-	Serial	20.0	0.0	0.0	0.0
1	K	Y	-	-	-	Serial	20.0	0.0	0.0	0.0
1	M	Y	-	-	-	Fibre	100.0	100.6	12.9	0.0
1	O	Y	-	-	-	Fibre	100.0	30.2	3.9	0.0
If the Interface Percent Active is greater than 50.0 for 240 contiguous minutes (4 hours) call your local SE support representative										

Figure 6-8 I/O Interface Report

Table 6-4, “I/O Interface Report Fields” provides definitions and calculations for all fields in the report.

Report Notes:

- If the `%Active` value is greater than 50.0 for 240 consecutive minutes (four hours), call your local StorageTek Customer Support Engineer.

Table 6-4 I/O Interface Report Fields

Report Field	Definition	Calculation
I/O Interface Cluster	The cluster number to which this I/O interface is assigned, if applicable.	None
I/O Interface ID	The ID of this I/O interface.	None
I/O Interface Enabled	Indicates whether the I/O interface is currently enabled (Y/N).	None
Name	Name of the I/O interface	None
SCSI Domain	The SCSI Domain of the interface when an XSA is attached to the interface.	None
Type	Type of interface	None
Speed	The transfer rate (in megabytes) of the interface	`Channel Speed / 10`
IO/sec	The average number of I/O operations per second for the interface	`Number of I/Os / duration interval in seconds / 1000`
% Active	The percentage of the interval that the subsystem was active on this interface. The time that the subsystem was active on the interface is the total of all the connect times on the interface.	`((time ctl unit busy + 500) / 1000) / duration interval in milliseconds * 100`
Average MB/sec	The percentage of interface activity in regards to total interface activity	`Total Average MB/Sec * (((time control unit busy + 500) / 1000) / interval duration) / (((total control unit busy time + 500) / 1000) / interval duration in seconds)`

Chapter 7. SVAA CLI Commands

This chapter describes the SVAA Command Line Interface (CLI), a user interface used to issue commands to an SVAA server. To use the CLI, you must be logged in (either locally or remotely) to the computer where the SVAA server is running.

The CLI can be used in the following ways:

- From the Windows **Start** menu, click **Programs>StorageTek>Shared Virtual Array Administrator 3.1.0>SIBADMIN**. You can then issue one of the commands in this chapter (except for `sibadmin`).
- From a Command Prompt window, enter the `sibadmin` command to access the SVAA CLI shell. You can then issue one of the commands in this chapter (except for `sibadmin`).
- Include scripted CLI commands in a Windows batch file, `sibadmin.bat`. Using this method you can automate SVAA functions.

Advantages of using the CLI shell are that commands are shorter and that you do not need to be concerned about certain Windows shell restrictions (see "Windows Shell Restrictions," on page 87). On the other hand, while using the CLI shell, you cannot redirect input and output, as you can while using the Windows shell directly.

Command Format

The generic command format is a command verb followed by one or more parameters. Depending on the command, some parameters are required and some are optional.

Most SVAA CLI commands have both a long and a short format.You can use either one. For example, the command to display I/O interface information has the following forms:

- long format: `displayiointerface`
- short format: `dioint`

Rules for Entering Commands

For typographic conventions used in this book for showing command syntax and examples, see "Conventions," on page xiii.

Follow these rules when entering CLI commands and their parameters:

- Enter each parameter name only once—SVAA does not allow duplicate parameters.
- The parameter name must be preceded by a dash (-). For example, `-subsys`.

- Follow each parameter name by the value(s) for that parameter.
- Except where indicated, parameter values are case-sensitive.
- Enter each parameter value only once where multiple values are allowed—SVAA ignores duplicate values (for example, an entry of `-name CORC1,CORC1,CORC2` will be interpreted as `-name CORC1,CORC2`).
- You can arrange the parameters in any order.
- Where indicated, you can use the standard wildcard symbols `*` and `?`. For example: `-subsys ABC* -ifid 0:?`.
- When a parameter supports both lists and ranges, you can use both at once. For example, `-aryid 1:4,7`.
- When a parameter supports a list of values, you must separate the values by commas (,). For example, `-aryid 1,3,6`.

  Note: The Windows command line shell treats a comma (,) as an argument separator, the same as a blank space. For this reason, if you include CLI commands in a batch file (`sibadmin.bat`), the arguments must all be enclosed in quotation marks (`" "`) or the commands will not be parsed correctly. For example:
  `sibadmin.bat displaydevice "-subsys SOLAR -fdid 23,25"`
- Rules for entering ranges:
  - You must separate the “from” and “to” entries with a colon (:). For example: `-aryid 2:7`.
  - The “from” and “to” entries must be entered in ascending order. For example: `20:50` is valid; `50:20` is not.
- Rules specific to entering string ranges:
  - The “from” and “to” entries must be the same length. For example: `-subsys XYZ00:XYZ12` and `-name 00ABC:05ABC` are valid; `-subsys XYZ0:XYZ12` and `-name 00ABC:5ABC` are not.
  - The alpha portions of the two entries must be the same. For example: `-name A01:A05` and `-subsys ABC1:ABC7` are valid; `-name A01:B02` and `-subsys ABC:XYZ` are not.
  - The “from” numeric value must be less than the “to” numeric value. For example: `-subsys ABC00:ABC05` is valid; `-subsys ABC05:ABC00` is not.
  - Neither entry can contain wildcards (* or ?). For example: `-name A?1:A?C5` and `-subsys ABC01:ABC*` are not valid.

Using the SVAA CLI

Starting the SVAA CLI Shell

You can start the CLI either from the Windows **Start** menu or from a Command Prompt window.

To start the CLI from the Windows **Start** menu, click **Programs>StorageTek>Shared Virtual Array Administrator 3.1.0>sibadmin**. This will open a Command Prompt window displaying the CLI prompt, `SIB>`. You can then issue one of the commands in this chapter (except for `sibadmin`). Using this method, you cannot use any of the optional CLI startup parameters (see the `sibadmin` man page for details).

To start the CLI from a Command Prompt window, use the `sibadmin` command, as in the following example:

```
C:\> sibadmin
SIB>
```

Using this method, you can specify optional CLI startup parameters at the time you issue the `sibadmin` command.

The `SIB>` prompt indicates that the CLI is awaiting commands from the user.

Entering Commands

From the CLI shell prompt, you can enter CLI commands directly. For example:

```
SIB> displayiointerface -subsys SubSys02 -ifid 1.B,1.D
SIB>
```

Exiting the SVAA CLI Shell Interpreter

To quit the CLI shell and return to the system prompt, use the `exit` command:

```
SIB> exit
C:\>
```

Displaying the SVAA CLI Version

You can display the current version of the SVAA CLI software at any time by entering the following command in a Command Prompt window:

```
C:\> sibadmin -version
```

Issuing SVAA CLI Commands From a Windows Command Prompt

From a Windows Command Prompt, you can issue any CLI command using the following format:

```
C:\> sibadmin command parameters
```

For example:

```
C:\> sibadmin ddev -subsys SVA2 -fdid C02
```

Windows Shell Restrictions

When you issue a command from a Windows Command Prompt, the command is first interpreted by the system shell itself; then, if the command is valid, the command is started. Therefore, you should use care when using wildcards (*) and quotes (') from the system shell. When a CLI command requires such characters, they must be preceded with a backslash (\); this will allow the characters to be transmitted to the command as is.

The following commands are not correct:

```
C:\> sibadmin dioint -subsys SVASYS00 -ifid *
C:\> sibadmin dioint -subsys 'SVAsuba1' -ifid 0.?
```

The correct forms ares:

```
C:\> sibadmin dioint -subsys SVASYS00 -ifid \*
C:\> sibadmin dioint -subsys \'SVAsuba1\' -ifid 0.\?
```

Additionally , the Windows system shell treats a comma (,) as an argument separator, like a space. Therefore, when using a Command Prompt to issue commands containing commas you should do either of the following:

- Enclose the comma (,) in double-quotes. For example:

  ```
  C:\> sibadmin ddev -subsys SVASYS00 -fdid 23","24
  ```

- Enclose the entire command in double-quotes. For example:

  ```
  C:\> sibadmin "ddev -subsys SVASYS00 -fdid 23,24"
  ```

Redirecting Command Input and Output

When issuing commands from a Windows Command Prompt, you can redirect the command input and/or output using standard redirection methods.

Note: You cannot use these methods to redirect the input or output when issuing commands from within the CLI shell.

Input

By default, CLI command input is through the standard input device, `stdin`, attached to the current terminal session. To use input from a file, you can use the < symbol, as in the following example, which directs the contents of the `/tmp/inputdioint.txt` file into the `sibadmin` command:

```
C:\> sibadmin < /tmp/inputdioint.txt
```

You can also use the pipe (`|`) symbol, as in the following example, which pipes the contents of the `/tmp/inputdioint.txt` file into the `sibadmin` command.

```
C:\> cat /tmp/inputdioint.txt | sibadmin
```

Output

By default, CLI command output is directed to the standard output device `stdout`. To direct output to a file, use the > symbol, as in the following example, which directs the output of the `queryversion` command into the `/tmp/outpqvers.txt` file.

```
C:\> sibadmin qvers > /tmp/outpqvers.txt
```

Displaying SVAA CLI Help Files

SVAA provides help files for each CLI command. To access a help file, use the `help` command from within the CLI. You can specify either the full command name or the abbreviation. For example, to display help for the SVAA `deletedevice` command, enter either:

```
SIB> help deldev
```

or

```
SIB> help deletedevice
```

SVAA CLI Command Reference (Help Files)

The following pages describe the CLI commands, arranged in alphabetical order.

Note: The following section also includes the `sibadmin` command, which is not a CLI command. It can be issued only from the Windows **Start** menu or from a Command Prompt window. (`sibadmin` is used to start the CLI.)

addsubsystempath

Name `addsubsystempath` — add a new device path to the SVA subsystem.

Abbreviation `addsubsyspath`

Synopsis `addsubsystempath -devpath` *device_path*

Description This command provides the SVAA server with an additional path to an SVA subsystem. Devices added with this command are known as ECAM devices. (This command is equivalent to the `SET ECAMDEVICE` subcommand in SVAA for S/390.

When you use this command, the SVAA server probes the specified device path to verify that the device is in a valid SVA subsystem; if it is, the server associates the device path with the SVA subsystem name that has been defined at the subsystem LOP (local operator panel).

This command updates the Windows Registry, so ECAM devices defined with this command do not have to be redefined if the host system is rebooted.

Note: If the subsystem path is to be used for commands that modify the SVA subsystem configuration (such as `definedevice` and `deletedevice`), the device path must be a privileged ECAM device. See the `alterdevice` or `definedevice` command for information on defining a privileged ECAM device.

Security Requires SVAA administrator privileges.

Options `-devpath` *device_path*

Specifies the name of the new path to the SVA subsystem.

This is the same as the Windows drive letter. For example, the *device_path* for drive D: would be `D:` or `d:`. Entry is not case-sensitive.

Note: The device you specify must have been configured with the Disk Management administrative tool. See "Configuring New Functional Devices on the Windows Computer," on page 39.

Examples **Example 1:**

```
SIB> addsubsyspath -devpath g:
```

Defines the path `G:` to be an ECAM device.

See Also `querysubsystempath`
`removesubsystempath`

alterdevice

Name `alterdevice` — change the definition of a functional or SCSI larger LUN device, or change a functional device into a SCSI larger LUN device.

Abbreviation `altdev`

Synopsis

```
alterdevice -subsys subsys_name
-fdid fdid...|-devpath device_path[,device_path]
[-adjustfcapa +/-size|-fcapa new_size [-fdidpool fdid...]]
[-cache ena|enable|dis|disable] [-ckdena yes|no]
[-ckdrw yes|no] [-name device_name...] [-noconfirm]
[-privlg yes|no] [-scsiaddr dis|disable|domain.target.lun...]
[-scsiena yes|no] [-scsirw yes|no]
```

for PPRC (peer-to-peer remote copy):

Note: Only certain SVAA platforms and SVA subsystem configurations support the use of PPRC commands and functions. Contact StorageTek Software Support for details.

```
[-addrcpairs
-trgssname subsys_name|-trgserialnumber serial_number
-trgssid ssid_number|-trgvcu vcu_number
[-trgid id...][-mode sync_mode]]|
[-modifyrcpairs -mode pprc_mode]|[-deletercpairs]
```

Description This command changes the definition of one or more functional or SCSI larger LUN devices. It has no defaults. At least one of the optional parameters must be used. You can identify the device to be modified by specifying either its FDID (functional device ID) or its host-specific device path, but not both.

Note: Any parameter not specified retains its current setting.

Note: You cannot issue this command against an ECAM device that is currently being used for communication between the SVAA server and SVA subsystem unless there is at least one additional privileged ECAM device between the two.

With this command you can make the following modifications to a functional device:

- Change the device name
- Enable/disable a SCSI device
- Enable/disable SCSI read/write capability
- Change the domain, target, and LUN for a SCSI device (the device must be disabled first)
- Enable/disable a CKD device
- Enable/disable CKD read/write capability
- Change the privileged ECAM status for the device

- Change a functional device into a SCSI larger LUN device
- Create PPRC pairs (use the `-addrcpairs` parameter and its supporting parameters)
- Modify PPRC pairs (use the `-modifyrcpairs` parameter and its supporting option)
- Delete PPRC pairs (use the `-deletercpairs` parameter)

With this command you can make the following modifications to a SCSI larger LUN device (no other modifications are allowed):

- Change the device name
- Enable/disable the device as a CKD device
- Enable/disable CKD device read/write capability
- Change the privileged ECAM status for the device
- Increase the capacity of the device
- Create, modify, or delete PPRC pairs. Using the `-devpath` parameter, you can create, modify, or delete PPRC pairs for all devices (parent and all children) within the SCSI larger LUN device at one time.

Note: You can set up PPRC only on devices that all exist within the same subsystem ID (SSID)/virtual control unit (VCU). To set up PPRC on multiple SSIDs/VCUs, you must issue multiple `alterdevice` commands. You can modify devices and add PPRC pairs in the same command if the secondary volumes already exist.

Note: The following restrictions apply to this command:

- You cannot alter an existing device's FDID or device type. Instead, you must delete the device and then redefine it with the new characteristics. If you want to retain the data on the device, you must first back it up. Then, once you have redefined the device, you can restore the data.
- For SCSI larger LUN devices, you can use this command to change only the characteristics explicitly listed above. If you want to change any other characteristics of a SCSI larger LUN, you must: 1) define a new larger LUN device with the desired characteristics (this device should be assigned the same domain and target, but different LUN and FDID, as the device you are changing); 2) move any data from the old larger LUN device to the new one; 3) delete the old device.
- You cannot change the characteristics of a device that is part of an active PPRC pair or is a PPRC bridge volume; you must first use the `alterdevice -deletercpairs` command to delete the pair definition, then use the `alterdevice -addrcpairs` command to redefine the pair with the new characteristics.

Security For non-PPRC-related data: Requires write access to a privileged ECAM device.

For PPRC-related data: If a PPRC administrator group has been defined, requires PPRC administrator privileges; otherwise, requires write access to a privileged ECAM device.

Options

`-subsys` *subsys_name*

Specifies the name of the SVA subsystem with which the device(s) is(are) associated. You can specify only one SVA subsystem name. Entry is case-sensitive, and wildcards are not supported.

Note: You must use either the `-fdid` or the `-devpath` parameter to specify the device(s) to be modified; you cannot use both.

`-fdid` *fdid...*

Specifies the subsystem FDID (functional device ID) for the device(s) you wish to modify. In the case of a SCSI larger LUN device, this must be the parent (index 0) device.

Entry is one to three hexadecimal digits (0 to 3FF for V960 and earlier subsystems; 0 to FFF for V2X subsystems). Decimal representation is not supported. You can use ranges and lists, but not wildcards.

Note: If you specify a SCSI larger LUN device, you can use the following options only: `-cache`, `-ckdena`, `-ckdrw`, `-name`, `-privlg`, `-adjustfcapa +`, `-fcapa` (to increase the capacity only), `-fdidpool`, `-noconfirm`.

`-devpath` *device_path*`[,`*device_path*`]`

Note: This parameter cannot be used with the `-scsiaddr` parameter.

Note: This parameter cannot be used with the `-scsiena yes` option; this is because the specified *device_path* must already be enabled for SCSI access. This parameter can be used, however, with the `-scsiena no` option, which has the effect of disabling host access to the SCSI device.

device_path specifies the path from the host to the device(s) you wish to modify. The path must be enabled for SCSI access.Enter a drive letter (such as `G:` or `g:`).

You can specify a list of devices but not a range. Entry is not case-sensitive. Wildcards are not supported.

Note: You can use either the `-adjustfcapa` or the `-fcapa` parameter, but not both.

`-adjustfcapa +/-`*size*

Note: This parameter cannot be used with the `-addrcpairs`, `-modifyrcpairs`, or `-deletercpairs` parameters.

Optional parameter. Used either to change an existing functional device into a SCSI larger LUN device, or to increase the capacity of an existing SCSI larger LUN device:

– To change an existing functional device into a SCSI larger LUN device, specify an increase to the capacity of the device. The *size* you specify

is added to the device's current capacity to produce the total capacity of the larger LUN (for example, if the functional device's current capacity is 8.5G and you specify a *size* of +15G, the capacity of the larger LUN will be 23.5G).

– To increase the capacity of an existing SCSI larger LUN device, enter a value prefixed with +.

Note: You should always back up data on a device before altering its capacity.

Note: Currently, specifying a capacity prefixed with - has no effect.

Entry is a decimal integer (+1 to +32767) followed by M for megabytes or G for gigabytes (examples: `+90M`, `+10g`). Entry is not case-sensitive.

When you use this parameter to change a functional device into a SCSI larger LUN device or increase the size of a SCSI larger LUN device, you can optionally specify a pool of FDIDs to use. If you do not specify an FDID pool, the system chooses any available FDIDs. New capacity is added only in integral numbers of devices of the existing device type (for example, if the specified device is a SCSI-A device, the system only uses SCSI-A devices).

Note: By default, a confirmation prompt is displayed when you use this parameter. You can bypass this prompt with the `-noconfirm` parameter.

`-fcapa` *new_size*

Note: This parameter cannot be used with the `-addrcpairs`, `-modifyrcpairs`, or `-deletercpairs` parameters.

Optional parameter. Used either to change an existing functional device into a SCSI larger LUN device, or to increase the capacity of an existing SCSI larger LUN:

– To change an existing functional device into a SCSI larger LUN device, specify a new total capacity.

– To increase the capacity of an existing SCSI larger LUN, enter a new total capacity.

Entry is a decimal integer (1 to 65535) followed by M for megabytes or G for gigabytes (examples: `1000M`, `50g`). Entry is not case-sensitive.

When you use this parameter to change a functional device into a SCSI larger LUN device or increase the size of a SCSI larger LUN, you can optionally specify a pool of FDIDs to use. If you do not specify an FDID pool, the system chooses any available FDIDs. New capacity is added only in integral numbers of devices of the existing device type (for example, if the specified device is a SCSI-A device, the system only uses SCSI-A devices).

Note: A confirmation prompt is displayed when you use this parameter. You can bypass this prompt with the `-noconfirm` parameter.

`-fdidpool` *fdid...*

Note: To use this parameter, you must also use either the `-adjustfcapa` or `-fcapa` parameter.

Optional parameter. Used only when you change a functional device into a SCSI larger LUN device by increasing its size, or when you increase the size of a SCSI larger LUN device.

Specifies the pool of FDIDs to be used to increase the device's size. The devices must all be of the same device type (that is, either all SCSI-A or all SCSI-B). When you increase the size of a SCSI larger LUN device, the system automatically adds functional devices of the proper type.

Any devices in the pool that have already been used in another SCSI larger LUN device are ignored in this request.

The request fails if the capacity of the devices in the pool is less than the requested capacity increase. If, however, you specify more devices than are actually needed to fulfill the request, the extra ones are ignored. For example, if you specify four SCSI-B (8.5GB capacity each) devices to fulfill a request for an increase of 24GB, the fourth device will not be used.

Entry is one to three hexadecimal digits (0 to 3FF for V960 and earlier subsystems; 0 to FFF for V2X subsystems). Decimal representation is not supported. You can use ranges and lists, but not wildcards.

`-cache ena|enable|dis|disable`

Optional parameter. Enables or disables caching for the device. Entry is case-sensitive. With cache disabled, data for this device is immediately written to back-end storage.

Note: Setting `-cache` to `disable` actually has no effect on the SVA subsystem. SVA architecture prohibits cache from being disabled.

`-ckdena yes|no`

Optional parameter. Specifies whether to enable S/390 host access to the CKD functional device. If this parameter is set to `yes`, S/390 hosts will be able to access the device. Entry is case-sensitive.

Note: This parameter must be set to `yes` for all devices that are part of a PPRC pair or PPRC bridge pair.

`-ckdrw yes|no`

Optional parameter. Specifies whether to allow S/390 host write access to the CKD functional device. If this parameter is set to `yes`, S/390 hosts will be able to write data to the device. Entry is case-sensitive.

Note: This parameter must be set to `yes` for all devices that are part of a PPRC pair or PPRC bridge pair.

`-name` *device_name...*

Optional parameter. Names you want to assign to the devices identified by the `-fdid` or `-devpath` parameter. You can use ranges or lists. (See the `sibadmin` command reference for rules about entering string ranges.)

Entry is one to eight characters, or a null string. The following characters are accepted, even as the first one: a to z, A to Z, 0 to 9, $, @, #, -, _, +, &, ., and

/. A null string, entered as `-name ''` or `-name ""`, sets the name to all blanks.

Note: For SCSI devices, you may want to use this field to map the drive designation on the Windows host to the DTL (domain, target, and LUN) on the SVA subsystem. For example, a functional device assigned on the Windows host to drive `L:` could be given a *device_name* of `L` or `drive_L`.

`-noconfirm`

Optional parameter. Indicates that you do not want a confirmation prompt displayed when you use either the `-adjustfcapa` or the `-fcapa` parameter to increase the capacity of the device.

`-privlg yes|no`

Optional parameter. Specifies whether the device is eligible for use as the transmission path for privileged ECAM messages to the SVA subsystem.

You can specify `no` for most devices—to limit the number of privileged ECAM devices—but you must specify `yes` for at least one device.

Note: If you specify `yes`, before you can use the device for ECAM messages you must also define it as an ECAM device with the `addsubsystempath` command.

`-scsiaddr dis|disable|`*domain.target.lun*`...`

Note: You must unmount the device before changing this parameter.

Note: This parameter cannot be used with the `-devpath` parameter.

Optional parameter. Cannot be specified for SCSI larger LUN devices.

Either assigns or removes the SCSI address identifier for a SCSI functional device. A unique identifier is required (that is, you cannot enter `dis` or `disable`) if the device is a functional device with the `-scsiena` parameter set to `yes`.

Entering a SCSI identifier (in the form of *domain.target.lun*) assigns the identifier to the device. *domain.target.lun* consists of three decimal numbers separated by periods, where:

– *domain* identifies one of sixteen segments of SVAA storage (0 to 15).

– *target* is the SCSI target identifier (0 to 15).

Note: The target must be 0 for all devices on a fibre-channel interface.

– *lun* is the LUN (logical unit number) for the SCSI physical device (0 to 255).

You can use ranges or lists, but not wildcards.

Note: When entering ranges, you must follow these restrictions:

– A range can span only one domain; for example, `14.0.00:14.0.99` is valid, but `14.0.00:15.0.99` is not.

– The LUN values on the beginning and ending entries of a range must contain the same number of digits, therefore you must use leading zeroes where necessary; for example, `14.0.000:14.0.255` is valid, but `14.0.0:14.0.255` is not; `5.0.05:5.0.20` is valid, but `5.0.5:5.0.20` is not.

Entering `dis` or `disable` removes the current SCSI address assignment from the device. This entry is case-sensitive.

Note: Changing the SCSI identifier for an existing functional device is a two-step process:

– First, disable access to the SCSI device:

```
altdev -subsys subsys_name -fdid fdid -scsiaddr dis
-scsiena no
```

– Then, assign the new address and re-enable access:

```
altdev -subsys subsys_name -fdid fdid
-scsiaddr domain.target.lun -scsiena yes
```

`-scsiena yes|no`

Note: You must unmount the device before changing this parameter.

Note: The `-scsiena yes` option cannot be used with the `-devpath` parameter.

Optional parameter. Cannot be specified for SCSI larger LUN devices.

Specifies whether to enable UNIX and Windows computer access to the SCSI device. If this parameter is set to `yes`, UNIX and Windows computers will be able to access the device. Entry is case-sensitive.

Note: This parameter also updates the "valid?" designation on the specified domain, target, and LUN in the SVA subsystem.

`-scsirw yes|no`

Note: You must unmount the device before changing this parameter.

Optional parameter. Cannot be specified for SCSI larger LUN devices.

Specifies whether to allow UNIX and Windows computer write access to the SCSI device. If this parameter is set to `yes`, SCSI computers will be able to write data to the device. Entry is case-sensitive.

Note: The following parameters are used for defining open PPRC pairs only. You can use only one of the following optional parameters: `-addrcpairs`, `-modifyrcpairs`, or `-deletercpairs`.

`-addrcpairs`

Optional parameter. Used to establish PPRC pairs for the specified device(s).

If you use this parameter, you must also specify the following parameters:

– `-trgssname` or `-trgserialnumber`, and

– `-trgssid` or `-trgvcu`

Note: If you specify the `-addrcpairs` parameter, you must also use either the `-trgssname` or the `-trgserialnumber` parameter, but not both.

`-trgssname` *subsys_name*

Used with the `-addrcpairs` parameter to specify the name of the secondary SVA subsystem where the secondary volume(s) exist(s). You can use this parameter only if the SVAA server has access to the secondary SVA subsystem. (The `querysubsystem` command displays all subsystems to which the local server has access.)

`-trgserialnumber` *serial_number*

Used with the `-addrcpairs` parameter to specify the serial number of the secondary SVA subsystem where the secondary volume(s) exist(s). You can use this parameter regardless of whether the SVAA server has access to the secondary SVA subsystem. (To display the serial number, use the `displaysubsystem` command against the secondary subsystem; if no SVAA is connected to the secondary subsystem, use the LOP.)

Entry is up to 12 numeric digits; entry will be automatically left-padded with zeroes if you enter fewer than 12 digits.

Note: If you specify the `-addrcpairs` parameter, you must also use either the `-trgssid` or `-trgvcu` parameter, but not both.

`-trgssid` *ssid*

Used with the `-addrcpairs` parameter to specify the subsystem ID (SSID) of the secondary SVA subsystem control unit where the secondary volume(s) exist(s).

You can use this parameter regardless of whether the local SVAA server has access to the secondary SVA subsystem. (To display SSIDs, use the `displaysubsystem` command against the secondary subsystem; if no SVAA is connected to the secondary subsystem, use the LOP.)

Entry is a hexadecimal number.

`-trgvcu` *vcu*

Used with the `-addrcpairs` parameter to specify the number of the secondary SVA subsystem virtual control unit (VCU) where the secondary volume(s) exist(s).

You can use this parameter only if the local SVAA server has access to the secondary SVA subsystem. (The `querysubsystem` command displays subsystems to which the server has access.) If the secondary SVA subsystem does not appear in the `querysubsystem` display, use the `-trgssid` parameter instead.

`-trgid` *id...*

Optional parameter. Used with the `-addrcpairs` parameter to specify the FDID (device address) of the secondary volume in a PPRC pair. If you do not use this parameter, the FDID of the secondary volume defaults to the lower

byte of the specified primary volume's FDID (for example, if the primary volume is FDID 256, the secondary volume will be FDID 56); use this parameter if you want the FDID of the secondary volume to be something different.

If you enter multiple secondary FDIDs, they will be assigned to the secondary volumes in left-to-right order. The number of PPRC pairs established is limited to the number of secondary FDIDs supplied.

Entry is a hexadecimal number; decimal representation is not supported. You can use ranges and lists, but not wildcards.

`-mode` *sync_mode*

Optional parameter. Used with the `-addrcpairs` parameter to specify the extent to which the PPRC pairs will be synchronized at the time they are established. Only one *sync_mode* can be specified. Valid entries are as follows:

– `copy`—Performs a full synchronization of the primary volume to the secondary volume. That is, all tracks on the primary volume are copied to the secondary volume.

– `nocopy`—Does not synchronize any data from the primary volume to the secondary volume. This mode is useful if the volumes have been synchronized previously and no updates have since occurred to the primary volume. It is also useful if the secondary volume will be overwritten at some time in the future, as in the case of PPRC SnapShot.

– `resync`—Re-establishes a suspended PPRC pair. Only tracks that were updated during the suspension period will be copied from the primary volume to the secondary. A PPRC pair can be suspended in any of the following ways:

  • The PPRC pair is explicitly suspended with an `alterdevice -modifyrcpairs -mode suspend` command.

  • The secondary volume is released from PPRC activity with an `alterdevice -modifyrcpairs -mode recover` command.

  • Either the primary or secondary SVA subsystem detects a communication problem between the two subsystems and suspends the PPRC pair.

  Note: The `resync` option cannot be used with a PPRC bridge pair (`-devtyp DATABRIDGE` or `STATUSBRIDGE`).

Default is `copy`.

See "Synchronization of Data Within a PPRC Pair," on page 60 for a complete explanation of the modes and states.

`-modifyrcpairs`

Used to modify the PPRC mode of existing PPRC pairs. You can use a single command to modify multiple subsystem IDs (SSIDs)/virtual control units (VCUs) within the specified SVA subsystem.

Note: If you use the `-modifyrcpairs` parameter, you must also specify the `-mode` parameter.

`-mode` *pprc_mode*

Used with the `-modifyrcpairs` parameter to set the PPRC mode which effectively suspends all PPRC activity between the PPRC pair. Only one *pprc_mode* can be specified. Valid entries are as follows:

– `recover`—Valid only when issued to a secondary volume. Releases a secondary volume from the PPRC environment in order to gain control of it on the secondary SVA subsystem. (Normally an active secondary volume cannot be accessed directly by the secondary subsystem.) To re-activate PPRC control over a released secondary volume, you must re-establish the pair with the `-addrcpairs -mode resync` option on the `alterdevice` command.

– `suspend`—Can be issued to either a primary or a secondary volume. Suspends all PPRC operations between the primary and secondary volumes within a PPRC pair. While the pair is suspended, no data is transferred to the secondary volume. The primary volume keeps track of updates made to it, and when an `-addrcpairs -mode resync` option is used on the `alterdevice` command, only the updated tracks are copied to the secondary volume.

Note: The `suspend` option cannot be used on a PPRC bridge volume (`-devtyp DATABRIDGE` or `STATUSBRIDGE`).

There is no default.

See “Synchronization of Data Within a PPRC Pair,” on page 60 for a complete explanation of the modes and states.

`-deletercpairs`

Deletes PPRC pairs for the specified devices on the specified primary SVA subsystem. Deletes all PPRC pairs that include the specified devices as primary volumes.

Examples

Example 1 (change a single device):

```
SIB> alterdevice -subsys ABC1 -fdid 0D -privlg yes -cache ena
-ckdena yes -scsiena yes
```

Updates the following characteristics for functional device 0D on subsystem ABC1: privileged ECAM, cache enabled, CKD enabled, and SCSI enabled.

Example 2 (change multiple devices by specifying FDIDs):

```
SIB> altdev -subsys ABC2 -fdid 0A:0F,19 -ckdena yes
```

Sets CKD enabled to "yes" (enabled) for the following devices on subsystem ABC2: 0A through 0F, and 19.

Example 3 (change multiple devices by specifying device paths):

```
SIB> altdev -subsys ABC2 -devpath J:,M: -name xmp1,xmp5
```

Changes the names of the two specified devices on subsystem ABC2 to xmp1 and xmp5.

Example 4 (change a SCSI address):

```
SIB> altdev -subsys ABC3 -fdid FF -scsiaddr dis -scsiena no
SIB> altdev -subsys ABC3 -fdid FF -scsiaddr 0.0.15
-scsiena yes
```

Changes the SCSI address for device FF on subsystem ABC3 in two steps: first, disables device FF on subsystem ABC3; then updates the device with a new SCSI address and enables it for SCSI access.

Example 5 (change a functional device into a SCSI larger LUN device):

```
SIB> altdev -subsys ABC2 -fdid 21 -fcapa 500G -fdidpool 100:1A0
SIB9822D: You are about to change the size of device 021. You will want to
back your data up first. Do you want to continue (y/n)? y
SIB9824I: Device (FDID 021) successfully altered with an exact size of
500.80GB.
```

Sets the capacity of FDID 21 to 500GB. Uses capacity from FDIDs 100 through 1A0 to do so.

Example 6 (change a functional device into a SCSI larger LUN device):

```
SIB> altdev -subsys ABC2 -fdid 47 -adjustfcapa +10G -noconfirm
SIB9824I: Device (FDID 047) successfully altered with an exact size of
511.12GB.
```

Increases the capacity of FDID 47 by 10GB. Does not prompt for confirmation.

Example 7 (establish PPRC pairs with a known secondary SVA subsystem—same FDIDs):

```
SIB> altdev -subsys LOCAL -fdid 21:25 -addrcpairs
-trgssname REMOTE -trgvcu 0
```

Creates PPRC pairs for FDIDs 21 through 25 on primary subsystem LOCAL. The secondary subsystem, which is accessible to the local SVAA server, is named REMOTE. The secondary volumes are on VCU 0 of the secondary subsystem. The FDIDs of the secondary volumes are the same as those of the primary volumes.

Example 8 (establish PPRC pairs with a known secondary SVA subsystem—different FDIDs):

```
SIB> altdev -subsys LOCAL -fdid 42:44 -addrcpairs
-trgssname REMOTE -trgvcu 0 -trgid 52:54
```

Creates PPRC pairs for FDIDs 42 through 44 on primary subsystem LOCAL. The secondary subsystem, which is accessible to the local SVAA server, is

named REMOTE. The secondary volumes are on VCU 0 of the secondary subsystem. The FDIDs of the secondary volumes are different (52 through 54) from those of the primary volumes.

Example 9 (establish PPRC pairs with an unknown secondary SVA subsystem):

```
SIB> altdev -subsys LOCAL -fdid 21:25 -addrcpairs
-trgserialnumber 12345 -trgssid 1200
```

Creates PPRC pairs for FDIDs 21 through 25 on primary subsystem LOCAL. The secondary subsystem, which is not accessible to the local SVAA server, has serial number 12345. The secondary volumes are on SSID 1200 of the secondary subsystem.

Example 10 (establish a PPRC pair with an unknown secondary SVA subsystem—"nocopy" mode):

```
SIB> altdev -subsys LOCAL -fdid 154 -addrcpairs -mode nocopy
-trgserialnumber 12345 -trgssid 1200
```

Creates a PPRC pair for FDID 154 on primary subsystem LOCAL. The secondary subsystem, which is not accessible to the local SVAA server, has serial number 12345. The secondary volume is on SSID 1200 of the secondary subsystem. The PPRC pair is created in "nocopy" mode, meaning no data is copied from the primary to the secondary volume when the pair is established.

Example 11 (delete PPRC pairs):

```
SIB> altdev -subsys LOCAL -fdid 21:25,135:140 -deletercpairs
```

Deletes all PPRC pairs within the FDID ranges 21 through 25, and 135 through 140 between primary subsystem LOCAL and any secondary subsystem.

Example 12 (suspend PPRC pairs):

```
SIB> altdev -subsys LOCAL -fdid 256:258 -modifyrcpairs
-mode suspend
```

Suspends PPRC operations on all PPRC pairs that include FDID 256 through 258 on primary subsystem LOCAL and any secondary subsystem.

Example 13 (recover a secondary volume from PPRC activity):

```
SIB> altdev -subsys REMOTE -fdid 56 -modifyrcpairs
-mode recover
```

Recovers the secondary volume in the PPRC pair that includes FDID 56 on secondary subsystem REMOTE. This makes the secondary volume available for active use by the secondary subsystem.

Example 14 (re-synchronize a suspended PPRC pair):

```
SIB> altdev -subsys LOCAL -fdid 256 -addrcpairs -mode resync
-trgssname REMOTE -trgvcu 0
```

Resynchronizes the PPRC pair that includes FDID 256 on primary subsystem LOCAL. The secondary subsystem is named REMOTE. The secondary volume is on VCU 0 of the secondary subsystem.

See Also

```
attndevice
definedevice
deletedevice
displaydevice
displaysubsystem
querysubsystem
```

alteriointerface

Name `alteriointerface` — change the name of an I/O interface, or add or delete PPRC paths for the specified interface ID.

Abbreviation `altioint`

Synopsis

```
alteriointerface -subsys subsys_name -ifid c.i...
[-name interface_name...]|
```

for PPRC (peer-to-peer remote copy):

Note: Only certain SVAA platforms and SVA subsystem configurations support the use of PPRC commands and functions. Contact StorageTek Software Support for details.

```
[-addpath -vcu vcu_number...|-ssid ssid_number...
-trgssname subsys_name|-trgserialnumber serial_number
-trgvcussid vcu_ssid_number...|-trgvcu vcu_number...|
-trgssid ssid_number... [-linkdest link_dest_number...]]|
[-deletepath -vcu vcu_number...|-ssid ssid_number...
-trgvcu vcu_number...|-trgssid ssid_number...
[-trgssname subsys_name|-trgserialnumber serial_number]]
```

Description This command allows you to make any of the following changes to an enabled I/O interface:

- Change the I/O interface name.
- Add a PPRC path between a primary and a secondary SVA subsystem (use the `-addpath` parameter and its supporting parameters).
- Delete a PPRC path between a primary and a secondary SVA subsystem (use the `-deletepath` parameter and its supporting parameters).

To change any other characteristic of an I/O interface, you must use the LOP.

Note: The I/O interface must be enabled in order for you to use this command.

Security For non-PPRC-related data: Requires write access to a privileged ECAM device.

For PPRC-related data: If a PPRC administrator group has been defined, requires PPRC administrator privileges; otherwise, requires SVAA administrator privileges.

Options `-subsys` *subsys_name*

Specifies the name of the SVA subsystem with which the I/O interface is associated. You can specify only one SVA subsystem name. Entry is case-sensitive, and wildcards are not supported.

`-ifid` *c.i...*

Specifies the channel interface identifiers, where:

– *c* identifies one of the two clusters (0 or 1) in the SVA subsystem.

– *i* is the interface ID (A through P) associated with a specific cluster.

You can use ranges or lists, but not wildcards.

Note: You must use one and only one of the following optional parameters: `-name`, `-addpath`, or `-deletepath`.

`-name` *interface_name...*

Note: This parameter cannot be used with the `-addpath` nor `-deletepath` parameter.

Optional parameter. Specifies the names you want to assign respectively to each of the I/O interfaces.

Entry is one to eight characters, or a null string. The following characters are accepted, even as the first one: a to z, A to Z, 0 to 9, $, @, #, -, _, +, &, ., and /. A null string, entered as `-name ''` or `-name ""`, sets the name to all blanks.

Entry is case-sensitive. You can use ranges and lists. (See the `sibadmin` command reference for rules about entering string ranges.)

Note: The following parameters are used for configuring PPRC paths only. You can use either the `-addpath` or the `-deletepath` parameter, but not both.

`-addpath`

Note: This parameter cannot be used with the `-name` nor `-deletepath` parameter.

Note: At least one FDID must have been defined for this SVA subsystem in order for you to use this parameter.

Optional parameter. Used to establish PPRC paths for the I/O interfaces specified by the `-ifid` parameter. You can assign paths to up to four (in the case of V960 or earlier subsystems) or 16 (in the case of V2X) I/O interfaces, with each interface having up to four paths each. You can use a single command to define paths between only one primary and secondary SVA subsystem pair.

If you use this parameter, you must also specify the following parameters:

– `-vcu` or `-ssid`, and

– `-trgvcu`, `-trgssid`, or `-trgvcussid`, and

– `-trgssname` or `-trgserialnumber`

Note: If you specify the `-addpath` parameter, you must also use either the `-vcu` or the `-ssid` parameter, but not both.

`-vcu` *vcu_number...*

Used with the `-addpath` parameter to specify the virtual control unit (VCU) on the primary SVA subsystem that will have a path established.

`-ssid` *ssid...*

Used with the `-addpath` parameter to specify the subsystem ID (SSID) on the primary SVA subsystem that will have a path established. This entry corresponds to a VCU within the SVA subsystem.

Entry is a hexadecimal number.

Note: If you specify the `-addpath` parameter, you must also use one and only one of the following parameters: `-trgvcussid`, `-trgvcu`, `-trgssid`.

`-trgvcussid` *vcu.ssid...*

Used with the `-addpath` parameter to specify the virtual control unit (VCU) number and subsystem ID (SSID), separated by a period, of the secondary SVA subsystem for the new path. You can use this parameter regardless of whether the local SVAA server has access to the secondary SVA subsystem.

You can enter up to four VCU/SSID pairs. *vcu* is a decimal number. *ssid* is a hexadecimal number. Ranges and wildcards are not supported.

`-trgvcu` *vcu_number...*

Used with the `-addpath` parameter to specify the virtual control unit (VCU) number on the secondary SVA subsystem for the new path.

You can use this parameter only if the local SVAA server has access to the secondary SVA subsystem. (The `querysubsystem` command displays all subsystems to which the local server has access.) If the secondary SVA subsystem does not appear in the `querysubsystem` display, use the `-trgvcussid` parameter instead.

You can enter ranges and lists, but not wildcards.

`-trgssid` *ssid...*

Used with the `-addpath` parameter to specify the subsystem ID (SSID) of the secondary SVA subsystem for the new path.

You can use this parameter only if the local SVAA server has access to the secondary SVA subsystem. (The `querysubsystem` command displays all subsystems to which the local server has access.) If the secondary SVA subsystem does not appear in the `querysubsystem` display, use the `-trgvcussid` parameter instead.

Entry is a hexadecimal number. You can enter ranges and lists, but not wildcards.

Note: If you specify the `-addpath` parameter, you must also use either the `-trgssname` or the `-trgserialnumber` parameter, but not both.

`-trgssname` *subsys_name*

Used with the `-addpath` parameter to specify the name of the secondary SVA subsystem for the new path. You can use this parameter only if the local SVAA server has access to the secondary SVA subsystem. (The `querysubsystem` command displays all subsystems to which the local server has access.)

`-trgserialnumber` *serial_number*

Used with the `-addpath` parameter to specify the serial number of the secondary SVA subsystem for the new path. You can use this parameter regardless of whether the local SVAA server has access to the secondary SVA subsystem. (To display the serial number, use the `displaysubsystem` command against the secondary subsystem; if no SVAA is connected to the secondary subsystem, use the LOP.)

Entry is up to 12 numeric digits; entry will be automatically left-padded with zeroes if you enter fewer than 12 digits.

`-linkdest` *link_dest_number...*

Optional parameter. Used with the `-addpath` parameter to specify the destination logical address.

– If the connection is direct, enter `0`.

– If the connection is going through an ESCON director, enter the assigned number.

Entry is a one-byte hexadecimal number.

`-deletepath`

Optional parameter. Used to delete PPRC paths for the specified I/O interface and virtual control units (VCUs)/subsystem IDs (SSIDs). You can use a single command to delete paths for multiple VCUs/SSIDs within the specified SVA subsystem. Only active paths are deleted.

If you use this parameter, you must also specify the following parameters:

– `-vcu` or `-ssid`, and

– `-trgvcu` or `-trgssid`

Note: If you specify the `-deletepath` parameter, you must use either the `-vcu` or `-ssid` parameter, but not both.

`-vcu` *vcu_number...*

Used with the `-deletepath` parameter to specify the virtual control unit (VCU) number of the primary SVA subsystem from which you want to delete a path.

You can enter ranges, lists, and wildcards. An entry of * indicates that you want to delete all paths for the specified I/O interfaces.

`-ssid` *ssid...*

Used with the `-deletepath` parameter to specify the subsystem ID (SSID) of the primary SVA subsystem from which you want to delete a path.

You can enter ranges, lists, and wildcards. An entry of * indicates that you want to delete all paths for the specified I/O interfaces.

Note: If you specify the `-deletepath` parameter, you must use either the `-trgvcu` or the `-trgssid` parameter, but not both.

`-trgvcu` *vcu_number...*

Used with the `-deletepath` parameter to specify the virtual control unit (VCU) number on the secondary SVA subsystem from which you want to delete a path. You can use this parameter only if the local SVAA server has access to the secondary SVA subsystem. (The `querysubsystem` command displays all subsystems to which the local server has access.)

You can enter ranges and lists, but not wildcards.

`-trgssid` *ssid...*

Used with the `-deletepath` parameter to specify the subsystem ID (SSID) of the secondary SVA subsystem from which you want to delete a path. You can use this parameter regardless of whether the local SVAA server has access to the secondary SVA subsystem. (To display SSIDs, use the `displaysubsystem` command against the secondary subsystem; if no SVAA is connected to the secondary subsystem, use the LOP.)

Entry is a hexadecimal number. You can enter ranges and lists, but not wildcards.

Note: If you specify the `-deletepath` parameter, you can also use either the `-trgssname` or the `-trgserialnumber` parameter, but not both.

`-trgssname` *subsys_name*

Optionally used with the `-deletepath` parameter to specify the secondary SVA subsystem from which you want to delete a path. You can use this parameter only if the local SVAA server has access to the secondary SVA subsystem. (The `querysubsystem` command displays all subsystems to which the local server has access.)

`-trgserialnumber` *serial_number*

Optionally used with the `-deletepath` parameter to specify the serial number of the secondary SVA subsystem from which you want to delete a path. You can use this parameter regardless of whether the local SVAA server has access to the secondary SVA subsystem. (To display the serial number, use the `displaysubsystem` command against the secondary subsystem; if no SVAA is connected to the secondary subsystem, use the LOP.)

Entry is up to 12 numeric digits; entry will be automatically left-padded with zeroes if you enter fewer than 12 digits.

Examples

Example 1:

```
SIB> alteriointerface -subsys ABC2 -ifid 0.A:0.H
-name CH1:CH8
```

Changes the interface names CH1 through CH8 to 0.A through 0.H, respectively, on subsystem ABC2.

Example 2:

```
SIB> altioint -subsys ABC1 -ifid 0.H,0.I,0.L
-name CHAN0,CHAN1,CHAN2
```

Assigns the interface names CHAN0, CHAN1, and CHAN2 to 0.H, 0.I, and 0.L, respectively, on subsystem ABC1.

Example 3 (define a PPRC path to a known SVA subsystem):

```
SIB> altioint -subsys LOCAL -ifid 0.I -addpath -vcu 0
-trgssname REMOTE -trgvcu 0
```

Establishes a PPRC path from subsystem LOCAL, interface ID 0.I, to subsystem REMOTE which is accessible to the local SVAA server. The path goes between VCU 0 on both subsystems.

Example 4 (define multiple PPRC paths to a known SVA subsystem):

```
SIB> altioint -subsys LOCAL -ifid 0.A,0.C -addpath -vcu 0:3
-trgssname REMOTE -trgvcu 0:3
```

Establishes PPRC paths from subsystem LOCAL, interface IDs 0.A and 0.C, to the subsystem named REMOTE which is accessible to the local SVAA server. The paths use VCUs 0 through 3 on both the primary and secondary subsystems.

Example 5 (define a PPRC path to a known SVA subsystem):

```
SIB> altioint -subsys LOCAL -ifid 0.I -addpath -ssid 4100
-trgssname REMOTE -trgssid 5100
```

Establishes a PPRC path from subsystem LOCAL, interface ID 0.I, to subsystem REMOTE which is accessible to the local SVAA server. The path goes between SSID 4100 on the primary subsystem and SSID 5100 on the secondary subsystem.

Example 6 (define a PPRC path to an unknown SVA subsystem):

```
SIB> altioint -subsys LOCAL -ifid 0.I -addpath -vcu 2
-trgserialnumber 12345 -trgvcussid 2.5200
```

Establishes a PPRC path from subsystem LOCAL, interface ID 0.I, to the subsystem with serial number 12345 which is not accessible to the local SVAA server. The path goes between VCU 2 on the primary subsystem and VCU 2 with SSID 5200 on the secondary subsystem.

Example 7 (define a PPRC path to an unknown SVA subsystem—through an ESCON director):

```
SIB> altioint -subsys LOCAL -ifid 0.I -addpath -ssid 4003
-trgserialnumber 54321 -trgvcussid 3.5300 -linkdest 1
```

Establishes a PPRC path from subsystem LOCAL, interface ID 0.I, to the subsystem with serial number 54321 which is not accessible to the local SVAA server. The path goes between SSID 4003 on the primary subsystem and VCU 3 with SSID 5300 on the secondary subsystem, through an ESCON director.

Example 8 (delete PPRC paths to any secondary subsystem):

```
SIB> alteriointerface -subsys LOCAL -ifid 0.I
-deletepath -ssid 4100 -trgssid 5100
```

Deletes all PPRC paths between primary subsystem LOCAL and any secondary subsystem that are associated with interface ID 0.I and SSID 4100 on the primary subsystem and SSID 5100 on the secondary subsystem.

Example 9 (delete PPRC paths to any secondary subsystem):

```
SIB> altioint -subsys LOCAL -ifid 0.I -deletepath -vcu 0
-trgvcu 0
```

Deletes all PPRC paths between subsystem LOCAL and any secondary subsystem that are associated with interface ID 0.I and VCU 0 on the primary subsystem and VCU 0 on the secondary subsystem.

Example 10 (delete PPRC paths to a specific known secondary SVA subsystem):

```
SIB> altioint -subsys LOCAL -ifid 0.I -deletepath -vcu 1
-trgssname REMOTE -trgvcu 1
```

Deletes all PPRC paths between primary subsystem LOCAL and secondary subsystem REMOTE that are associated with interface ID 0.I and VCU 1 on the primary subsystem and VCU 1 on the secondary subsystem. The secondary subsystem is accessible to the local SVAA server.

Example 11 (delete PPRC paths to a specific unknown secondary SVA subsystem):

```
SIB> altioint -subsys LOCAL -ifid 0.I -deletepath -ssid 4100
-trgserialnumber 12345 -trgssid 5100
```

Deletes all PPRC paths between primary subsystem LOCAL and the secondary subsystem with serial number 12345 that are associated with interface ID 0.I and SSID 4100 on the primary subsystem and SSID 5100 on the secondary subsystem. The secondary subsystem is not accessible to the local SVAA server.

See Also

```
displayiointerface
displaysubsystem
querysubsystem
```

altersubsystem

Name `altersubsystem` — change SVA subsystem characteristics.

Abbreviation `altsubsys`

Synopsis

```
altersubsystem -subsys subsys_name [-arysi 5+2|13+2]
[-newname new_subsystem_name] [-scsisim enable|disable]
[-site site_name]
[[-ssid0 ssid0] [-ssid1 ssid1] [-ssid2 ssid2] [-ssid3 ssid3]]|
[-vcussid vcu.ssid[, vcu.ssid]]|[-ssidbase base_ssid]
```

Description This command changes the characteristics of an SVA subsystem. It has no defaults. At least one of the optional parameters must be used.

Note: Any parameter not specified retains its current setting.

With this command you can:

- Change the current default array size for an SVA subsystem: 5+2 or 13+2 (the default as set initially at the LOP or as changed by the most recent `altersubsystem` command).

  Note: For V960 and V2X subsystems, the only valid array size is 13+2.

- Change an existing SVA subsystem name
- Enable service information messages (SIMs) for SCSI devices (relevant only if your installation includes SVAA running on S/390 hosts)
- Change your site name
- Assign up to four (in the case of V960 or earlier subsystems) or 16 (in the case of V2X) subsystem IDs (SSIDs) for emulated 3990 controllers (relevant only if your installation includes S/390 hosts)

  Note: You cannot redefine an SSID that includes part or all of a PPRC (peer-to-peer remote copy) pair.

- Assign subsystem IDs (SSIDs) to virtual control units (VCUs).

Security Requires write access to a privileged ECAM device.

Options `-subsys` *subsys_name*

Specifies the existing name for the SVA subsystem. You can specify only one SVA subsystem name. Entry is case-sensitive, and wildcards are not supported.

`-arysi 5+2|13+2`

Optional parameter. Specifies the number of drive modules to be used in forming a new array (user data drives + redundancy data drives). This does not affect existing arrays.

Valid entries are `5+2` or `13+2`.

Note: `5+2` is not valid for V960 nor V2X subsystems.

`-newname` *new_subsystem_name*

Optional parameter. Specifies the new name you want to assign to the SVA subsystem. Only one name can be specified. Entry is case-sensitive, and wildcards are not supported.

Entry is one to eight characters. The following characters are accepted, even as the first one: a to z, A to Z, 0 to 9, $, @, #, -, _, +, &, ., and /.

Note: Within your site, each SVA subsystem name must be unique.

`-scsisim enable|disable`

Optional parameter. Enables or disables service information messages (SIMs) to SCSI devices.

Entry is case-sensitive. Abbreviations are `ena` and `dis`.

Note: Use `enable` only when the SVA subsystem is attached to an S/390 host (SVAA converts SIMs to meaningful messages only if the messages are sent to an SVAA server running on an S/390 host; messages to SCSI devices are displayed in hexadecimal and thus not easily interpreted.)

`-site` *site_name*

Optional parameter. Identifies your company, organization, or location. Use this name when you call StorageTek Software Support.

Entry is one to eight characters, or a null string. The following characters are accepted, even as the first one: a to z, A to Z, 0 to 9, $, @, #, -, _, +, &, ., and /. A null string, entered as `''` or `""`, sets the name to all blanks.

Entry is case-sensitive, and wildcards are not supported.

Note: You can use only one of the following optional parameters: any or all of the `-ssid`*n* parameters (`-ssid0` through `-ssid3`), `-vcussid`, or `-ssidbase`.

`-ssid0` *ssid0,* `-ssid1` *ssid1,* `-ssid2` *ssid2,* `-ssid3` *ssid3*

Note: Although currently valid for all SVA subsystem models, these parameters will no longer be supported at some point in the future. Additionally, since this set of parameters only supports the first four VCUs, it is recommended that you use either the `-vcussid` or `-ssidbase` parameter instead.

Note: These parameters cannot be used with the `-vcussid` nor the `-ssidbase` parameter.

Optional parameters. Using these parameters, you can assign up to four subsystem identifiers (SSIDs) per SVA subsystem. Each SSID is used to identify a 3390 virtual control unit (VCU) emulated by the SVA subsystem. If your installation includes SVAA running on S/390 hosts, you must assign

a unique SSID to each logical 3990 in your installation. For example, a unique SSID is required for each set of 256 virtual devices.

Note: SSIDs are not used by UNIX nor Windows computers. Therefore, if your installation includes SVAA running on UNIX and/or Windows computers only, you can assign any value you want to each SSID. If, however, your installation includes S/390 hosts, you must make sure that the SSIDs you enter here match what are defined on the S/390 hosts.

ssid0 through *ssid3* are unique, four-digit (hexadecimal) subsystem identifiers, ranging from 0001 to FFFF (on S/390 hosts). Decimal representation and wildcards are not supported.

Note: You cannot redefine an SSID that includes part or all of a PPRC (peer-to-peer remote copy) pair.

`-vcussid` *vcu*`.`*ssid*`[,` *vcu.ssid*`]`

Note: This parameter cannot be used with any of the `-ssid`*n* parameters (`-ssid0` through `-ssid3`) nor with the `-ssidbase` parameter.

Optional parameter. Used to explicitly assign a subsystem ID (SSID) to a particular 3390-emulated virtual control unit (VCU) within the specified SVA subsystem. Each VCU contains 256 FDIDs. You can specify up to 16 VCU/SSID pairs. Only the specified VCU/SSID pairs are assigned.

vcu is a decimal number; *ssid* is a hexadecimal number (for example: 0.00F0, 12.00EC, 5.00E5). Wildcards and ranges are not supported.

`-ssidbase` *base_ssid*

Note: This parameter cannot be used with any of the `-ssid`*n* parameters (`-ssid0` through `-ssid3`) nor with the `-vcussid` parameter.

Optional parameter. Used to implicitly assign subsystem IDs (SSIDs) to 3390-emulated virtual control units (VCUs) within the specified SVA subsystem. Specifies the starting SSID for the first 3390-emulated VCU (VCU 0) within the SVA subsystem. The remaining VCUs will be assigned SSID values in ascending hexadecimal sequence.

Entry is a single hexadecimal value; decimal representation is not supported. Wildcards, lists, and ranges are not supported. The value specified for *base_ssid* must not allow the last SSID to exceed FFFF. For example, on an SVA subsystem that supports 16 SSIDs, FFF0 is the highest valid *base_ssid* value.

Examples

Example 1:

```
SIB> altsubsys -subsys ABC1 -newname ABC001 -site NY01
-scsisim dis
```

Updates the name, site name, and SCSI SIM status for the subsystem previously named ABC1.

Example 2:

```
SIB> altersubsystem -subsys ABC2 -arysi 13+2
```

Changes the default array size for subsystem ABC2.

Example 3:

```
SIB> altsubsys -subsys ABC2 -ssid0 0010 -ssid1 0011
-ssid2 0012 -ssid3 0013
```

Updates four SSIDs for subsystem ABC2.

Example 4 (explicitly assign VCU/SSID pairs):

```
SIB> altsubsys -subsys ABC2 -vcussid 10.0C 11.1F
```

Assigns SSID 0C to VCU 10 and SSID 1F to VCU 11.

Example 5 (assign SSIDs to all VCUs in an SVA subsystem**):**

```
SIB> altsubsys -subsys ABC2 -ssidbase A1
```

Assigns SSID A1 to VCU 0 in subsystem ABC2. The remaining VCUs in the subsystem are assigned the next available sequential SSIDs.

See Also `displaysubsystem`
`querysubsystem`

attndevice

Name `attndevice` — initiate a state-change interrupt for a CKD device.

Abbreviation `adev`

Synopsis `attndevice -subsys` *subsys_name* `-fdid` *fdid...*

Description This command initiates a state-change interrupt for specific CKD devices. State-change interrupts are sent to the interface in each path group and all interfaces without path groups formed that are associated with a device. If the host operating system misses a "device end" from the SVA subsystem and experiences a hang condition, you can try to free the condition by issuing this command.

Note: This command applies only to devices attached to ESCON channels. Therefore, if you issue it against a device that is available only on a SCSI bus, the command is ignored. If the device is attached to both an ESCON channel and a SCSI bus, the command is processed on the ESCON channel and ignored on the SCSI bus.

Security Requires write access to a privileged ECAM device.

Options `-subsys` *subsys_name*

Specifies the name of the SVA subsystem with which the functional device is associated. You can specify only one SVA subsystem name. Entry is case-sensitive and wildcards are not supported.

`-fdid` *fdid...*

Specifies the subsystem FDID (functional device ID) for the CKD device. Entry is one to three hexadecimal digits (0 to 3FF for V960 and earlier subsystems; 0 to FFF for V2X subsystems). Decimal representation is not supported. You can use ranges and lists, but not wildcards.

Examples

Example 1

```
SIB> attndevice -subsys ABC2 -fdid 10:3F,30F
```

Initiates state-change interrupts for CKD functional devices 10 through 3F, and 30F on subsystem ABC2.

Example 2

```
SIB> adev -subsys ABC2 -fdid 1A
```

Initiates a state-change interrupt for CKD functional device 1A on subsystem ABC2.

See Also `displaydevice`

definedevice

Name `definedevice` — define a functional or SCSI larger LUN device.

Abbreviation `defdev`

Synopsis

options for all device types:

```
definedevice -subsys subsys_name -fdid fdid...
-devtyp device_type [-cache ena|enable|dis|disable]
[-ckdena yes|no] [-ckdrw yes|no] [-name device_name...]
[-privlg yes|no] [-scsiaddr dis|disable|domain.target.lun...]
[-scsiena yes|no] [-scsirw yes|no]
```

option for SCSI functional devices only:

```
[-scsiblks entry]
```

options for SCSI larger LUN devices only:

```
-fcapa minimum_size [-fdidpool fdid...]
```

options for PPRC (peer-to-peer remote copy):

Note: Only certain SVAA platforms and SVA subsystem configurations support the use of PPRC commands and functions. Contact StorageTek Software Support for details.

```
[-addrcpairs
-trgssname subsys_name |-trgserialnumber serial_number
-trgssid ssid|-trgvcu vcu [-trgid id...] [-mode sync_mode]]
```

Description This command defines a functional or SCSI larger LUN device and its characteristics. You can define up to 1024 functional devices in a V960 or earlier subsystem (0 to 3FF), 4096 in a V2X (0 to FFF)—fewer than that if you define any 3390-9 or SCSI-B devices. (If you define only 3390-9 or SCSI-B devices, you can define only one-third as many devices in an SVA subsystem—about 340 for V960 or earlier, 1365 for V2X (0 to 554). With a mixture of 3390-9 and SCSI-B devices with other devices, the maximum number lies between 340 and 1024 for V960 and earlier, and between 1365 and 4096 for V2X, depending on the mix. If you try to define more devices than the SVA has room for in its mapping tables, an error message is displayed.)

With this command you can define the functional devices to emulate SCSI devices and different IBM devices (3380 and 3390 device types), all within the same SVA subsystem.

This command specifies all the SVA subsystem data required to configure one or more devices. Physical storage devices referred to as drive modules are completely distinct and different entities which may be manipulated in a limited way by the `startdrain` and `formarray` commands.

Note: If you are using SVA Path on the Windows computer, any devices you create with the `definedevice` command must be made available to SVA Path in

order to be under its control. For details, see both of the following sections in the *SVA Path Windows User's Guide*: "Installing SVA Path on Windows" and "Dynamic Device Detection."

Security

For non-PPRC-related data: Requires write access to a privileged ECAM device.

For PPRC-related data: If a PPRC administrator group has been defined, requires PPRC administrator privileges; otherwise, requires SVAA administrator privileges.

Options

`-subsys` *subsys_name*

Specifies the name of the SVA subsystem with which the device is associated. You can specify only one SVA subsystem name. Entry is case-sensitive, and wildcards are not supported.

`-fdid` *fdid...*

Specifies the subsystem FDID (functional device ID) for the device. Entry is one to three hexadecimal digits (0 to 3FF for V960 and earlier subsystems; 0 to FFF for V2X subsystems). Decimal representation is not supported. You can use ranges or lists, but not wildcards.

Note: In the case of SCSI larger LUN devices, this is the parent (index 0) device.

`-devtyp` *device_type*

Specifies the device type and model descriptor of the functional or SCSI larger LUN device to be emulated. Entry is four to ten alphanumeric characters, not case-sensitive.

Valid entries are as follows:

```
3380J   3380-J CKD device
3380K   3380-K CKD device
3380KE  3380-KE CKD device
33901   3390-1 CKD device
33902   3390-2 CKD device
33903   3390-3 CKD device
33909   3390-9 CKD device
SCSI    SCSI larger LUN device; the system will create
        a SCSI-A larger LUN device.
SCSIA   3390-3 with SCSI format
SCSIB   3390-9 with SCSI format
DATABRIDGE    Power PPRC data bridge volume. Implies a
              SCSI-B device. Cannot be used with
              -fdidpool nor -fcapa.
STATUSBRIDGE  Power PPRC status bridge volume. Implies a
              SCSI-A device. Cannot be used with
               -fdidpool nor -fcapa.
```

To define a SCSI functional device, you must specify either `SCSIA` or `SCSIB`.

To define a SCSI larger LUN device, you must specify `SCSI`, `SCSIA`, or `SCSIB`. If you specify `SCSI` or `SCSIA`, the system will create a SCSI-A larger

LUN device; if you specify SCSIB, the system will create a SCSI-B larger LUN device.

To define a PPRC data bridge volume, you must specify either DATABRIDGE or STATUSBRIDGE.

Note: The -scsiblks parameter can be used only if -devtyp is SCSI, SCSIA, or SCSIB.

-fcapa *minimum_size*

Note: This parameter cannot be used for a PPRC bridge volume (-devtyp DATABRIDGE or STATUSBRIDGE).

This parameter is valid, and required, only for SCSI larger LUN devices. Specifies the requested minimum device size.

Entry is a decimal integer (1 to 65535) followed by M for megabytes or G for gigabytes (examples: 500G, 800g, 10000m). Entry is not case-sensitive.

-cache ena|enable|dis|disable

Optional parameter. Enables or disables caching for a CKD device. Entry is case-sensitive. With cache disabled, data for this device that is in the subsystem cache is immediately written to back-end storage.

Note: Setting -cache to disable actually has no effect on the SVA subsystem. SVA architecture prohibits caching from being disabled.

Default is enable.

-ckdena yes|no

Optional parameter. Specifies whether to enable S/390 host access to the CKD functional device. If this parameter is set to yes, S/390 hosts will be able to access the device. Entry is case-sensitive.

Default is yes for CKD devices and PPRC bridge devices (-devtype set to DATABRIDGE or STATUSBRIDGE), no for SCSI devices.

Note: You must set this parameter to yes for all devices that are part of a PPRC pair or PPRC bridge pair.

-ckdrw yes|no

Optional parameter. Specifies whether to allow S/390 host write access to the CKD functional device. If this parameter is set to yes, S/390 hosts will be able to write data to the device. Entry is case-sensitive.

Default is yes for CKD devices and PPRC bridge devices (-devtype set to DATABRIDGE or STATUSBRIDGE), no for SCSI devices.

Note: You must set this parameter to yes for all devices that are part of a PPRC pair or PPRC bridge pair.

-fdidpool *fdid...*

Note: This parameter cannot be used for a PPRC bridge volume (-devtyp DATABRIDGE or STATUSBRIDGE).

Optional parameter. Valid only for SCSI larger LUN devices. Specifies the pool of FDIDs to be used to define the device(s).

Note: The FDID pool cannot include any FDIDs that have already been defined as functional or SCSI larger LUN devices.

The request fails if the total capacity of the devices in the pool is less than the requested capacity. If, however, you specify more devices than are actually needed to fulfill the request, the extra ones are ignored. For example, if you specify four SCSI-B (8.5GB capacity) devices to fulfill a request for a 24GB larger LUN device, the fourth device will not be used.

Entry is one to three hexadecimal digits (0 to 3FF for V960 and earlier subsystems; 0 to FFF for V2X subsystems). Decimal representation is not supported. You can use ranges or lists, but not wildcards.

`-name` *device_name*

Optional parameter. Specifies the name you want to assign to each device identified by the `-fdid` parameter. You can use ranges or lists. (See the `sibadmin` command reference for rules about entering string ranges.)

Entry is one to eight characters, or a null string. The following characters are accepted, even as the first one: a to z, A to Z, 0 to 9, $, @, #, -, _, +, &, ., and /. A null string, entered as `-name ''` or `-name ""`, sets the name to all blanks.

`-privlg yes|no`

Optional parameter. Specifies whether the device is eligible for use as the transmission path for privileged ECAM messages to the SVA subsystem.

You can specify `no` for most devices—to limit the number of privileged ECAM devices—but you must specify `yes` for at least one device in the SVA subsystem.

Default is `no`.

Note: If you specify `yes`, before you can use the device for ECAM messages you must also define it as an ECAM device with the `addsubsystempath` command.

`-scsiaddr dis|disable|`*domain.target.lun* **. . .**

Assigns the SCSI identifier for the device. A unique identifier is required (that is, you cannot enter `dis` or `disable`) if the device is either:

– a SCSI larger LUN device, or

– a functional device with the `-scsiena` parameter set to `yes`.

Entering `dis` or `disable` removes the current SCSI address assignment from the device. This entry is case-sensitive.

domain.target.lun consists of three decimal numbers separated by periods, where:

– *domain* identifies one of two logical domain addresses (0 to 15).

– *target* is the SCSI target identifier (0 to 15).

Note: The target must be 0 for all devices on a fibre-channel interface.

– *lun* is the LUN (logical unit number) for the SCSI physical device (0 to 255).

You can use ranges or lists, but not wildcards.

Note: When entering ranges, you must follow these restrictions:

– A range can span only one domain; for example, `14.0.00:14.0.99` is valid, but `14.0.00:15.0.99` is not.

– The LUN values on the beginning and ending entries of a range must contain the same number of digits, therefore you must use leading zeroes where necessary; for example, `14.0.000:14.0.255` is valid, but `14.0.0:14.0.255` is not; `5.0.05:5.0.20` is valid, but `5.0.5:5.0.20` is not.

`-scsiblks` *entry*

Optional parameter. Defines the logical block size for a SCSI device. This parameter can be used only if `-scsiena` is `yes` and `-devtyp` is set to `SCSI`, `SCSIA`, or `SCSIB` (that is, it is not valid for CKD devices).

To achieve the best system performance, you should base your entry on the type of system that will be using the device(s) you are creating. Only one *entry* can be specified. Valid entries and the resulting block sizes are as follows (entries are not case-sensitive):

```
Entry    Block Size
512      512 bytes
2048     2048 bytes
4096     4096 bytes
8192     8192 bytes
16384    16384 bytes
aix      4096 bytes
hp-ux    4096 bytes
solaris  4096 bytes
unix     4096 bytes
win      4096 bytes
```

If you do not specify an *entry*, SVAA will choose the block size best suited for the type of system on which it is running (for example, if SVAA is running on a Solaris or Windows-based system it will choose 4096 bytes).

`-scsiena yes|no`

Optional parameter. Specifies whether to enable UNIX and Windows computer access to the SCSI functional device. If this parameter is set to `yes`, UNIX and Windows computers will be able to access the device. Entry is case-sensitive.

Note: This parameter also updates the “valid?” designation on the specified domain, target, and LUN in the SVA subsystem.

Default is `yes` for SCSI devices and PPRC bridge devices (`-devtype` set to `DATABRIDGE` or `STATUSBRIDGE`), `no` for CKD devices.

`-scsirw yes|no`

Optional parameter. Specifies whether to allow UNIX and Windows computer write access to the SCSI functional device. If this parameter is set to `yes`, SCSI computers will be able to write data to the device. Entry is case-sensitive.

Default is `yes` for SCSI devices and PPRC bridge devices (`-devtype` set to `DATABRIDGE` or `STATUSBRIDGE`), `no` for CKD devices.

Note: The following parameters are used for defining open PPRC pairs only.

`-addrcpairs`

Optional parameter. Used to establish PPRC pairs for the specified device(s).

If you use this parameter, you must also specify the following parameters:

– `-trgssname` or `-trgserialnumber`, and

– `-trgssid` or `-trgvcu`

Note: If you specify the `-addrcpairs` parameter, you must also use either the `-trgssname` or the `-trgserialnumber` parameter, but not both.

`-trgssname` *subsys_name*

Used with the `-addrcpairs` parameter to specify the name of the secondary SVA subsystem where the secondary volume(s) exist(s). You can use this parameter only if the SVAA server has access to the secondary SVA subsystem. (The `querysubsystem` command displays all subsystems to which the local server has access.)

`-trgserialnumber` *serial_number*

Used with the `-addrcpairs` parameter to specify the serial number of the secondary SVA subsystem where the secondary volume(s) exist(s). You can use this parameter regardless of whether the SVAA server has access to the secondary SVA subsystem. (To display the serial number, use the `displaysubsystem` command against the secondary subsystem; if no SVAA is connected to the secondary subsystem, use the LOP.)

Entry is up to 12 numeric digits; entry will be automatically left-padded with zeroes if you enter fewer than 12 digits.

Note: If you specify the `-addrcpairs` parameter, you must also use either the `-trgssid` or `-trgvcu` parameter, but not both.

`-trgssid` *ssid*

Used with the `-addrcpairs` parameter to specify the subsystem ID (SSID) of the secondary SVA subsystem control unit where the secondary volume(s) exist(s).

You can use this parameter regardless of whether the local SVAA server has access to the secondary SVA subsystem. (To display SSIDs, use the

displaysubsystem command against the secondary subsystem; if no SVAA is connected to the secondary subsystem, use the LOP.)

Entry is a hexadecimal number.

-trgvcu *vcu*

Used with the -addrcpairs parameter to specify the number of the secondary SVA subsystem virtual control unit (VCU) where the secondary volume(s) exist(s).

You can use this parameter only if the local SVAA server has access to the secondary SVA subsystem. (The querysubsystem command displays subsystems to which the server has access.) If the secondary SVA subsystem does not appear in the querysubsystem display, use the -trgssid parameter instead.

-trgid *id...*

Optional parameter. Used with the -addrcpairs parameter to specify the FDID (device address) of the secondary volume in a PPRC pair. If you do not use this parameter, the FDID of the secondary volume defaults to the lower byte of the specified primary volume's FDID (for example, if the primary volume is FDID 256, the secondary volume will be FDID 56); use this parameter if you want the FDID of the secondary volume to be something different.

If you enter multiple secondary FDIDs, they will be assigned to the secondary volumes in left-to-right order. The number of PPRC pairs established is limited to the number of secondary FDIDs supplied.

Entry is a hexadecimal number; decimal representation is not supported. You can use ranges and lists, but not wildcards.

-mode *sync_mode*

Optional parameter. Used with the -addrcpairs parameter to specify the extent to which the PPRC pairs will be synchronized at the time they are established. Only one *sync_mode* can be specified. Valid entries are as follows:

– copy—Performs a full synchronization of the primary volume to the secondary volume. That is, all tracks on the primary volume are copied to the secondary volume.

– nocopy—Does not synchronize any data from the primary volume to the secondary volume. This mode is useful if the volumes have been synchronized previously and no updates have since occurred to the primary volume. It is also useful if the secondary volume will be overwritten at some time in the future, as in the case of PPRC SnapShot.

Default is copy.

See "Synchronization of Data Within a PPRC Pair," on page 60 for a complete explanation of the modes and states.

Examples

Example 1 (SCSI device):

```
SIB> defdev -subsys ABC1 -fdid 8A -devtyp SCSIA -scsiaddr
0.0.10 -scsiena yes -scsirw yes
```

Defines device 8A on subsystem ABC1 as a SCSI-A device. The device has SCSI identifier 0.0.10, is enabled to the computer, and has write access enabled.

Example 2 (SCSI larger LUN device):

```
SIB> defdev -subsys ABC2 -fdid 21 -scsiblks 512 -devtyp scsi -scsiaddr 1.0.7
-fcapa 450g
SIB9824I Device (FDID 021) successfully defined with an exact size of
451.12 GB.
```

Defines device 21 on subsystem ABC2 as a SCSI-A larger LUN device. The device has SCSI identifier 1.0.7 a 512 block size, and a capacity of 451.12GB.

Example 3 (multiple SCSI larger LUN devices):

```
SIB> defdev -subsys ABC1 -fdid d0,d4 -fdidpool d1:d3,d5:ef -devtyp scsi
-scsiaddr 1.0.2,1.0.23 -fcapa 12g
```

Defines larger LUN devices d0 and d4 on subsystem ABC1. They are SCSI-A devices with 12GB capacity, and devices from the specified FDID pool (d1–d3, d5–ef) are used to create them.

Note: d0 and d4 must be excluded from the FDID pool because they are the IDs of the larger LUN devices.

Example 4 (CKD device):

```
SIB> definedevice -subsys ABC1 -fdid 80 -devtyp 3380J -name COMM02 -cache ena
-privlg yes
```

Defines functional device 80 on subsystem ABC1 as a 3380J device. The device is assigned the name COMM02, has cache enabled, and is a privileged ECAM device.

Example 5 (define a data bridge device):

```
SIB> defdev -subsys LOCAL -fdid 48 -devtyp databridge
-name BRIDGE1 -ckdena yes -ckdrw yes
```

Defines functional device 48 on subsystem LOCAL as a data bridge device. The device, which is defined as a SCSI-B device, is assigned the name BRIDGE1 and has CKD read/write access set to “yes”.

Example 6 (define a status bridge device):

```
SIB> defdev -subsys LOCAL -fdid 49 -devtyp STATUSBRIDGE
-name STATUS1 -ckdena yes -ckdrw yes
```

Defines functional device 49 on subsystem LOCAL as a status bridge device. The device, which is defined as a SCSI-A device, is assigned the name STATUS1 and has CKD read/write access set to “yes”.

Example 7 (define a device and create a PPRC pair with a known secondary SVA subsystem):

```
SIB> defdev -subsys LOCAL -fdid 15a -devtyp SCSIB
-scsiaddr 4.0.18 -name PPRC1 -ckdena yes -ckdrw yes
-addrcpairs -mode nocopy -trgssname REMOTE -trgvcu 3
-trgid 6a
```

Defines a new SCSI-B device, FDID 15a, on subsystem LOCAL. The device is the primary volume in a PPRC pair. The secondary subsystem, which is accessible to the local SVAA server, is named REMOTE. The secondary volume is on VCU 3 of the secondary subsystem. The FDID of the secondary volume is different (6a) from that of the primary volume (15a). The PPRC pair is created in "nocopy" mode, meaning no data is copied from the primary to the secondary volume when the pair is established.

Example 8 (define a device and create a PPRC pair with an unknown secondary SVA subsystem):

```
SIB> defdev -subsys LOCAL -fdid 35d -devtyp SCSIB
-scsiaddr 4.0.22 -name PPRC4 -ckdena yes -ckdrw yes
-addrcpairs -trgserialnumber 12345 -trgssid 5100
```

Defines a new SCSI-B device, FDID 35d, on subsystem LOCAL. The device is the primary volume in a PPRC pair. The secondary subsystem, which is not accessible to the local SVAA server, has serial number 12345. The secondary volume is on SSID 5100 of the secondary subsystem. The FDID of the secondary volume is 5d, the lower byte of the primary volume's FDID.

See Also

```
alterdevice
definedevice
displaydevice
```

deletedevice

Name `deletedevice` — delete functional or SCSI larger LUN device definition.

Abbreviation `deldev`

Synopsis `deletedevice -subsys` *subsys_name* `-fdid` *fdid*`... [-force] [-noconfirm]`

Description Deletes the definition of one or more SVA subsystem functional or SCSI larger LUN devices. Before you issue this command, you should perform all the following steps for each device to be deleted:

- Unmount the device from all UNIX hosts.
- Verify that the device is not being accessed by any Windows computers.
- On all UNIX or Windows computers where the device is defined, use the computer's disk administration tool to delete the device.
- Vary the device offline to all S/390 hosts.
- If the device is an ECAM device, use the `removesubsystempath` command to remove the device path to the SVA subsystem.
- It is strongly recommended that you remove all user data and partitioning information from an FDID before deleting it. Deleting an FDID essentially has the same effect as unplugging a hard drive from a running system. Therefore, you should perform the same steps before deleting an FDID as you would before removing a hard drive.

  For example, FDID B00 is seen by Windows Disk Management as `PhysicalDrive9`. All partitions, dynamic disk volumes, or striped and spanned volumes located completely or partially on `PhysicalDrive9` should be removed before deleting FDID B00.

  Failure to follow these recommendations may have adverse consequences.

  In the example above, assuming there was a basic disk primary partition on `PhysicalDrive9`, accessible through drive letter `I:`, if you deleted FDID B00 before deleting drive letter `I:`, the drive letter would still be accessible through Windows Explorer. Users might, therefore, be able to read and write files to drive letter `I:`, possibly resulting in loss of data.

Only index 0 devices can be deleted with this command. If the device is a SCSI larger LUN device, then all of its "child" FDIDs are deleted along with it.

Caution: All data on the device is effectively erased by this command. Therefore, before you delete the device, you must back up any data you want to preserve, or move it to another device.

To prevent inadvertent loss of data, the command uses the following precautions:

- Prompts you to confirm the deletion. You can bypass the confirmation prompt by issuing the command with the `-noconfirm` parameter.
- Fails if the specified device contains data. You can force the deletion by issuing the command with the `-force` parameter.
- Fails if the specified device is part of an active PPRC (peer-to-peer remote copy) pair or is a PPRC bridge volume. You must first use the `alterdevice -deletercpairs` command to delete the pair definition, then use the `deletedevice` command to delete the device.

Security

Requires write access to a privileged ECAM device.

Options

`-subsys` *subsys_name*

Specifies the name of the SVA subsystem with which the device is associated. You can specify only one SVA subsystem name. Entry is case-sensitive, and wildcards are not supported.

`-fdid` *fdid...*

Specifies the subsystem FDID (functional device ID) for the device. Entry is one to three hexadecimal digits (0 to 3FF for V960 and earlier subsystems; 0 to FFF for V2X subsystems). Decimal representation is not supported. You can use ranges or lists, but not wildcards.

`-force`

Optional parameter. Indicates that the specified devices are to be deleted, even if they contain user data. If you do not use this parameter, any devices containing user data will not be deleted.

Note: This parameter has no effect if the specified device is part of a PPRC pair or is a PPRC bridge volume.

`-noconfirm`

Optional parameter. Indicates that you do not want a confirmation prompt displayed after you issue the command.

Examples

Example 1 (deletion of devices without data):

```
SIB> deldev -subsys ABC2 -fdid 1A:1F,30B
SIB9833D: About to delete device(s): 1A:1F, 30B
Do you really want to delete the device(s) (y/n)? y
```

Deletes functional devices 1A through 1F, and 30B on subsystem ABC2.

Example 2 (deletion of a device without data—no confirmation):

```
SIB> deldev -subsys ABC2 -fdid 1A -noconfirm
```

Deletes functional device 1A without prompting for confirmation.

Example 3 (failed deletion of a device with data):

```
SIB> deldev -subsys ABC2 -fdid 21
SIB9833D: About to delete device(s): 21
```

```
Do you really want to delete the device(s) (y/n)? y
SIB9634W: (WARNING) FDID 21: Rejected by the hardware: Functional device has
back-end storage defined
```

Attempts to delete device 21 on subsystem ABC2, but the device is not deleted because it contains data and the `-force` parameter was not used.

Example 4 (forced deletion of devices with data):

```
SIB> deldev -subsys ABC2 -fdid 15,17 -force
SIB9833D: About to delete device(s): 15, 17
Do you really want to delete the device(s) (y/n)? y
```

Deletes functional devices 15 and 17 on subsystem ABC2, even if they contain data.

Example 5 (forced deletion of a device—no confirmation):

```
SIB> deldev -subsys ABC2 -fdid 2C -force -noconfirm
```

Deletes functional device 2C on subsystem ABC2 without prompting for confirmation, even if it contains data.

Example 6 (deletion cancelled by user):

```
SIB> deldev -subsys ABC1 -fdid 1A
SIB9833D: About to delete device(s): 1A
Do you really want to delete the device(s) (y/n)? n
SIB9637I: Operation aborted. No device deleted.
```

Request is to delete device 1A on subsystem ABC1, but the device is not deleted because the user does not confirm the operation at the prompt.

Example 7 (failed deletion of a device included in a SCSI larger LUN device):

```
SIB> deldev -subsys ABC2 -fdid 28
SIB9833D: About to delete device(s): 28
Do you really want to delete the device(s) (y/n)? y
SIB9638W WARNING: Can not delete device 028, it is part of a Larger Device
(SCSI address 12.0.1), parent FDID 021. Use the AlterDevice command if you
want to shrink the device.
SIB9836E ERROR: No devices have been deleted on subsystem ABC2.
```

Request is to delete device 28 on subsystem ABC2, but the device is not deleted because it is part of larger LUN device 21.

Example 8 (failed forced deletion of a device included in a SCSI larger LUN device):

```
SIB> deldev -subsys ABC2 -fdid 84 -force
SIB9638W WARNING: Can not delete device 084, it is part of a Larger Device
(SCSI address 12.0.1), parent FDID 021. Use the AlterDevice command if you
want to shrink the device.
SIB9836E ERROR: No devices have been deleted on subsystem ABC2.
```

Request is to delete device 84 on subsystem ABC2, but the device is not deleted because it is part of larger LUN device 21. Device is not deleted, even though the `-force` parameter is used.

Example 9 (failed deletion of a device included in an active PPRC pair):

```
SIB> deldev -subsys ABC1 -fdid 20 -force -noconfirm
SIB7704E Rejected by Hardware: completion code 0x09, reason code 0x21,
```

```
message no. 0x03 - Request denied - PPRC is active on volume or VCU.
SIB9836E ERROR: No devices have been deleted on subsystem ABC1.
```

Request is to delete device 20 on subsystem ABC1, but the device is not deleted because it is part of an active PPRC pair.

See Also `definedevice`
`displaydevice`

displayarray

Name `displayarray` — produce an array report.

Abbreviation `dary`

Synopsis `displayarray -subsys` *`subsys_name...`* `[-aryid` *`array_id`*`[,`*`array_id`*`]]`

Description This command produces a report showing the characteristics of one or more arrays, including:

- Current SVA subsystem name
- Array size (5+2 or 13+2)

  Note: For V960 and V2X subsystems, the only valid array size is 13+2.
- Partition
- Configuration size
- Current status of the array, such as initializations and drains in progress.

Security All users allowed.

Options `-subsys` *`subsys_name...`*

Specifies the names of the SVA subsystem with which the arrays are associated. Entry is case-sensitive. You can use ranges and lists. (See the `sibadmin` command reference for rules about entering string ranges.)

`-aryid` *`array_id`*`[,`*`array_id`*`]`

Optional parameter. Identifies the arrays to be included in the report. Each *array_id* is an integer from 0 to 7. You can use lists and wildcards, but not ranges.

If you do not enter a value for this parameter, the report shows all arrays for the specified SVA subsystem(s).

If you have specified more than one SVA subsystem name, the command attempts to display the specified arrays on all the specified subsystems.

Examples **Example 1:**

```
SIB> displayarray -subsys ABC1
```

Displays characteristics of all arrays for subsystem ABC1.

```
SVAA Server: Server1         Date/Time: 10-28-2001 14:40:41

Subsystem Name:      ABC1
Subsystem Model:     V960
Frame Serial Number: 310000001041
Site Name:           STK-A1
Site Location No.:   1

              FORMATION  SPARE(S) TWO DRIVE GROUP DRAIN ----------DEVICE STATUSES-----------
ARRAY CONFIG IN PROGRESS  NEEDED   FAILURE    STATUS    INITIALIZING DRAINING RECONSTRUCTING
----- ------ ----------- -------- --------- ----------- ------------ -------- --------------
  0    13+2      NO        NO       NO         N.A.         NO         N.A.        N.A.
  1    13+2      NO        NO       NO         N.A.         NO         N.A.        N.A.
  2    13+2      NO        NO       NO         N.A.         NO         N.A.        N.A.
  3    13+2      NO        NO       NO         N.A.         NO         N.A.        N.A.
  4    13+2      NO        NO       NO         N.A.         NO         N.A.        N.A.
  5    13+2      NO        NO       NO         N.A.         NO         N.A.        N.A.
  6    13+2      NO        NO       NO         N.A.         NO         N.A.        N.A.
  7    13+2      NO        NO       NO         N.A.         NO         N.A.        N.A.
```

See Also displaydevice
formarray
startdrain

displaydevice

Name `displaydevice` — display current functional device status.

Abbreviation `ddev`

Synopsis `displaydevice -subsys` *`subsys_name...`* `-fdid` *`fdid...`* `[-detail] [-functional|-logical] [-pprcinfo] [-subsysfcapa]`

Description This command displays the current status of one or more devices. By default, only logical devices are displayed. You can optionally specify to display the functional devices that make up a SCSI larger LUN.

Note: Capacity is shown in gigabytes (GB) rounded to the nearest tenth (one position after the decimal); therefore, if the capacity of a device is less than 50MB, it will be displayed as 0.0GB.

Security All users allowed.

Options `-subsys` *`subsys_name...`*

Specifies the name of the SVA subsystem with which the devices are associated. Entry is case-sensitive. You can use ranges and lists. (See the `sibadmin` command reference for rules about entering string ranges.)

`-fdid` *`fdid...`*

Specifies the functional identifiers (FDIDs) for the devices to be displayed. Entry is one to three hexadecimal digits (0 to 3FF for V960 and earlier subsystems; 0 to FFF for V2X subsystems). Decimal representation is not supported. You can use ranges, lists, and wildcards.

If you have specified more than one SVA subsystem name, the command attempts to display the specified FDIDs on all the specified subsystems.

`-detail`

Optional parameter. Specifies to display additional detail for individual devices. This detail consists of each device's functional capacity, physical capacity used, logical block size, logical blocks per track, and compression ratio, as well as an average compression ratio for all devices displayed. Also, in the case of logical devices, the detail includes a list of the devices that make up each SCSI larger LUN device. There are no values for this parameter.

Note: If in addition to using this parameter, you also use the * wildcard on the `-fdid` parameter to indicate you want to display all devices for the specified SVA subsystem, the detail display will also include the compression ratio for the entire SVA subsystem.

Note: You can use either the `-functional` or the `-logical` parameter, but not both. The default is `-logical`.

`-functional`

Optional parameter. Specifies to display functional devices only. There are no values for this parameter.

`-logical`

Optional parameter. Specifies to display logical devices only. There are no values for this parameter.

This is the default display if neither `-functional` nor `-logical` is specified.

Note: If the `-detail` parameter is also specified, the display will include a "Device Composition" section showing all child FDIDs that make up each SCSI larger LUN parent.

`-pprcinfo`

Optional parameter. Displays PPRC (peer-to-peer remote copy) status information for those devices that are part of a PPRC pair or are PPRC bridge volumes.

Note: If client tracing is enabled (see `startclienttrace` for details) when you specify this parameter, a large amount of I/O activity will be generated.

`-subsysfcapa`

Optional parameter. Displays the capacity of the entire subsystem. There are no values for this parameter.

Examples

Example 1 (default output—logical view):

```
SIB> displaydevice -subsys ABC2 -fdid 40:49
```

Displays the status of logical devices 40 through 49 on subsystem ABC2.

```
SVAA Server: Server1 Date/Time: 02-16-2002  23:12:47

Subsystem Name:      ABC2
Subsystem Model:     V960
Frame Serial Number: 3100013200123
Site Name:           STK-A1
Site Location No.:   1
Logical Devices:     6
Device Information:
===================
        FDID             CAPACITY    ----------------SCSI--------------       ---CKD--- PRVLG
FDID    NAME      TYPE     (GB)      DOMAIN TARGET LUN ENBLD R/W ACCESS CACHE ENBLD R/W ECAM     PPRC STATE
----- --------   ------ ------------ ------ ------ --- ----- --- ------ ----- ----- --- ----- ----------------
  040 CORC0      SCSI           2.46   0       0     0   Y    Y    2      Y     Y    Y    Y
  041 CORC1      SCSI           1.23   0       0     1   Y    N    1      Y     Y    Y    N
  044 CORC4      SCSI           2.46   0       0     2   Y    N    0      Y     Y    Y    N
  046 CORC6      SCSI           7.38   0       0     3   Y    N    0      Y     Y    Y    N
  049 CORC7      SCSI          29.52   4       0     1   Y    N    2      Y     Y    Y    N
```

Example 2 (default output—logical view—with detail):

```
SIB> ddev -subsys ABC2 -fdid 40:49 -detail
```

Displays the status of logical devices 40 through 49 on subsystem ABC2, including detail.

```
SVAA Server: Server1                Date/Time: 05-09-2002 11:23:19

Subsystem Name:      ABC2
Subsystem Model:     V960
Frame Serial Number: 3100013200123
Site Name:           STK-A1
Site Location No.:   1
Logical Devices:     6

Device Information:
===================
        FDID              CAPACITY    ----------------SCSI--------------       ---CKD--- PRVLG
FDID    NAME      TYPE      (GB)      DOMAIN TARGET LUN ENBLD R/W ACCESS CACHE ENBLD R/W ECAM     PPRC STATE
----- --------   ------ ------------ ------ ------ --- ----- --- ------ ----- ----- --- ----- ----------------
  040 CORC0      SCSI           2.46   0       0     0   Y    Y    2      Y     Y    Y    Y
  041 CORC1      SCSI           1.23   0       0     1   Y    N    1      Y     Y    Y    N
  044 CORC4      SCSI           2.46   0       0     2   Y    N    0      Y     Y    Y    N
  046 CORC6      SCSI           7.38   0       0     3   Y    N    0      Y     Y    Y    N
  049 CORC7      SCSI          29.52   4       0     1   Y    N    2      Y     Y    Y    N

Device Composition:
===================
FDID  CHILD FDIDS
----- ------------------------------------------------------------------------------------
040   -
041   -
044   -
046   -
049   050, 051, 052, 053

Device Capacity:
================
         FUNCTIONAL CAP(GB)    ---PHYSICAL CAP USED (GB)----         PHYSICAL CAP  COMP
FDID   AVAILABLE      STORED     SHARED    UNIQUE     TOTAL    CYLS  LBLKS  LBPT  RATIO
----- ------------ ------------ --------- --------- --------- ------ ------ ----- -----
  040         2.46         0.00      0.00      0.00      0.00   6678   4096    12   0.0
  041         1.23         0.00      0.00      0.00      0.00   3339   4096    12   0.0
  044         2.46         0.00      0.00      0.00      0.00   6678   4096    12   0.0
  046         7.38         3.05      0.00      0.50      0.50  10017   4096    12   6.1
  049        29.52         0.67      0.00      0.32      0.32  56763   4096    12   2.1
===== ============ ============ ========= ========= =========                      =====
TOTAL        43.05         3.72      0.00      0.82      0.82                        1.6  (device level)
```

Example 3 (default output—logical view—with subsystem capacity):

```
SIB> ddev -subsys ABC2 -fdid 40 -subsysfcapa
```

Displays the status of logical device 40 on subsystem ABC2, including total subsystem capacity.

```
SVAA Server: Server1          Date/Time: 03-16-2001 14:05:22

Subsystem Name:      ABC2
Subsystem Model:     V960
Frame Serial Number: 310000001041
Site Name:           STK-A1
Site Location No.:   1
Logical Devices:     6
Functional Capacity: 463.6 (GB)

Device Information:
===================
        FDID              CAPACITY    ----------------SCSI--------------       ---CKD--- PRVLG
FDID    NAME      TYPE      (GB)      DOMAIN TARGET LUN ENBLD R/W ACCESS CACHE ENBLD R/W ECAM     PPRC STATE
----- --------   ------ ------------ ------ ------ --- ----- --- ------ ----- ----- --- ----- ----------------
  040 CORC0      SCSI           2.46   0       0     0   Y    Y    2      Y     Y    Y    Y
```

Example 4 (functional view):

```
SIB> ddev -subsys ABC2 -fdid 40:55 -functional
```

Displays the status of functional devices 40 through 55 on subsystem ABC2.

```
SVAA Server: Server1                Date/Time: 03-16-2001 14:10:15

Subsystem Name:       ABC2
Subsystem Model:      V960
Frame Serial Number: 3100013200123
Site Name:            STK-A1
Site Location No.:    1
Logical Devices:      6

Device Information:
===================
        FDID              CAPACITY    ----------------SCSI--------------                ---CKD--- PRVLG
FDID    NAME      TYPE      (GB)      DOMAIN TARGET LUN INDEX  ENBLD R/W ACCESS CACHE ENBLD R/W ECAM    PPRC STATE
----- --------   ------ ------------ ------ ------ --- -----  ----- --- ------ ----- ----- --- ----- ----------------
  040 CORC0      SCSI           2.46   0       0     0   0      Y     Y    2      Y     Y     Y    Y
  041 CORC1      SCSI           1.23   0       0     1   0      Y     N    1      Y     N     N    N
  044 CORC4      SCSI           2.46   0       0     2   0      N     N    0      Y     Y     Y    N
  046 CORC6      SCSI           7.38   0       0     3   0      N     N    0      Y     Y     Y    N
  049 CORC7      SCSI          29.52   4       0     1   0      Y     Y    2      Y     Y     Y    N
  050 CORC7      SCSI           7.38   4       0     1   1      Y     Y    2      Y     Y     Y    N
  051 CORC7      SCSI           7.38   4       0     1   2      Y     Y    2      Y     Y     Y    N
  052 CORC7      SCSI           7.38   4       0     1   3      Y     Y    2      Y     Y     Y    N
```

Example 5 (functional view with detail):

```
SIB> ddev -subsys ABC2 -fdid 40:52 -functional -detail
```

Displays the status of functional devices 40 through 52 on subsystem ABC2.

```
SVAA Server: Server1                 Date/Time: 05-07-2002 12:23:02

Subsystem Name:       ABC2
Subsystem Model:      V960
Frame Serial Number: 3100013200123
Site Name:            STK-A1
Site Location No.:    1
Logical Devices:      6

Device Information:
===================
        FDID              CAPACITY    ----------------SCSI--------------                ---CKD--- PRVLG
FDID    NAME      TYPE      (GB)      DOMAIN TARGET LUN INDEX  ENBLD R/W ACCESS CACHE ENBLD R/W ECAM    PPRC STATE
----- --------   ------ ------------ ------ ------ --- -----  ----- --- ------ ----- ----- --- ----- ----------------
  040 CORC0      SCSI           2.46   0       0     0   0      Y     Y    2      Y     Y     Y    Y
  041 CORC1      SCSI           1.23   0       0     1   0      Y     N    1      Y     N     N    N
  044 CORC4      SCSI           2.46   0       0     2   0      Y     N    0      Y     Y     Y    N
  046 CORC6      SCSI           7.38   0       0     3   0      Y     N    0      Y     Y     Y    N
  049 CORC7      SCSI          29.52   4       0     1   0      Y     Y    2      Y     Y     Y    N
  050 CORC7      SCSI           7.38   4       0     1   1      Y     Y    2      Y     Y     Y    N
  051 CORC7      SCSI           7.38   4       0     1   2      Y     Y    2      Y     Y     Y    N
  052 CORC7      SCSI           7.38   4       0     1   3      Y     Y    2      Y     Y     Y    N

Device Capacity:
================
        FUNCTIONAL CAP(GB)    ---PHYSICAL CAP USED (GB)----        PHYSICAL CAP  COMP
FDID   AVAILABLE      STORED     SHARED    UNIQUE     TOTAL    CYLS  LBLKS  LBPT  RATIO
----- ------------ ------------ --------- --------- --------- ------ ------ ----- -----
  040         2.46         0.00      0.00      0.00      0.00   6678   4096    12   0.0
  041         1.23         0.00      0.00      0.00      0.00   3339   4096    12   0.0
  044         2.46         0.00      0.00      0.00      0.00   6678   4096    12   0.0
  046         7.38         3.05      0.00      0.50      0.50  10017   4096    12   6.1
  049        29.52         0.67      0.00      0.32      0.32  56763   4096    12   2.1
  050         7.38         0.10      0.00      0.20      0.20  10017   4096    12   2.0
  051         7.38         0.27      0.00      0.40      0.40  10017   4096    12   1.8
  052         7.38         0.30      0.00      0.40      0.40  10017   4096    12   2.5
===== ============ ============ ========= ========= =========                      =====
TOTAL        65.19         4.39      0.00      1.82      1.82                       1.8  (device level)
```

Example 6 (PPRC status information):

```
SIB> ddev -subsys ABC2 -fdid 343:346 -pprcinfo
```

Displays PPRC status information for FDIDs 343 through 346.

```
SVAA Server: Server1          Date/Time: 12-16-2001 14:05:22

Subsystem Name:      ABC2
Subsystem Model:     V960
Frame Serial Number: 310013200123
Site Name:           STK-A1
Site Location No.:   1
Logical Devices:     6

Device Information:
===================
        FDID             CAPACITY    ----------------SCSI--------------       ---CKD--- PRVLG
FDID    NAME      TYPE     (GB)      DOMAIN TARGET LUN ENBLD R/W ACCESS CACHE ENBLD R/W ECAM   PPRC STATE
------ --------  ------ ------------ ------ ------ --- ----- --- ------ ----- ----- --- -----  --------------
  343 DATA3      DATA            8.5   -       -     -   N    N    0      Y     Y    Y    N    data primary
  344 STATUS4    STATUS          2.8   -       -     -   N    N    0      Y     Y    Y    N    status simplex
  345 DATA5      DATA            8.5   -       -     -   N    N    0      Y     Y    Y    N    data simplex
  346 STATUS6    STATUS          2.8   -       -     -   N    N    0      Y     Y    Y    N    status simplex


PPRC Information
================
        -------PRIMARY-------- ------SECONDARY------- FIRST CYL LAST CYL
FDID    ID  SSID  SERIAL NUM   ID  SSID  SERIAL NUM   NOT SYNC  NOT SYNC     STATE DESC          QUERY FLAGS
-----   --- ----- ----------- ---- ---- ------------ --------- --------- ------------------   -------------------
 343    43  4200    13200123   43  4300   78539727     FFFF      FFFF    full copy mode      prim,path act
 344    44  4200    13200123   44  4300   78539727     FFFF      FFFF
 345    45  4200    13200123   45  4300   78539727     FFFF      FFFF
 346    46  4200    13200123   46  4300   78539727     FFFF      FFFF
```

See Also alterdevice
attndevice
deletedevice

displaydrivemodule

Name `displaydrivemodule` — produce a drive module report.

Abbreviation `ddmod`

Synopsis
`displaydrivemodule -subsys` *subsys_name...*
`-aryid` *array_id*`[,`*array_id*`]|-dmod` *u.t.s...*`| -status` *status*

Description This command produces a report showing the characteristics of one or more drive modules.

Note: Unavailable drives are not displayed unless you explicitly request them by specifying either `-status all` or `-status unavailable`.

Note: Capacity is shown in gigabytes (GB) rounded to the nearest tenth (one position after the decimal); therefore, if the capacity of a device is less than 50MB, it will be displayed as 0.0GB.

Security All users allowed.

Options `-subsys` *subsys_name...*

Specifies the names of the SVA subsystems with which the drive modules are associated. Entry is case-sensitive. You can use ranges and lists. (See the `sibadmin` command reference for rules about entering string ranges.)

Note: You must use one and only one of the following parameters: `-aryid`, `-dmod`, or `-status`.

`-aryid` *array_id*`[,`*array_id*`]`

Identifies the arrays that contain the drive modules to be included in the report. Each *array_id* is an integer from 0 to 7. You can use lists and wildcards, but not ranges.

If you have specified more than one SVA subsystem name, the command attempts to display the characteristics of the specified arrays for each of the specified subsystems.

`-dmod` *u.t.s...*

Specifies the drive modules to be included in the report. Each *u.t.s* (three digits separated by periods) identifies a specific drive module, where:

– *u* identifies the disk array unit (0 to 3)

– *t* identifies the tray (0 to 3)

– *s* identifies the slot (0 to 7)

You can use ranges, lists, and wildcards.

If you have specified more than one SVA subsystem name, the command attempts to display the specified drive modules on all the specified subsystems.

`-status` *status*

Specifies the status of the drive modules to be displayed. You may specify one of the following.

– `all`—Includes drive modules in all SVA subsystem partitions.

– `production`—Includes drive modules in the Production partition only.

– `spare`—Includes drive modules in the Spares and/or MAT partition(s) only.

– `unavailable`—Includes drive modules in the Unavailable partition only.

Examples

Example 1:

```
SIB> displaydrivemodule -subsys ABC1 -dmod 0.0.?
```

Displays characteristics of all drive modules in unit and tray 0.0 on subsystem ABC1.

```
SVAA Server: Server1          Date/Time: 03-28-2001 14:40:41

Subsystem Name:      ABC1
Subsystem Model:     9500
Frame Serial Number: 310000001041
Site Name:           STK-A1
Site Location No.:   1
Capacity:            4636.0 (GB)

-DEVICE LOCATION-  SERIAL  DEVICE       CAPACITY  ----------------DEVICE TYPE-----------------
UNIT  TRAY  SLOT   NUMBER  STATUS ARRAY   (GB)         CLASS               SUB-CLASS
----- ----- ----- -------- ------ ----- --------- --------------- ----------------------------
  0     0     2   00033725 P.A      1         4.5 SCSI-unstrapped 1.6 GB, ESDI, SCSI-strapped
  0     0     4   00036729 P.A      6         4.5 SCSI-unstrapped 1.6 GB, ESDI, SCSI-strapped
  0     0     5   00036569 M.A      -         4.5 SCSI-unstrapped 1.6 GB, ESDI, SCSI-strapped
  0     0     6   00456212 S.A      -         4.5 SCSI-unstrapped 1.6 GB, ESDI, SCSI-strapped
```

Example 2:

```
SIB> ddmod -subsys ABC1 -aryid 0,3
```

Displays characteristics of arrays 0 and 3 on subsystem ABC1.

```
SVAA Server: Server1          Date/Time: 10-28-2000 14:47:18

Subsystem Name:      ABC1
Subsystem Model:     9500
Frame Serial Number: 310000001041
Site Name:           STK-A1
Site Location No.:   1
Capacity:            46360.0 (GB)

-DEVICE LOCATION-  SERIAL  DEVICE       CAPACITY  ----------------DEVICE TYPE-----------------
UNIT  TRAY  SLOT   NUMBER  STATUS ARRAY   (GB)         CLASS               SUB-CLASS
----- ----- ----- -------- ------ ----- --------- --------------- ----------------------------
  0     1     4   01125675 P.A      0        18.0 SCSI-unstrapped 18 GB, ESDI, SCSI-strapped
  0     0     7   01166162 P.A      3        18.0 SCSI-unstrapped 18 GB, ESDI, SCSI-strapped
```

Example 3:

```
SIB> ddmod -subsys ABC1 -status spare
```

Displays all drive modules in the Spares and MAT partitions on subsystem ABC1.

```
SVAA Server: Server1          Date/Time: 03-28-2001 14:47:18

Subsystem Name:      ABC1
Subsystem Model:     9500
Frame Serial Number: 310000001041
Site Name:           STK-A1
Site Location No.:   1
Capacity:            46360.0 (GB)

-DEVICE LOCATION-  SERIAL  DEVICE       CAPACITY ----------------DEVICE TYPE-----------------
UNIT  TRAY  SLOT   NUMBER  STATUS ARRAY   (GB)        CLASS               SUB-CLASS
----- ----- ----- --------- ------ ----- --------- --------------- ---------------------------
  0     0     6    00456212 S.A    -          4.5 SCSI-unstrapped 1.6 GB, ESDI, SCSI-strapped
```

See Also `displayarray`

displayiointerface

Name `displayiointerface` — display the current I/O interface status.

Abbreviation `dioint`

Synopsis `displayiointerface -subsys` *subsys_name*`...` `-ifid` *c.i*`...` `[-pprcinfo] [-type` *type*`]`

Description This command displays the current I/O interface status, including PPRC (peer-to-peer remote copy) path information, if applicable; up to four PPRC paths can be displayed for each I/O interface.

Note: By default, the command displays fibre, parallel, SCSI, and serial (ESCON) interfaces only. Uninstalled I/O interfaces are not displayed unless you explicitly request them by specifying either `-type all` or `-type uninstalled`.

Security All users allowed.

Options `-subsys` *subsys_name*`...`

Specifies the names of the SVA subsystems for which the I/O interfaces are installed. Entry is case-sensitive. You can use ranges and lists. (See the `sibadmin` command reference for rules about entering string ranges.)

`-ifid` *c.i*`...`

Specifies one or more (up to 32) I/O interfaces to be displayed.

– *c* identifies one of the two clusters (0 or 1) in the SVA subsystem.

– *i* is the interface ID (A through P) associated with a specific cluster.

You can use ranges, lists, and wildcards.

If you have specified more than one SVA subsystem name, the command attempts to display the specified I/O interfaces for all the specified subsystems.

`-pprcinfo`

Optional parameter. Displays PPRC path information for the specified I/O interface(s).

`-type` *type*

Optional parameter. Specifies the type of I/O interface(s) to be displayed.

You may specify one of the following:

– `fibre`—Displays fibre channels.

– `parallel`—Displays parallel channels (not available for V2X subsystems).

- `scsi`—Displays SCSI I/O interfaces (not available for V2X subsystems).
- `serial`—Displays serial (ESCON) channels.
- `uninstalled`—Displays all uninstalled I/O interfaces of any type.
- `all`—Displays all I/O interface types. Entries are grouped by type.

Examples

Example 1 (default output):

```
SIB> displayiointerface -subsys ABC1 -ifid 0.A:0.G
```

Displays the current status of I/O interfaces 0.A through 0.G on subsystem ABC1.

```
SVAA Server: Server1          Date/Time: 10-28-2000 14:39:51

Subsystem Name:      ABC1
Subsystem Model:     9500
Frame Serial Number: 310001320012
Site Name:           SVAA
Site Location No.:   97802
No. of Interfaces:   32

IFID NAME     STATUS   TYPE     SPEED
---- -------- -------- -------- -----
0.A           ENABLED  FIBRE    100.0
0.B  0.B      ENABLED  FIBRE    100.0
0.C           ENABLED  FIBRE    100.0
0.D           DISABLED FIBRE    100.0
0.E           ENABLED  FIBRE    100.0
0.F           DISABLED FIBRE    100.0
```

Example 2 (default output):

```
SIB> dioint -subsys ABC1,ABC2 -ifid 0.*
```

Displays current status of all interfaces on cluster 0 for subsystems ABC1 and ABC2.

```
SVAA Server: Server1           Date/Time: 1-28-2001 14:39:51

Subsystem Name:      ABC1
Subsystem Model:     9500
Frame Serial Number: 310001320012
Site Name:           SVAA
Site Location No.:   97802
No. of Interfaces:   32

IFID NAME     STATUS   TYPE     SPEED
---- -------- -------- -------- -----
0.A           ENABLED  FIBRE    100.0
0.B  0.B      ENABLED  FIBRE    100.0
0.C           ENABLED  FIBRE    100.0
0.D           DISABLED FIBRE    100.0
0.E           ENABLED  FIBRE    100.0
0.F           DISABLED FIBRE    100.0
```

```
Subsystem Name:      ABC2
Subsystem Model:     V960
Frame Serial Number: 310000001042
Site Name:           STK-A1
Site Location No.:   1
No. of Interfaces:   8

IFID NAME     STATUS   TYPE     SPEED
---- -------- -------- -------- -----
0.G           ENABLED  FIBRE    100.0
0.H           ENABLED  SERIAL    20.0
0.P           ENABLED  SERIAL    20.0
```

Example 3 (serial I/O interfaces only):

```
SIB> dioint -subsys ABC2 -ifid * -type serial
```

Displays all serial I/O interfaces on subsystem ABC2.

```
SVAA Server: Server1           Date/Time: 03-28-2001 14:39:51

Subsystem Name:      ABC1
Subsystem Model:     9500
Frame Serial Number: 310001320012
Site Name:           SVAA
Site Location No.:   97802
No. of Interfaces:   32

IFID   NAME     STATUS      TYPE     SPEED  BASE  RANGE  BFDID
---- -------- ----------- --------   -----  ----  -----  -----
0.H           ENABLED   SERIAL      20.0   0C0   0064   000
0.P           ENABLED   SERIAL      20.0   0C0   0064   000
```

Example 4 (All I/O interface types, grouped by type):

```
SIB> dioint -subsys ABC3 -ifid 0.A:0.G -type all
```

Displays I/O interfaces 0.A through 0.G on subsystem ABC3.

```
SVAA Server: BHDev02 Date/Time: 03-28-2001 14:39:51

Subsystem Name:      ABC5
Subsystem Model:     V2X
Frame Serial Number: 310001325712
Site Name:           SVAA
Site Location No.:   97802
No. of Interfaces:   32

Parallel channel(s):
====================
IFID   NAME      STATUS       TYPE   SPEED RANGE BASE BFDID       LINK USAGE
---- -------- ------------- -------- ----- ----- ---- ----- ---------------------

Serial channel(s):
==================
IFID   NAME      STATUS       TYPE   SPEED RANGE BASE BFDID       LINK USAGE
---- -------- ------------- -------- ----- ----- ---- ----- ---------------------
0.B              ENABLED    SERIAL    20.0     1  0C0   000 PPRC LOCAL PRIMARY
0.C              ENABLED    SERIAL    20.0   128  0C0   000
0.D              ENABLED    SERIAL    20.0     1  0C0   000

SCSI interface(s):
==================
IFID   NAME      STATUS       TYPE   SPEED RANGE BASE BFDID       LINK USAGE
---- -------- ------------- -------- ----- ----- ---- ----- ------------------------

Fibre Channel PLDA interface(s):
================================
IFID  NAME   STATUS   TYPE    SPEED        WORLD WIDE NAME              DOMAIN    ADDRESS                    PORT
---- ------ -------- -------  -----  --------------------------------- ------ ------------ ----------------------------------------
1.C          ENABLED  FIBRE   100.0  0x20,0x00,0x00,0xE0,0x8B,0x00,0x4E    5       0x07A    0x20,0x00,0x00,0xE0,0x8B,0x00,0x4E,0xD0

Uninstalled interface(s):
=========================
IFID
----
0.G
```

Example 5 (display PPRC information):

```
SIB> dioint -subsys ABC1 -ifid 0.A,0.C -pprcinfo
```

Displays PPRC information for I/O interfaces 0.A and 0.C on subsystem ABC1.

```
SVAA Server: Server1          Date/Time: 01-02-2002 16:18:22

Subsystem Name:      ABC1
Subsystem Model:     V960
Frame Serial Number: 310001320012
Site Name:           SVAA
Site Location No.:   97802
No. of Interfaces:   32

IFID NAME     STATUS   TYPE     SPEED
---- -------- -------- -------- -----
0.A            ENABLED  FIBRE    100.0
0.C            ENABLED  FIBRE    100.0

PPRC PATH Information:
====================
     INTERNAL PRIMARY SECONDARY SECONDARY SECONDARY   LINK
IFID   IFID   SSID    VCU  SSID  SSNAME   SERIAL NUM  DEST    STATUS    FLAGS
---- -------- ------- --------- --------- ----------- ---- ------------ ------------
0.A    0000    4000   0         5000       78539727    00
       0000    4100   1         5100       78539727    00  Established
0.C    0040    4000   0         5000       78532601    00  Established
       0040    4100   1         5100       78532601    00  Established
       0040    4200   2         5200       78532601    00  Established
```

See Also `alteriointerface`

displayncaplodpct

Name `displayncaplodpct` — display SVA subsystem net capacity load percentage.

Abbreviation `dncl`

Synopsis `displayncaplodpct -subsys` *subsys_name...*

Description This command displays NCL (net capacity load) percentages for specified SVA subsystems. It also displays the total capacity in gigabytes and the collected free space in gigabytes.

Note: NCL percentages are always based on the number of arrays formed.

Note: Capacity is shown in gigabytes (GB) rounded to the nearest tenth (one position after the decimal); therefore, if the capacity is less than 100MB, it will be displayed as 0.0GB.

Security All users allowed.

Options `-subsys` *subsys_name...*

Specifies the name of the SVA subsystem. Entry is case-sensitive. You can use ranges and lists. (See the `sibadmin` command reference for rules about entering string ranges.)

Examples **Example 1:**

```
SIB> displayncaplodpct -subsys *
```

Displays NCL percentages for all SVA subsystems.

```
SVAA Server: Server1           Date/Time: 04-28-2002 09:51:06

SUBSYS    DISK ARRAY  NET CAPACITY          COLLECTED   UNCOLLECTED
 NAME    CAPACITY(GB)  LOAD (GB)   NCL %   FREE SPACE % FREE SPACE %
-------- ------------ ------------ ------- ------------ ------------
ABC2            472.4          1.5     0.3         99.7          0.0
XYZ1            297.5         86.5    29.1         70.9          0.0
XYZ2             94.8         65.2    68.8         31.2          0.0
```

Example 2:

```
SIB> dncl -subsys ABC2
```

Displays NCL percentages for subsystem ABC2.

```
SVAA Server: Server1           Date/Time: 04-28-2002 09:52:13

SUBSYS    DISK ARRAY  NET CAPACITY          COLLECTED   UNCOLLECTED
 NAME    CAPACITY(GB)  LOAD (GB)    NCL %  FREE SPACE % FREE SPACE %
-------- ------------ ------------ ------- ------------ ------------
ABC2            472.4          1.5     0.3         99.7          0.0
```

See Also `release`

displayserver

Name `displayserver` — display information about the SVAA server.

Abbreviation `dserver`

Synopsis `displayserver`

Description This command displays information about the SVAA server, including the server name, SVAA software version, tracing status, installed features, and computer information.

Security All users allowed.

Options None.

Examples **Example 1:**

```
SIB> displayserver
```

Displays server information.

```
SVAA Server: Server1         Date/Time: 10-28-2001 14:39:51

Version: 3.1
Description: Server reserved for IS.
Trace Status: Disabled.
Installed Features:
  SnapShot
Maintenance Level:
  0 PPFinfo PTF=L2P004J Patch=3 FIX=0 Issue=695582
Host:
  Name: Server1
  TCP/IP Port: 41248
  Host OS Level:
    5
    0
```

See Also `get`
`queryversion`
`set`

displaysubsystem

Name `displaysubsystem` — produce a report for an SVA subsystem.

Abbreviation `dsubsys`

Synopsis `displaysubsystem -subsys` *`subsys_name...`* `[-fcapa]`

Description This command produces a report showing the following characteristics of one or more SVA subsystems:

- General information
- Supported features
- Revision level
- Total subsystem capacity

Security All users allowed.

Options `-subsys` *`subsys_name...`*

Specifies the name of the SVA subsystem you want to include on the report. Entry is case-sensitive. You can use ranges and lists. (See the `sibadmin` command reference for rules about entering string ranges.)

`-fcapa`

Optional parameter. Displays the overall capacity of the subsystem. There are no values for this parameter.

Examples **Example 1:**

```
SIB> dsubsys -subsys ABC1
```

Produces a report for subsystem ABC1.

```
SVAA Server: Server1          Date/Time: 4-20-2002 10:08:01

Subsystem Name:      ABC1
Subsystem Model:     V2X
Frame Serial Number: 310000001041
Site Name:           STK-A1
Site Location No.:   1

General Information:
====================
  Subsystem Identifiers          : 00F8 00F9 00FA 00FB 00FC 00FD 00FE 00FF
  Number of Spares               : 8
  Number of Arrays               : 8
  Default Array Size             : 13+2
  Number of:
    SCSI Domains                 : 16
    SCSI Targets per Domain      : 16
    SCSI LUNs per Target         : 256
    Licensed Functional Devices  : 4096
```

```
  Cache Size                     : 8192 MB
  NVS Size                       : 128 MB
  Maximum Number of:
    Channel Interface Supported  : 32
    Physical Devices Supported   : 64
    Virtual Devices Supported    : 4096
    Arrays Supported             : 8
  Physical Capacity Control Limit: 2266.9 GB
  Virtual Volumes per VCU/SSID   : 256
  SCSI SIMS Status               : Disabled
  Remote Operator Panel Status   : Enabled
  Functional Capacity            : 5542.1 GB

Installed Features/Options:
===========================
  CSRC Status                    : Full
  Maintenance Level              : None
  Hot/Cold Spare option
  SnapShot
  Extended Capacity
  Additional Statistics by Extent
  Variable Size Volumes
  High Speed Data Mover (HSDM)
  PPRC
  Full Track Read (FTR)
  SCSI Logical Volume Manager
  SnapShot to PPRC Primary Volume

Revision Level Information:
===========================
  Component Id: 0123456789       Level: 0123456789
  Component Id: 0123456789       Level: 0123456789
  Component Id: 0123456789       Level: 0123456789
  Component Id: 0123456789       Level: 0123456789
```

Example 2 (subsystem capacity):

```
SIB> dsubsys -subsys ABC1 -fcapa
```

Produces a report for subsystem ABC1 that includes total subsystem capacity.

```
SVAA Server: Server1          Date/Time: 4-20-2002 10:12:23

Subsystem Name:      ABC1
Subsystem Model:     V2X
Frame Serial Number: 310000001041
Site Name:           STK-A1
Site Location No.:   1

General Information:
====================
  Subsystem Identifiers          : 00F8 00F9 00FA 00FB 00FC 00FD 00FE 00FF
  Number of Spares               : 8
  Number of Arrays               : 8
  Default Array Size             : 13+2
  Number of:
    SCSI Domains                 : 16
    SCSI Targets per Domain      : 16
    SCSI LUNs per Target         : 256
    Licensed Functional Devices  : 4096
  Cache Size                     : 8192 MB
  NVS Size                       : 128 MB
  Maximum Number of:
    Channel Interface Supported  : 32
    Physical Devices Supported   : 64
```

```
    Virtual Devices Supported    : 4096
    Arrays Supported             : 8
  Physical Capacity Control Limit: 2266.9 GB
  Virtual Volumes per VCU/SSID   : 256
  SCSI SIMS Status               : Disabled
  Remote Operator Panel Status   : Enabled
  Functional Capacity            : 5542.1 GB

Installed Features/Options:
===========================
  CSRC Status                    : Full
  Maintenance Level              : None
  Hot/Cold Spare option
  SnapShot
  Extended Capacity
  Additional Statistics by Extent
  Variable Size Volumes
  High Speed Data Mover (HSDM)
  PPRC
  Full Track Read (FTR)
  SCSI Logical Volume Manager
  SnapShot to PPRC Primary Volume

Revision Level Information:
===========================
  Component Id: 0123456789       Level: 0123456789
  Component Id: 0123456789       Level: 0123456789
  Component Id: 0123456789       Level: 0123456789
  Component Id: 0123456789       Level: 0123456789
```

See Also altersubsystem
querysubsystem
querysubsystempath

dropadministratorgroup

Name `dropadministratorgroup` — remove SVAA administrator and/or PPRC admininistrator privileges from the current SVAA and/or PPRC administrator group.

Abbreviation `dropadmingroup`

Synopsis

```
dropadministratorgroup [-group] [-pprcadmin]
```

Description This command removes SVAA server administrator and/or PPRC (peer-to-peer remote copy) administrator privileges from the Windows group(s) to which the privileges are currently assigned. The affected users lose their administrator privileges immediately.

Note: If you specify no parameters, privileges will be removed from the SVAA administrator group only.

To remove privileges from the current SVAA administrator group only, use either no parameters or the `-group` parameter. Upon completion of the command, `Administrator` will be the only user with SVAA administrator privileges.

Note: Only certain SVA subsystem configurations support the use of PPRC (peer-to-peer remote copy) commands and functions. Contact StorageTek Software Support for details.

To remove privileges from the current PPRC administrator group only, use the `-pprcadmin` parameter. Upon completion of the command, PPRC-related requests will not require PPRC administrator privileges; instead, standard SVAA server and SVA subsystem administration security will be used for PPRC requests.

Note: SVAA always prompts for confirmation.

Security Requires SVAA administrator privileges.

Options

`-group`

Optional parameter. Specifies that you want to drop the current SVAA administrator group only.

`-pprcadmin`

Optional parameter. Specifies that you want to drop the current PPRC administrator group only.

Examples **Example 1 (drop current SVAA administrator group):**

```
SIB> dropadministratorgroup
SIB9841D Drop administrator permissions for -group (y/n)? y
```

Removes the current SVAA administrator group.

Example 2 (drop current SVAA administrator group):

```
SIB> dropadministratorgroup -group
SIB9841D Drop administrator permissions for -group (y/n)? y
```

Removes the current SVAA administrator group.

Example 3 (drop current PPRC administrator group):

```
SIB> dropadministratorgroup -pprcadmin
SIB9841D Drop administrator permissions for -pprcadmin
(y/n)? y
```

Removes the current PPRC administrator group. Once this is done, all PPRC-related requests will use standard SVAA server and SVA subsystem security.

Example 4 (failed request):

```
SIB> dropadministratorgroup
SIB9842W WARNING: No administrator group currently defined.
```

Request fails because there is currently no SVAA nor PPRC administrator group defined.

See Also

`queryadministratorgroup`
`querypermissions`
`querysecuritymode`
`setadministratorgroup`
`setsecuritymode`

dropconnection

Name `dropconnection` — disconnect a user from the SVAA server.

Abbreviation `dropcon`

Synopsis `dropconnection -all|-connection` *connection_id* `[-force]`

Description This command explicitly disconnects a client/user from the server. You must supply the connection identifier (as displayed with the `queryconnections` command). The server drops the user as soon as it receives the request.

Note: By default, SVAA prompts for confirmation. You can bypass the prompt with the `-force` parameter.

Security Requires SVAA administrator privileges.

Options **Note:** You must use either the `-all` or `-connection` parameter, but not both.

`-all`

Drops all users (except the current one) connected to the server. Unless the `-force` parameter is used, SVAA asks you to confirm each user being disconnected.

`-connection` *connection_id*

Specifies the connection ID of the user to be dropped. Entry is case-sensitive. You can enter only one value at a time. Ranges, lists, and wildcards are not supported.

`-force`

Optional parameter. Bypasses the confirmation prompt.

Examples **Example 1:**

```
SIB> dropconnection -connection 26
SIB9807D: Drop user danielm connected on eagle.stortek.com
(y/n)? y
```

Disconnects the user with connection ID 26 from the server.

Example 2:

```
SIB> dropcon -force -all
```

Disconnects all users (except the current one) from the server without prompting for confirmation.

See Also `queryconnections`

`dumpconf`

```
dumpconfiguration -subsys subsystem_name -logfile filename
```

exit

Name `exit` — quit the CLI shell interpreter.

Abbreviation None.

Synopsis

```
exit
```

Description This command quits the CLI shell interpreter and returns to the Command Prompt window. This command is only available from the CLI command shell.

Security All users allowed.

Options None.

Examples

Example 1:

```
SIB> exit
C:\>
```

Exits the CLI interpreter and returns to the Command Prompt window.

See Also `sibadmin`

formarray

Name `formarray` — form one or more arrays.

Abbreviation `formary`

Synopsis `formarray -subsys` *subsys_name* `[-numarys all|`*n*`]`

Note: The `-arysi` parameter, available in prior releases, has been deprecated.

Description This command forms one or more arrays for the Production partition. If you do not enter any of the optional parameters, the command will form as many arrays as the number of spares allows.

All arrays are formed using the default array size for the SVA subsystem, as set initially at the LOP or as changed by the most recent `altersubsystem -arysi` command (for V960 and V2X subsystems, the size must be 13+2; for earlier SVA subsystems, the size can be 5+2 or 13+2). If you want to change the default array size for the SVA subsystem, see the `altersubsystem` command.

Note: Before using this command, you must:

- Form at least one array at the SVA subsystem LOP (local operator panel), and
- Use the `displaydrivemodule` command to make sure you have enough spares to cover the arrays you wish to create (this consists of the total number of drive modules in the Spares and MAT partitions combined).

Security Requires write access to a privileged ECAM device.

Options `-subsys` *subsys_name*

Specifies the name of the SVA subsystem in which the array is to be formed. You can specify only one SVA subsystem name. Entry is case-sensitive, and wildcards are not supported.

`-numarys all|`*n*

Optional parameter. Specifies the number of arrays to be formed. The default value is `all`.

Entering `all` forms as many arrays as the number of available spares allows. This entry is not case-sensitive.

Entering *n* (an integer from 1 to 7) specifies the number of arrays to be formed. If the number of spares available is insufficient to form *n* arrays, the SVA subsystem forms as many arrays as possible.

Note: The maximum number of arrays you can form with this command is one less than the maximum number allowed on the SVA subsystem. This is because one array must already have been formed at the LOP (local operator

panel) when the subsystem was installed. So, for example, if an SVA subsystem is allowed a total of four arrays, you can actually form a maximum of three with this command.

For an explanation of the number of spares required to form arrays, see "Forming Additional Arrays".

Examples

Example 1:

```
SIB> formarray -subsys ABC1 -numarys 7
```

Forms as many arrays for subsystem ABC1 as the number of spares allows, up to a maximum of seven. The size of the arrays is taken from the current default array size for the SVA subsystem, as set initially at the LOP or as changed by the most recent `altersubsystem` command.

```
SIB> formary -subsys ABC2
```

Forms as many arrays for subsystem ABC2 as the number of spares allows. The size of the arrays is taken from the current default array size for the SVA subsystem, as set initially at the LOP or as changed by the most recent `altersubsystem` command.

See Also

`displayarray`
`displaydrivemodule`

get

Name `get` — display the current port on which the SVAA server is running.

Abbreviation None.

Synopsis

```
get -serverport
```

Description This command displays the TCP/IP port number of the currently connected SVAA server.

Note: This command does not check whether a connection has been made on the displayed port. It simply displays the current TCP/IP port number that is or will be used.

Security All users allowed.

Options `-serverport`

Requests the port on which the SVAA server is running.

Examples **Example 1:**

```
SIB> get -serverport
Server port: 41248
```

Displays the current port number, which in this case is the default port number since a different one has not been explicitly assigned.

Example 2:

```
C:\> sibadmin -port 12345
SIB> get -serverport
Server port: 12345
```

Explicitly assigns a port number, then displays it.

See Also `set`

help

Name `help` — display help for a command.

Abbreviation `h`

Synopsis `help [`*CLI_command*`]`

Description This command displays help for a CLI command.

Security All users allowed.

Options *CLI_command*

Optional parameter. Specifies the CLI command for which help is desired. You can enter either the full command name or the abbreviation.

Note: If you do not specify a *CLI_command*, the `sibadmin` help is displayed.

Examples **Example 1:**

```
SIB> help formary
```

Displays help for the `formarray` command.

Example 2:

```
SIB> h displaydevice
```

Displays help for the `displaydevice` command.

See Also `sibadmin`

queryadministratorgroup

Name `queryadministratorgroup` — display the SVAA server and PPRC administrator groups.

Abbreviation `qadmingroup`

Synopsis `queryadministratorgroup`

Description This command displays the Windows groups currently defined as the SVAA administrator group and the PPRC (peer-to-peer remote copy) administrator group.

Security All users allowed.

Options None.

Examples

Example 1:

```
SIB> qadmingroup
```

Displays the current SVAA server administrator and PPRC administrator group(s).

```
SVAA Server: Server1        Date/Time: 12-22-2001 17:28:11

SVAA administrator group: svaa_admin
SVAA PPRC administrator group: svaa_pprc_admin
```

Example 2:

```
SIB> qadmingroup
```

Displays the current SVAA server administrator and PPRC administrator group(s). In this example, `<defaults>` indicates that the SVAA administrator group is set to the default Windows groups, `Administrators` and `Domain Admins`.

```
SVAA Server: Server1        Date/Time: 10-12-2000 14:40:41

SVAA administrator group: <defaults>
SVAA PPRC administrator group: <defaults>
```

See Also

`dropadministratorgroup`
`querypermissions`
`querysecuritymode`
`setadministratorgroup`
`setsecuritymode`

querycache

Name `querycache` — Display SVAA server cache settings.

Abbreviation `qcache`

Synopsis

```
querycache [[-subsys subsys_name...] [-arystatus] [-fdevperf]
[-fdevprops] [-fdevspace] [-intfperf] [-intfprops]
[-pdevperf] [-pdevstatus] [-subsysperf] [-subsysprops]]
```

Description This command displays SVAA server cache settings for one or more SVA subsystems. The data displayed includes cache refresh rates and the number of elements in the SVAA server cache.

Note: The SVAA server cache should not be confused with SVA subsystem cache. The SVAA server cache contains information about the overall SVA subsystem, I/O interfaces, physical devices, and functional devices. The SVA subsystem cache is used in the transfer of user data to and from the SVA subsystem. See "SVAA Server Cache," on page 17 for details.

No parameters are required. Following are parameter usage guidelines:

- To display selected cache settings for selected SVA subsystems, enter the cache type parameters and the subsystem names.
- To display selected cache settings for all SVA subsystems, enter the cache type parameters only and no subsystem name.
- To display all SVAA server cache settings for selected SVA subsystems, enter only the subsystem names.
- To refresh all cache settings for all SVA subsystems, enter no parameters.

Security All users allowed.

Options

`-subsys` *subsys_name...*

Optional parameter. Specifies the name(s) of the SVA subsystem(s) for which you want to display SVAA server cache settings. Entry is case-sensitive. You can specify ranges and lists. (See the `sibadmin` command reference for rules about entering string ranges.)

If no subsystem is specified, current SVAA server cache settings for all SVA subsystems will be displayed.

`-arystatus`

Optional parameter. Indicates to display current settings for array status information in SVAA server cache.

`-fdevperf`

Optional parameter. Indicates to display current settings for functional device performance information in SVAA server cache.

`-fdevprops`

Optional parameter. Indicates to display current settings for functional device properties information in SVAA server cache.

`-fdevspace`

Optional parameter. Indicates to display current settings for functional device space utilitization information in SVAA server cache.

`-intfperf`

Optional parameter. Indicates to display current settings for I/O interface performance information in SVAA server cache.

`-intfprops`

Optional parameter. Indicates to display current settings for I/O interface properties information in SVAA server cache.

`-pdevperf`

Optional parameter. Indicates to display current settings for physical device performance information in SVAA server cache.

`-pdevstatus`

Optional parameter. Indicates to display current settings for physical device status information in SVAA server cache.

`-subsysperf`

Optional parameter. Indicates to display current settings for subsystem performance information in SVAA server cache.

`-subsysprops`

Optional parameter. Indicates to display current settings for subsystem properties information in SVAA server cache.

Examples

Example 1 (display all cache settings for an SVA subsystem):

```
SIB> querycache -subsys ABC1
```

Displays all current SVAA server cache settings for subsystem ABC1.

```
SVAA Server: Server1             Date/Time: 01-24-2003 09:13:30 MST

                                          REFRESH RATE
SUBSYSTEM   CACHE          AREA             (SECS)         ELEMENTS
--------- ---------- ----------------- -------------- ---------------
 ABC1     functional    properties          300            1024
          functional    performance         300            1024
          functional space utilization      300            1024
           physical     performance         900        not initialized
           physical       status            900        not initialized
          subsystem     properties          900              1
          subsystem     performance         900              1
          interface     properties           60             32
          interface     performance          60             32
            array         status            900              8
```

Example 2 (display selected cache settings for multiple SVA subsystems):

```
SIB> qcache -subsys ABC1,ABC2 -fdevprops -subsysprops
-intfprops
```

Displays current settings for functional device, subsystem, and I/O interface properties information in SVAA server cache for subsystems ABC1 and ABC2.

```
SVAA Server: Server1             Date/Time: 01-24-2003 09:13:30 MST

                                          REFRESH RATE
SUBSYSTEM   CACHE          AREA             (SECS)         ELEMENTS
--------- ---------- ----------------- -------------- ---------------
 ABC1     functional    properties          300            1024
          subsystem     properties          900              1
          interface     properties           60             32

 ABC2     functional    properties          300            4096
          subsystem     properties          300              1
          interface     properties          300             32
```

See Also `refreshcache`
`setcacherefreshrate`

queryclientlogentries

Name `queryclientlogentries` — display CLI log.

Abbreviation `qclientlog`

Synopsis `queryclientlogentries [-level` *log_level*`] [-string` *entry*`] [-times` *start_time*`[:`*end_time*`]`

Description This command displays the CLI log entries which are stored in memory. The log is started with the `sibadmin` command. (This log is not associated with CLI tracing.)

You can select entries by log level, search string, and/or time of occurrence.

Note: If you do not specify any options, the entire log will be displayed. A confirmation message will be displayed first, as the log may have up to 1,000 entries.

Security All users allowed.

Options `-level` *log_level*

Optional parameter. Specifies the log level of the entries to be displayed. You may specify one of the following (listed in order of severity):

- `emergency`—Panic condition.
- `alert`—Conditions requiring immediate corrective action, such as a corrupted Windows Registry.
- `critical`—Critical conditions, such as I/O errors.
- `error`—All other errors.
- `warning`—Warning messages.
- `notice`—Conditions that are not errors, but possibly requiring special handling.
- `info`—Informational messages.
- `debug`—Debug messages.

`-string` *entry*

Optional parameter. Specifies a search string for entries to be displayed. Only log entries containing the string will be displayed. Entry is case-sensitive, and wildcards are not supported.

`-times` *start_time*`[:`*end_time*`]`

Optional parameter. Specifies the times for log entries to be displayed.

If you specify both a *start_time* and an *end_time*, the *start_time* must be earlier than, or equal to, the *end_time*.

If you specify only a single time (that is, no *end_time*), the display includes all entries from that time up to and including the current time.

Possible values for this parameter are as follows:

– *start_time* specifies the beginning time for the display period. You can enter it in one of two formats:

  mmddhhmm[yyyy] is an eight- or twelve-digit time string (where the four-digit year is optional). For example, 11:05 on January 18, 2000 would be represented as either `01181105` or `011811052000`.

  -nn is one or two digits (1 to 48), preceded by a minus sign, to indicate a number of hours prior to the current time.

– *:end_time* (optional) specifies the last time to be included in the display. You enter it in *mmddhhmm[yyyy]* format. The default is the current time.

Examples

Example 1:

```
SIB> qclientlog -level info -times -2
-string GetServerAttributes
```

Displays all CLI log entries with an “information” level that were logged within the last two hours and that contain the string “GetServerAttributes”.

```
1999.07.15 02:40:43.157 SIB7350I GetServerAttributes-72 started by hawk-gui. (Severity 6 INFO)
1999.07.15 02:40:43.827 SIB7351I GetServerAttributes-72 finished by hawk-gui. (Severity 6 INFO)
```

See Also

`queryserverlogentries`

queryclienttrace

Name `queryclienttrace` — display CLI trace information.

Abbreviation `qclienttrace`

Synopsis `queryclienttrace`

Description This command displays traced classes, current trace levels, and the trace output destination for the CLI.

Note: Normally, you would use this command only under the direction of StorageTek Software Support.

Security All users allowed.

Options None.

Examples **Example 1:**

```
SIB> qclienttrace
```

Displays client trace information.

```
SVAA Server: Server1     Date/Time: 10-28-2000 08:04:23

Trace side:       Client
Identifier:       Trace#0
Level(s):         severe, detail
Log file:         C:\temp\sibadminclitrace.log
Traced class(es): com.storagetek.blackhawk.application.cli.data.Server
                  com.storagetek.blackhawk.application.cli.data.Subsystem
```

See Also
`queryservertrace`
`startclienttrace`
`stopclienttrace`

queryconnections

Name `queryconnections` — display users connected to the server.

Abbreviation `qcon`

Synopsis `queryconnections [-detail]`

Description This command displays userid and computer information on all users currently connected to the server.

Security All users allowed.

Options `-detail`

Optional parameter. Indicates to display additional detail for each user consisting of how long the user has been logged in and whether the user is a member of the SVAA administrator or PPRC (peer-to-peer remote copy) administrator group(s) (see "Security Modes," on page 29 for details about the administrator groups). There are no values for this parameter.

Examples **Example 1:**

```
SIB> queryconnections
```

Displays all users connected to the SVAA server.

```
SVAA Server: Server1        Date/Time: 11-05-2000 08:04:23

        ID             USERID            HOST
------------------ ---------------- ----------------
                25 rwss             fred.stortek.com
                26 davidel          wilma.stortek.co
                                    m
               127 priverso         dino.stortek.com
               201 priverso         betty.stortek.co
                                    m
               203 oscarlr          barney.stortek.c
                                    om
```

Example 2 (detailed display):

```
SIB> qcon -detail
```

Displays detailed information about all users connected to the SVAA server.

```
SVAA Server: Server1          Date/Time: 11-22-2001 08:05:38
         ID      USERID             HOST           PERM                              LOGTIME
---------- ---------------- ---------------  -------------------------------   ----------------
         25 rwss             fred.stortek.com  Member of SVAA Admin Group       11-29-2001 14:48
         26 davidel          wilma.stortek.co  Member of PPRC Admin Group       11-29-2001 13:16
                             m
        127 priverso         dino.stortek.com  Not Member of SVAA Admin Group   11-19-2001 09:34
        201 priverso         betty.stortek.co  Not Member of SVAA Admin Group   11-20-2001 16:05
                             m
        203 oscarlr          barney.stortek.c  Member of SVAA Admin Group       11-21-2001 07:12
                             om
```

See Also `dropconnection`
`querypermissions`

querypermissions

Name querypermissions — display permission levels for Windows users or groups.

Abbreviation qperm

Synopsis querypermissions -user [*domain*\]*user_name*[,[*domain*\]*user_name*] |-group *group_name*[,*group_name*]

Description This command displays user privileges for specified Windows users or groups. If the security mode is set to either "active" or "warn," each user or group will be identified as either an administrator or a user. If the security mode is set to "none," all users are designated SVAA administrators. For a complete explanation of user privileges, see "Security," on page 22.

Security All users allowed.

Options **Note:** You must use either the -user or -group parameter, but not both.

-user [*domain*\]*user_name*[,[*domain*\]*user_name*]

Specifies the user for whom you want to display user privileges. You must specify a *domain*\ if the user is a member of a Windows domain. Valid entries are any users listed in the User Manager.

You can specify a list of users, but not a range. Wildcards are not supported.

-group *group_name*[,*group_name*]

Specifies the Windows group for which you want to display user privileges. Valid entries are any groups listed in the User Manager.

You can specify a list of groups, but not a range. Wildcards are not supported.

Examples **Example 1:**

```
SIB> querypermissions -user barney,betty,orion\robert
```

Displays the user privileges for the users barney and betty on the local computer, and robert in the orion domain.

```
SVAA Server: Server1          Date/Time 11-14-2001 11:19:13

      USERID                  PERM
--------------- ------------------------------
barney          Member of SVAA Admin Group
wilma           Member of PPRC Admin Group
betty           Not Member of SVAA Admin Group
betty           Not Member of PPRC Admin Group
orion\robert    Member of SVAA Admin Group
```

Example 2:

```
SIB> qperm -group svaa_admin,svaa_pprc_admin,acct
```

Displays the user privileges for the Windows groups svaa_admin, svaa_pprc_admin, and acct.

```
SVAA Server: Server1          Date/Time:12-11-2001 08:14:05

     GROUPID                  PERM
---------------- ------------------------------
svaa_admin       SVAA Admin Group
svaa_pprc_admin  SVAA PPRC Admin Group
acct             Not SVAA Admin Group
acct             Not SVAA PPRC Admin Group
```

See Also dropadministratorgroup
dropconnection
queryadministratorgroup
queryconnections
querysecuritymode
setadministratorgroup
setsecuritymode

queryrequests

Name `queryrequests` — display pending server requests.

Abbreviation `qreq`

Synopsis
`queryrequests -all|-user [`*domain*`\]`*user_name*`|`
`-connection` *connection_id*

Description This command displays pending requests on the SVAA server. Options are to display:

- All requests on the server.
- All requests for a specified user.
- Requests for a specified connection identifier (as displayed with the `queryconnections` command).

Security All users allowed.

Options **Note:** You must use one and only one of the following parameters: `-all`, `-user`, or `-connection`.

`-all`

Displays all pending requests on the server.

`-user [`*domain*`\]`*user_name*

Displays all pending requests for the specified user. You must specify a *domain*`\` if the user is a member of a Windows domain.

Entry is case-sensitive. You can enter only one value at a time. Ranges, lists, and wildcards are not supported.

`-connection` *connection_id*

Displays all pending requests for the specified connection ID. Entry is one to eight alphanumeric characters, case-sensitive. You can enter only one value at a time. Ranges, lists, and wildcards are not supported.

Examples **Example 1:**

```
SIB> queryrequests -all
```

Displays pending requests for all users on the server.

```
SVAA Server: Server1   Date/Time: 04-06-2001 19:01:51
      REQUEST ID         USER       CONNECTION ID        HOST            STARTTIME
---------------------- ------------ ---------------- --------------  --------------------
ListServerRequests-4  barney                     16 localhost       04-06-2000 19:01:16
```

See Also `queryconnections`

querysecuritymode

Name `querysecuritymode` — display the current SVAA security mode.

Abbreviation `qsecuritymode`

Synopsis `querysecuritymode`

Description This command displays the SVAA security mode currently in effect.

Security All users allowed.

Options None.

Examples **Example 1:**

```
SIB> querysecuritymode
```

Displays the current security mode.

```
SVAA Server: Server1          Date/Time:11-02-2000 15:10:21
SVAA security mode: ACTIVE
SIB>
```

See Also
`dropadministratorgroup`
`queryadministratorgroup`
`querypermissions`
`setadministratorgroup`
`setsecuritymode`

queryserverlogentries

Name `queryserverlogentries` — display SVAA server log.

Abbreviation `qserverlog`

Synopsis `queryserverlogentries [-level` *log_level*`] [-string` *entry*`] [-times` *start_time*`[:`*end_time*`]`

Description This command displays the SVAA server log entries which are stored in memory. The log is started with the SVAA Manager program. (This log is not associated with SVAA server tracing.)

You can select entries by log level, search string, and/or time of occurrence.

Note: If you do not specify any options, the entire log will be displayed. A confirmation message will be displayed first, as the log may have up to 1,000 entries.

Security All users allowed.

Options `-level` *log_level*

Optional parameter. Specifies the log level of the entries to be displayed. You may specify one of the following (listed in order of severity):

- `emergency`—Panic condition.
- `alert`—Conditions requiring immediate corrective action, such as a corrupted Windows Registry.
- `critical`—Critical conditions, such as I/O errors.
- `error`—All other errors.
- `warning`—Warning messages.
- `notice`—Conditions that are not errors, but possibly requiring special handling.
- `info`—Informational messages.
- `debug`—Debug messages.

`-string` *entry*

Optional parameter. Specifies a search string for entries to be displayed. Only log entries containing the string will be displayed. Entry is case-sensitive, and wildcards are not supported.

`-times` *start_time*`[:`*end_time*`]`

Optional parameter. Specifies the times for log entries to be displayed.

If you specify both a *start_time* and an *end_time*, the *start_time* must be earlier than, or equal to, the *end_time*.

If you specify only a single time (that is, no *end_time*), the display includes all entries from that time up to and including the current time.

Possible values for this parameter are as follows:

– *start_time* specifies the beginning time for the display period. You can enter it in one of two formats:

  mmddhhmm[yyyy] is an eight- or twelve-digit time string (where the four-digit year is optional). For example, 11:05 on January 18, 2000 would be represented as either `01181105` or `011811052000`.

  `-`*nn* is one or two digits (1 to 48), preceded by a minus sign, to indicate a number of hours prior to the current time.

– `:`*end_time* (optional) specifies the last time to be included in the display. You enter it in *mmddhhmm*[*yyyy*] format. The default is the current time.

Examples

Example 1:

```
SIB> qserverlog -level info -times -2
-string GetServerAttributes
```

Displays all SVAA server log entries with an “information” level that were logged within the last two hours and that contain the string “GetServerAttributes”.

```
2001.01.15 02:40:43.157 SIB7350I GetServerAttributes-72 started by hawk-gui. (Severity 6 INFO)
2001.01.15 02:40:43.827 SIB7351I GetServerAttributes-72 finished by hawk-gui. (Severity 6 INFO)
```

See Also

queryservertrace

Name `queryservertrace` — display trace information.

Abbreviation `qservtrace`

Synopsis `queryservertrace`

Description This command displays traced classes, current trace levels, and the current trace output destination for the server.

Note: Normally, you would use this command only under the direction of StorageTek Software Support.

Security All users allowed.

Options None.

Examples **Example 1:**

```
SIB> qservtrace
```

Displays server trace information.

```
SVAA Server: Server1         Date/Time:   03-28-2001 08:04:23

Trace side:       Server
Identifier:       Trace#0
Level(s):         severe, error, warning
Log file:         C:\temp\sibservertrace.log
Traced class(es): com.storagetek.blackhawk.request.admin.AddServer
                  com.storagetek.blackhawk.request.server.GetSubsystems
```

See Also `queryclienttrace`
`startservertrace`
`stopservertrace`

querysubsystem

Name `querysubsystem` — list available SVA subsystems.

Abbreviation `qsubsys`

Synopsis `querysubsystem`

Description This command displays a list of SVA subsystems accessible to the local SVAA server.

Security All users allowed.

Options None.

Examples **Example 1:**

```
SIB> querysubsystem
```

Displays the list of current SVA subsystems known to the local server.

```
SVAA Server: Server1      Date/Time:   03-28-2001 08:04:23

ABC1
ABC2
ABC3
XYZ1
```

See Also
`altersubsystem`
`displaysubsystem`
`querysubsystempath`

querysubsystempath

Name querysubsystempath — list available ECAM devices.

Abbreviation qsubsyspath

Synopsis querysubsystempath -subsys *subsys_name...*

Description This command displays the list of device paths to SVA subsystems known (accessible) to the local SVAA server. Devices displayed with this command are known as ECAM devices. (This command is equivalent to the QUERY ECAMDEVICE subcommand in SVAA for S/390.)

Security All users allowed.

Options -subsys *subsys_name...*

Specifies the SVA subsystem for which you want to display device paths. Entry is case-sensitive. You can use ranges and lists. (See the sibadmin command reference for rules about entering string ranges.)

Examples

Example 1:

```
SIB> querysubsystempath -subsys *
```

Displays the paths from the SVAA server to all SVA subsystems.

```
SVAA Server: Server1          Date/Time: 03-28-2001 08:04:23

SUBSYS
  NAME                DEVICE PATHS
-------- ----------------------------------------
ABC1     J:
ABC1     P:
ABC3     G:
ABC3     J:
```

Example 2:

```
SIB> qsubsyspath -subsys ABC1
```

Displays the paths from the SVAA server to the ABC1 subsystem.

```
SVAA Server: Server1           Date/Time: 03-28-2001 08:05:47

SUBSYS
  NAME                DEVICE PATHS
-------- ----------------------------------------
ABC1     J:
ABC1     P:
```

See Also

addsubsystempath
querysubsystem
removesubsystempath

queryversion

Name `queryversion` — display the SVAA server software version.

Abbreviation `qvers`

Synopsis `queryversion`

Description This command displays the version level of the SVAA server software.

Security All users allowed.

Options None.

Examples **Example 1:**

```
SIB> qvers
```

Displays the SVAA server software version

```
SVAA Server: Server1      Date/Time: 12-10-2001 13:01:19 MDT
Shared Virtual Array Administrator Server 3.1.0 PPFinfo PTF=L2P004J Patch=10 Fix=0 Issue=695582 for
Windows 2000
```

See Also `displayserver`
`sibadmin`

refreshcache

Name `refreshcache` — Force immediate refresh of all or part of the SVAA server cache.

Abbreviation `rcache`

Synopsis

```
refreshcache [[-subsys subsys_name] [-arystatus] [-fdevperf]
[-fdevprops] [-fdevspace] [-intfperf] [-intfprops]
[-pdevperf] [-pdevstatus] [-subsysperf] [-subsysprops]]
```

Description This command forces the immediate refresh of all or part of the SVAA server cache. The command allows you to specify the cache elements you wish to refresh. It also allows you to specify the particular SVA subsystems the refresh applies to.

Note: The SVAA server cache should not be confused with SVA subsystem cache. The SVAA server cache contains information about the overall SVA subsystem, I/O interfaces, physical devices, and functional devices; the server keeps a separate cache for each SVA subsystem to which it has access. The SVA subsystem cache is used in the transfer of user data to and from the SVA subsystem. See "SVAA Server Cache," on page 17 for details.

No parameters are required. Following are parameter usage guidelines:

- To refresh selected cache elements for selected SVA subsystems, enter the cache type parameters and the subsystem names.
- To refresh selected cache elements for all SVA subsystems, enter the cache type parameters only and no subsystem name.
- To refresh all SVAA server cache elements for selected SVA subsystems, enter the subsystem names only.
- To refresh all cache elements for all SVA subsystems, enter no parameters.

Security Requires SVAA server administrator privileges. Requires subsystem alter privileges for the affected SVA subsystems.

Options

`-subsys` *subsys_name*

Optional parameter. Specifies the name(s) of the SVA subsystem(s) for which you want to refresh all or part of SVAA server cache. Entry is case-sensitive. You can use ranges and lists.

If no subsystem is specified, the cache element(s) specified on the command will be refreshed for all SVA subsystems.

`-arystatus`

Optional parameter. Refreshes storage array cache.

`-fdevperf`

Optional parameter. Refreshes functional device performance cache.

`-fdevprops`

Optional parameter. Refreshes functional device property cache.

`-fdevspace`

Optional parameter. Refreshes functional device space utilization cache.

`-intfperf`

Optional parameter. Refreshes I/O interface performance cache.

`-intfprops`

Optional parameter. Refreshes I/O interface property cache.

`-pdevperf`

Optional parameter. Refreshes physical device performance cache.

`-pdevstatus`

Optional parameter. Refreshes physical device status cache.

`-subsysperf`

Optional parameter. Refreshes SVA subsystem performance cache.

`-subsysprops`

Optional parameter. Refreshes SVA subsystem property cache.

Examples

Example 1:

```
SIB> refreshcache -fdevperf -fdevspace
```

Refreshes functional device performance and space utilization cache for all SVA subsystems in SVAA server cache.

Example 2:

```
SIB> rcache -subsys ABC1 ABC2 -intfprops -intfperf
```

Refreshes I/O interface property and performance cache for subsystems ABC1 and ABC2 in SVAA server cache.

See Also

`querycache`
`setcacherefreshrate`

release

Name `release` — initiate space release on a SCSI partition.

Abbreviation None.

Synopsis `release -path` *drive_letter*`[,`*drive_letter*`] [-force]`

Description This command initiates space release on a SCSI partition which is identified by a drive letter.

Before issuing this command, you should make sure the partition is not being accessed by any application and that it has no open files. If there are open files on the partition, SVAA will attempt to close them; if it is unable to close them, the request will fail.

After releasing a partition, you must re-format it before you can use it.

Note: You cannot release a PPRC (peer-to-peer remote copy) secondary volume.

Note: Releasing space causes all data on the partition to be completely erased. By default, a confirmation prompt will be displayed. You can bypass the prompt with the `-force` parameter.

Security Requires write access to the partition named in the request.

Options `-path` *drive_letter*`[,`*drive_letter*`]`

Identifies specific drive letters to be released.

You can specify a list of drive letters, but not a range. Entry is not case-sensitive. Wildcards are not supported.

Note: *drive_letter* cannot be a PPRC secondary volume.

`-force`

Optional parameter. Bypasses the confirmation prompt.

Examples **Example 1 (no prompt for confirmation):**

```
SIB> release -force -path K:
```

Initiates space release on drive letter `K:`. Does not prompt for confirmation.

Example 2 (prompt for confirmation):

```
SIB> release -path m:
SIB9855D: Initiate data space release for M: (y/n)? y
```

Initiates space release on drive letter `M:`.

Example 3:

```
SIB> release -path L:,p:
SIB9855D: Initiate data space release for L: (y/n)? n
SIB9855D: Initiate data space release for P: (y/n) y
```

Initiates space release on drive letter P:, but not on drive letter L:.

See Also `displayncaplodpct`

removesubsystempath

Name `removesubsystempath` — remove a path to an SVA subsystem from use by the local SVAA server.

Abbreviation `removesubsyspath`

Synopsis `removesubsystempath -subsys` *subsys_name* `-devpath` *device_path* `[-force]`

Description This command removes a path to an SVA subsystem from use by the local SVAA server. After you use this command, the specified device is no longer an ECAM device. (This command is equivalent to the `DROP ECAMDEVICE` subcommand in SVAA for S/390.)

Security Requires SVAA administrator privileges.

Options

`-subsys` *subsys_name*

Specifies the SVA subsystem from which a path is to be removed. You can specify only one SVA subsystem name. Entry is case-sensitive, and wildcards are not supported.

`-devpath` *device_path*

Specifies the device path to be removed.

device_path must be entered exactly as it is shown in the `querysubsystempath` display.

`-force`

Optional parameter. Indicates that the specified path is to be removed, even if it is not a valid ECAM device. Certain changes to the SVA subsystem hardware may make previously defined ECAM devices invalid. This parameter enables you to remove such devices from the `config.dat` file; without this parameter, only valid ECAM devices can be removed.

Examples

Example 1:

```
SIB> removesubsystempath -subsys ABC1 -devpath F:
```

Removes the `F:` path between the SVAA server and the ABC1 subsystem.

Example 2 (forced deletion of an invalid ECAM device):

```
SIB> removesubsystempath -subsys ABC1 -devpath J:
SIB9633E ERROR: Server cannot process the request - Specified ECAM device is
not connected to the specified subsystem.
SIB> removesubsystempath -subsys ABC1 -devpath J: -force
```

First attempt to remove the `J:` path between the SVAA server and the ABC1 subsystem fails because the path is not a valid ECAM device. Second attempt, using the `-force` parameter, is successful.

See Also `addsubsystempath`
`querysubsystempath`

reportevents

Name `reportevents` — produce a drain event report.

Abbreviation None.

Synopsis

```
reportevents -subsys subsys_name...
[-ident event_id]|[[-dates startdate[:enddate]|last] [-detail]]
```

Description This command produces a report on drain events initiated within the most recent 60 days. You can limit the content of the report by specifying a single event number, or by specifying a date and whether the report is to be detailed.

Note: The event start and completion times shown on this report come from the SVA subsystem; they do not necessarily correspond to the times set on the SVAA server.

Security All users allowed.

Options

`-subsys` *subsys_name...*

Specifies the name of the SVA subsystem for which events are to be reported. Entry is case-sensitive. You can use ranges and lists. (See the `sibadmin` command reference for rules about entering string ranges.)

`-ident` *event_id*

Optional parameter. Specifies the SVAA-generated identifier for the event to be reported. Each *event_id* is a one- to ten-digit integer, with a value ranging from `1` to `4294967295`.

Note: If you enter this parameter, you cannot use the `-dates` or `-detail` parameters.

`-dates` *startdate*`[:`*enddate*`]|last`

Optional parameter. Specifies the dates for events to be included in the report.

Note: If you use the `-ident` parameter, you cannot use the `-dates` parameter.

The *startdate* and *enddate* cannot be more than 60 days prior to the current date. (A maximum of 60 days of event data is stored by the SVA subsystem.)

If you specify both a *startdate* and an *enddate*, the *startdate* must be earlier than, or equal to, the *enddate*. The report covers the entire day for each date in the range.

If you specify the same date for the *startdate* and *enddate*, the reports covers only that one day.

If you specify only a single date (that is, no *enddate*), the report includes all events from that date up to and including the current date and time.

Possible values for this parameter are as follows:

– *startdate* specifies the beginning date for the report period. You can enter it in one of four formats:

– *mmddyyyy* is an eight-digit date string. For example, January 18, 2000 would be represented as 01182000.

– `-`*nn* is one or two digits (1 to 60), preceded by a minus sign, to indicate a number of days prior to the current date.

`0` specifies today's date.

`*` specifies that all 60 days up to and including the current date are to be included in the report period.

– `:`*enddate* (optional) specifies the last date to be included in the report period. You can enter it in either *mmddyyyy* or `-`*nn* format. The default is today's date.

– `last` specifies that the report is to cover only the last drain event. This entry is case-sensitive.

Default is `-7` (the seven-day period up to and including the current date), except when you use the `-ident` parameter.

`-detail`

Optional parameter. Specifies that the report is to be detailed. There are no values for this parameter.

Note: If you do not specify this parameter, a summary report is produced.

Examples

Example 1:

```
SIB> reportevents -subsys ABC1
```

Prints a summary report for subsystem ABC1. The report includes all drain events that occurred during the period from seven days ago, up to and including the current date and time.

```
SVAA Server: Server1          Date/Time : 07-28-2000 06:01:49 MDT

Subsystem Name:      ABC1
Subsystem Model:     9500
Frame Serial Number: 310000001041
Site Name:           STK-A1
Site Location No.:   97802

                                                          PERCENT    ---INITIATING---- --NOTIFICATION---
IDENTIFIER TYPE         STARTED            COMPLETED      COMPLETE   NODE ID  USER ID  NODE ID  USER ID
---------- ------ ------------------- ------------------- --------   -------- -------- -------- --------
213984124  DRAIN  01-11-2000 16:31:49 01-11-2000 16:31:49      100   colorado edwards  toronto  reynold
213984125  DRAIN  01-11-2000 17:35:40                          7   colorado edwards  toronto  reynold
213984126  DRAIN  01-11-2000 18:35:49 01-11-2000 19:41:49      100   colorado edwards  toronto  reynold
213984127  DRAIN  01-11-2000 19:35:40 01-11-2000 21:16:40      100   colorado edwards  toronto  reynold
213984128  DRAIN  01-11-2000 20:31:49                         28   colorado edwards  toronto  reynold
213984129  DRAIN  01-11-2000 21:31:49                         35   colorado edwards  toronto  reynold
213984130  DRAIN  01-11-2000 22:31:49                         42   colorado edwards  toronto  reynold
213984131  DRAIN  01-01-2000 23:31:49                         49   colorado edwards  toronto  reynold
213984132  DRAIN  01-12-2000 00:31:49 01-12-2000 05:11:49          colorado edwards  toronto  reynold
213984133  DRAIN  01-12-2000 01:31:49                         63   colorado edwards  toronto  reynold
```

Example 2:

```
SIB> reportevents -subsys ABC1 -dates 0 -detail
```

Prints a detailed report for subsystem ABC1. The report includes all drain events that occurred today.

```
SVAA Server: Server1          Date/Time : 07-28-2000 06:05:41 MDT

Subsystem Name:      ABC1
Subsystem Model:     9500
Frame Serial Number: 310000001041
Site Name:           STK-A1
Site Location No.:   1

   Drain Identifier: 213984125                Initiating Node ID: colorado
            Started: 01-11-2000 17:35:40      Initiating User ID: edwards
          Completed:                          Completion Node ID: toronto
   Percent Complete: 7                                User ID: reynold

       Serial                      Serial                      Serial
U.T.S  Number  Status       U.T.S  Number Status       U.T.S  Number  Status
0.3.0 00005865  V.N         0.3.1 00005865 V.N         0.3.2 00005865  V.N
0.3.3 00005865  V.N         0.3.4 00005865 V.N         0.3.5 00005865  V.N
0.3.6 00005865  V.N         0.3.7 00005865 V.N

--------------------------------------------------------------------------
   Drain Identifier: 213984126                Initiating Node ID: colorado
            Started: 01-11-2000 18:35:40      Initiating User ID: edwards
          Completed: 01-11-2000 19:41:49      Completion Node ID: toronto
   Percent Complete:                                  User ID: reynold

       Serial                      Serial                      Serial
U.T.S  Number  Status       U.T.S  Number Status       U.T.S  Number  Status
0.3.0 00005865  V.N         0.3.1 00005865 V.N         0.3.2 00005865  V.N
0.3.3 00005865  V.N         0.3.4 00005865 V.N         0.3.5 00005865  V.N
0.3.6 00005865  V.N         0.3.7 00005865 V.N

--------------------------------------------------------------------------
   Drain Identifier: 213984127                Initiating Node ID: colorado
            Started: 01-11-2000 19:35:40      Initiating User ID: edwards
          Completed: 01-11-2000 21:20:40      Completion Node ID: toronto
   Percent Complete:                                  User ID: reynold

       Serial                      Serial                      Serial
U.T.S  Number Status       U.T.S  Number Status       U.T.S  Number  Status
0.3.0     5865 V.N         0.3.1     5865  V.N         0.3.2   5865   V.N
0.3.3     5865 V.N         0.3.4     5865  V.N         0.3.5   5865   V.N
0.3.6     5865 V.N         0.3.7     5865  V.N
```

See Also `startdrain`

set

Name `set` — direct the CLI to establish a connection to the SVAA server on a specified port.

Abbreviation None.

Synopsis `set -serverport` *server_port*

Description This command directs the CLI to connect to an SVAA server running on a specific TCP/IP port. All subsequent commands issued during this CLI session are directed to that port. This command performs a connection to the specified port to determine whether a server is actually running on the port.

Normally, the CLI starts with a connection to the default server port specified in the Windows Registry. If, however, you start a server on a different port, and then start the CLI without explicitly designating the new port, the CLI will not be able to communicate with the server. At this point you can perform either of the following steps:

- **Immediately** use the `set -serverport` command to direct the CLI to communicate with the server on the new port (if you try to use any other command prior to this one, the CLI will become disabled), or
- Exit the CLI and restart it with a connection to the new port (for example, `sibadmin -port 21176`).

Security All users allowed.

Options `-serverport` *server_port*

Specifies the TCP/IP port to be used for communication between the CLI and the SVAA server. Entry must be an integer.

Examples **Example 1:**

```
C:\> sibadmin
SIB> set -serverport 12345
SIB>
```

Assuming the server has been started with the SVAA Manager on port 12345, the first command starts the CLI communicating with the default SVAA server port (which may or may not have a server running). The second command connects the CLI to the SVAA server running on port number 12345; all subsequent commands will be issued to the server on that port.

See Also `displayserver`
`get`

setadministratorgroup

Name `setadministratorgroup` — set a Windows group as the SVAA administrator group or the PPRC administrator group.

Abbreviation `setadmingroup`

Synopsis

```
setadministratorgroup [-group group_name]
[-pprcadmin group_name]
```

Description This command identifies a Windows group that will be checked to determine a user's SVAA administrator status or PPRC (peer-to-peer remote copy) administrator status. The change takes place immediately; you do not need to shut down and restart the SVAA server.

Note: SVAA always prompts for confirmation.

Security Must be logged in as `Administrator`.

Options **Note:** You must use the `-group` parameter or the `-pprcadmin` parameter, or both.

`-group` *group_name*

Optional parameter. Specifies the Windows group that you want to define as the SVAA administrator group. This group is checked when determining a user's SVAA administrator status.

Entry is case-sensitive. You can enter only one value at a time. Ranges, lists, and wildcards are not supported.

`-pprcadmin` *group_name*

Note: Only certain SVA subsystem configurations support the use of PPRC (peer-to-peer remote copy) commands and functions. Contact StorageTek Software Support for details.

Optional parameter. Specifies the group that you want to define as the PPRC administrator group. If this group is defined, it will be checked whenever a PPRC-related command is issued to determine the user's PPRC administrator status. If this group is not defined, standard SVAA server and SVA subsystem security will be used for PPRC-related requests.

Entry is case-sensitive. You can enter only one value at a time. Ranges, lists, and wildcards are not supported.

Examples **Example 1 (define the SVAA administrator group):**

```
SIB> setadministratorgroup -group svaa_admin
SIB9854D: Set administrator rights for group(s) svaa_admin
(y/n)? y
```

Assigns SVAA administrator privileges to the Windows group `svaa_admin`.

Example 2 (failed command):

```
SIB> setadmingroup -group svaa_admin
SIB9859W: Administrator group already defined: svaa_admin.
```

Since the Windows group `svaa_admin` already has SVAA administrator privileges, the request is ignored.

Note: In order to change the SVAA administrator group from one group to another, you must use the `dropadministratorgroup` command before the `setadministratorgroup` command.

Example 3 (define the PPRC administrator group):

```
SIB> setadministratorgroup -pprcadmin svaa_pprc_admin
SIB9854D: Set administrator rights for group(s)
svaa_pprc_admin (y/n)? y
```

Assigns PPRC administrator privileges to the Windows group `svaa_pprc_admin`. Once this is done, to execute PPRC-related requests a user must be a member of the PPRC administrator group (as well as have write access to a privileged ECAM device).

See Also

`dropadministratorgroup`
`queryadministratorgroup`
`querysecuritymode`
`querypermissions`
`setsecuritymode`

setcacherefreshrate

Name `setcacherefreshrate` — Set the refresh rate for all or part of the SVAA server cache.

Abbreviation `setcache`

Synopsis

```
setcacherefreshrate -subsys subsys_name...
[-allcache seconds|default|off]|
[[-arystatus seconds|default|off]
[-fdevperf seconds|default|off]
[-fdevprops seconds|default|off]
[-fdevspace seconds|default|off]
[-intfperf seconds|default|off]
[-intfprops seconds|default|off]
[-pdevperf seconds|default|off]
[-pdevstatus seconds|default|off]
[-subsysperf seconds|default|off]
[-subsysprops seconds|default|off]]
```

Description This command sets the rate at which all or part of the SVAA server cache will be refreshed. SVAA server cache contains data about SVA subsystem elements, such as storage arrays, functional devices, physical devices, and the overall subsystem. The server keeps a separate cache for each SVA subsystem to which it has access.

Note: The SVAA server cache should not be confused with SVA subsystem cache. The latter is used in the transfer of user data to and from the SVA subsystem. See “SVAA Server Cache,” on page 17 for details.

All cache refresh rates are in number of seconds, from a minimum of 60 seconds (1 minute) to 86,400 seconds (24 hours). The default rate is 300 seconds (5 minutes).

You can specify the SVA subsystem(s) to which the cache refresh rates will apply. You can also turn off some or all of the SVAA server cache. At least one parameter must be entered. Following are parameter usage guidelines:

- To set the refresh rate for selected cache elements for selected SVA subsystems, enter the cache type parameters and the subsystem names.
- To set the refresh rate for selected cache elements for all SVA subsystems, enter the cache type parameters only and no subsystem name.
- To set the refresh rate for all SVAA server cache elements for selected SVA subsystems, enter the `-allcache` parameter and the subsystem names.
- To set the refresh rate for all cache elements for all SVA subsystems, enter the `-allcache` parameter only.

Note: PPRC (peer-to-peer remote copy) properties are not included in SVAA server cache. This information is always obtained directly from the SVA subsystem.

The changes specified by this command take effect immediately.

Security Requires SVAA server administrator privileges. Requires subsystem alter privileges for the affected SVA subsystems.

Options `-subsys` *subsys_name*

Specifies the name(s) of the SVA subsystem(s) to which you want to apply the rate(s) you set. Entry is case-sensitive. You can use ranges and lists.

`-allcache` *seconds*`|default|off`

Note: This parameter cannot be used with any other cache type parameters.

Optional parameter. Specifies how frequently all SVAA server cache information should be refreshed. You must specify one of the following:

– *seconds*—Sets all cache refresh rates to the specified number of seconds. Entry must be an integer from 60 to 86400 (1 minute to 24 hours).

– `default`—Sets all cache refresh rates to the default of 300 seconds.

– `off`—Turns all caching off.

`-arystatus` *seconds*`|default|off`

Note: This parameter cannot be used with the `-allcache` parameter.

Optional parameter. Specifies how frequently storage array cache should be refreshed. This information includes current array status, such as draining or initializing. You must specify one of the following:

– *seconds*—Sets the rate to the specified number of seconds. Entry must be an integer from 60 to 86400.

– `default`—Sets the rate to the default of 300 seconds.

– `off`—Turns off the array cache.

`-fdevperf` *seconds*`|default|off`

Note: This parameter cannot be used with the `-allcache` parameter.

Optional parameter. Specifies how frequently functional device performance cache should be refreshed. This information includes functional device capacity, number of requests per device, amount of data transferred, and time available. You must specify one of the following:

– *seconds*—Sets the rate to the specified number of seconds. Entry must be an integer from 60 to 86400.

– `default`—Sets the rate to the default of 300 seconds.

– `off`—Turns off the functional device performance cache.

`-fdevprops` *seconds*`|default|off`

Note: This parameter cannot be used with the `-allcache` parameter.

Optional parameter. Specifies how frequently functional device property cache should be refreshed. This information varies depending on the type of

device; it may include DTL (device, target, LUN) address, device type, read/write-enabled status, ECAM-privileged status, etc. You must specify one of the following:

- *seconds*—Sets the rate to the specified number of seconds. Entry must be an integer from 60 to 86400 (1 minute to 24 hours).
- `default`—Sets the rate to the default of 300 seconds.
- `off`—Turns off the functional device property cache.

`-fdevspace` *seconds*`|default|off`

Note: This parameter cannot be used with the `-allcache` parameter.

Optional parameter. Specifies how frequently the functional device space utilization cache should be refreshed. This information includes number of device tracks available and used. You must specify one of the following:

- *seconds*—Sets the rate to the specified number of seconds. Entry must be an integer from 60 to 86400.
- `default`—Sets the rate to the default of 300 seconds.
- `off`—Turns off the functional device space utilization cache.

`-intfperf` *seconds*`|default|off`

Note: This parameter cannot be used with the `-allcache` parameter.

Optional parameter. Specifies how frequently the I/O interface performance cache should be refreshed. This information includes I/O interface speed, number of I/Os, amount of time busy, etc. You must specify one of the following:

- *seconds*—Sets the rate to the specified number of seconds. Entry must be an integer from 60 to 86400.
- `default`—Sets the rate to the default of 300 seconds.
- `off`—Turns off the I/O interface performance cache.

`-intfprops` *seconds*`|default|off`

Note: This parameter cannot be used with the `-allcache` parameter.

Optional parameter. Specifies how frequently the I/O interface property cache should be refreshed. This information includes I/O interface name, type, and status. You must specify one of the following:

- *seconds*—Sets the rate to the specified number of seconds. Entry must be an integer from 60 to 86400.
- `default`—Sets the rate to the default of 300 seconds.
- `off`—Turns off the I/O interface property cache.

`-pdevperf` *seconds*`|default|off`

Note: This parameter cannot be used with the `-allcache` parameter.

Optional parameter. Specifies how frequently the physical device performance cache should be refreshed. This information includes amount of time a device is busy, and number of bytes transferred for read/write operations. You must specify one of the following:

- *seconds*—Sets the rate to the specified number of seconds. Entry must be an integer from 60 to 86400.
- `default`—Sets the rate to the default of 300 seconds.
- `off`—Turns off the physical device performance cache.

`-pdevstatus` *seconds*`|default|off`

Note: This parameter cannot be used with the `-allcache` parameter.

Optional parameter. Specifies how frequently the physical device status cache should be refreshed. This information includes physical device location, current status, array assignment, and capacity. You must specify one of the following:

- *seconds*—Sets the rate to the specified number of seconds. Entry must be an integer from 60 to 86400.
- `default`—Sets the rate to the default of 300 seconds.
- `off`—Turns off the physical device status cache.

`-subsysperf` *seconds*`|default|off`

Note: This parameter cannot be used with the `-allcache` parameter.

Optional parameter. Specifies how frequently the subsystem performance cache should be refreshed. This information includes subsystem free space, NCL, back-end capacity, cache sizes, and number of ECAM messages processed and bypassed. You must specify one of the following:

- *seconds*—Sets the rate to the specified number of seconds. Entry must be an integer from 60 to 86400.
- `default`—Sets the rate to the default of 300 seconds.
- `off`—Turns off the subsystem performance cache.

`-subsysprops` *seconds*`|default|off`

Note: This parameter cannot be used with the `-allcache` parameter.

Optional parameter. Specifies how frequently the subsystem property cache should be refreshed. This information includes installed features, number of devices, and number of I/O interfaces supported. You must specify one of the following:

- *seconds*—Sets the rate to the specified number of seconds. Entry must be an integer from 60 to 86400.
- `default`—Sets the rate to the default of 300 seconds.
- `off`—Turns off the subsystem property cache.

Examples

Example 1:

```
SIB> setcacherefreshrate -subsys ABC1 -fdevperf 60
-fdevspace default -arystatus off
```

Sets server cache refresh rates for data from SVA subsystem ABC1 only. Functional device performance cache is set to 60 seconds; functional device space cache is set to the default of 300 seconds; array status cache is turned off.

Example 2:

```
SIB> setcache -allcache 600
```

Sets server cache refresh rates for data from all SVA subsystems. All cache refresh rates are set to 600 seconds.

See Also

`querycache`
`refreshcache`

setsecuritymode

Name `setsecuritymode` — set the mode for SVAA security operations.

Abbreviation `setsmode`

Synopsis `setsecuritymode -mode` *mode*

Description This command sets the SVAA security mode. The security mode defines the level of authorization checking performed on SVAA commands. The change takes place immediately; you do not need to shut down and restart the SVAA server.

Security Requires SVAA administrator privileges.

Options `-mode` *mode*

Specifies the mode in which you want SVAA security to operate. You may specify one of the following:

- `active`—SVAA security is in full force. Authorization checks are performed on all commands. If a user is not authorized to perform a command, an error message is displayed and the command fails. This is the default mode.
- `warn`—Authorization checks are performed on all commands. If a user is not authorized to perform a command, a warning message is logged, but the command continues to completion. Essentially allows all users to run all commands, but with warning messages logged to the SVAA server log; no error message is displayed at the CLI.
- `none`—Authorization checks are not performed. All users can run all commands. It is not recommended that you use this mode.

Caution: While running in “none” mode, it is possible to set an SVAA administrator group that does not exist. This is because “none” mode provides no security checking to verify that the group you specify with the `setadministratorgroup` command actually exists. Should you do this and then subsequently change the mode to “active,” no users would be authorized, since the non-existent group would have no members; `Administrator` would be the only userid authorized to submit requests to SVAA.

Entry is not case-sensitive.

Examples **Example 1:**

```
SIB>setsecuritymode -mode None
SIB>
```

Sets the security mode to “none.” The change takes place immediately.

Example 2:

```
SIB>setsmode -mode ACTIVE
SIB>
```

Sets the security mode to “active.” The change takes place immediately.

See Also

```
dropadministratorgroup
queryadministratorgroup
querypermissions
querysecuritymode
setadministratorgroup
```

sibadmin

Name `sibadmin` — initiate the SVAA CLI (command line interface).

Abbreviation None.

Synopsis

```
sibadmin [[-config file_name] [-debug format]
[-port server_port]] | [-version]
```

Description This command initiates the SVAA CLI (command line interface) in the current Command Prompt window.

Note: This is not a CLI command. It can only be issued from a Command Prompt window. You can also start this command from the Windows **Start** menu, but without any of the optional parameters (see "Starting the SVAA CLI Shell," on page 86 for details).

The CLI gives access to all SVAA commands. Following are rules for entering parameters on any of the commands:

- Enter each parameter name only once—SVAA does not allow duplicate parameters.
- The parameter name must be preceded by a dash (`-`). For example, `-subsys`.
- Follow each parameter name by the value(s) for that parameter.
- Except where indicated, parameter values are case-sensitive.
- Enter each parameter value only once where multiple values are allowed—SVAA ignores duplicate values (for example, an entry of `-name CORC1,CORC1,CORC2` will be interpreted as `-name CORC1,CORC2`).
- You can arrange the parameters in any order.
- Where indicated, you can use the standard wildcard symbols `*` and `?`. For example: `-subsys ABC* -ifid 0:?`.
- When a parameter supports both lists and ranges, you can use both at once. For example, `-aryid 1:4,7`.
- When a parameter supports a list of values, you must separate the values by commas (,). For example, `-aryid 1,3,6`.

  Note: The Windows command line shell treats a comma (,) as an argument separator, the same as a blank space. For this reason, if you include CLI commands in a batch file (`sibadmin.bat`), the arguments must all be enclosed in quotation marks (`" "`) or the commands will not be parsed correctly. For example:
  `sibadmin.bat displaydevice "-subsys SOLAR -fdid 23,25"`
- Rules for entering ranges:

  - You must separate the “from” and “to” entries with a colon (:). For example: `-aryid 2:7`.
  - The “from” and “to” entries must be entered in ascending order. For example: `20:50` is valid; `50:20` is not.

- Rules specific to entering string ranges:
  - The “from” and “to” entries must be the same length. For example: `-subsys XYZ00:XYZ12` and `-name 00ABC:05ABC` are valid; `-subsys XYZ0:XYZ12` and `-name 00ABC:5ABC` are not.
  - The alpha portions of the two entries must be the same. For example: `-name A01:A05` and `-subsys ABC1:ABC7` are valid; `-name A01:B02` and `-subsys ABC:XYZ` are not.
  - The “from” numeric value must be less than the “to” numeric value. For example: `-subsys ABC00:ABC05` is valid; `-subsys ABC05:ABC00` is not.
  - Neither entry can contain wildcards (* or ?). For example: `-name A?1:A?C5` and `-subsys ABC01:ABC*` are not valid.

Options

`-config` *file_name*

Optional parameter. Specifies the location (full pathname) of the configuration file to be used by the CLI. When the CLI starts, it will use the startup values specified in this file instead of those in the Windows Registry set during SVAA configuration.

`-debug` *format*

Optional parameter. Activates trace logging on the CLI for diagnostic purposes.

format (an integer from 1 to 7) specifies the format for the trace entries. You can enter only one value at a time. Ranges, lists, and wildcards are not supported.

Note: Normally, you would use this parameter only under the direction of StorageTek Software Support. Your representative will give you the appropriate *format* value to use.

`-port` *server_port*

Optional parameter. Connects the CLI to the SVAA server running on the specified TCP/IP port. All subsequent commands issued during this CLI session are directed to that server port. (Normally, when the CLI starts it forms a connection to the SVAA server port specified in the Windows Registry).

Entry must be an integer.

`-version`

Optional parameter. Displays the version level of the SVAA CLI software, but does not actually initiate the CLI.

Note: If you use this parameter, any other parameters will be ignored. Therefore, you should use this parameter alone.

Examples

Example 1:

```
C:\> sibadmin
SIB>
```

Initiates the CLI in the current Command Prompt window.

Example 2 (display CLI version):

```
C:\> sibadmin -version
Shared Virtual Array Administrator 3.1 for Windows.
0 PPFinfo PTF=L2P004J Patch=3 FIX=0 Issue=695582
C:\>
```

Displays the version of the CLI.

See Also

addsubsystempath
alterdevice
alteriointerface
altersubsystem
attndevice
definedevice
deletedevice
displayarray
displaydevice
displaydrivemodule
displayiointerface
displayncaplodpct
displayserver
displaysubsystem
dropadministratorgroup
dropconnection
exit
formarray
get
help
queryadministratorgroup
querycache
queryclientlogentries
queryclienttrace
queryconnections
querypermissions
queryrequests
querysecuritymode
queryserverlogentries
queryservertrace
querysubsystem
querysubsystempath

```
queryversion
refreshcache
release
removesubsystempath
reportevents
set
setadministratorgroup
setcacherefreshrate
setsecuritymode
sibadmin
snap
startclienttrace
startdrain
startservertrace
stopclienttrace
stopservertrace
```

`-config` *pathname*

Optional parameter. Specifies the location of the `config.dat` file to be used by the SVAA server. When the server starts up, it will use the values specified in this file instead of those in the Windows Registry which were defined during SVAA configuration.

Enter the full pathname where the `config.dat` file is located.

`-debug` *format*

Optional parameter. Activates trace logging on the server side for diagnostic purposes.

format (an integer from 1 to 7) specifies the format for the trace entries. You can enter only one value at a time. Ranges, lists, and wildcards are not supported.

Note: Normally, you would use this parameter only under the direction of StorageTek Software Support. Your support representative will give you the appropriate *format* value to use.

`-ecamdev` *device_path*

Optional parameter. Specifies the ECAM device to be used by the SVAA server to access the SVA subsystem. This entry overrides the default ECAM device specified in the Windows Registry.

Entry must be a valid ECAM device previously defined with the `addsubsystempath` command. Enter the full pathname of the device.

`-log` *file_name*

Optional parameter. Specifies the log file to be used by the SVAA server (where SVAA error messages are sent). This entry overrides the log file specified in the Windows Registry.

Enter the full pathname, including the file name.

`-nowebport`

Optional parameter. Disables the WBI for this SVAA server. This entry overwrites the WBI status in the Windows Registry, disabling the WBI for this SVAA server.

Once this parameter is used, the WBI remains disabled for this SVAA server until the interface is explicitly re-enabled.

`-port` *port_number*

Optional parameter. Specifies the port number to be used by the SVAA server's TCP/IP connections. This entry overrides the port number in the Windows Registry.

Enter any integer between 1024 and 64567.

Note: If you have multiple SVAA servers running on a single host machine, each server must have a unique port number.

`-serveridletime` *minutes*

Optional parameter. Specifies that the SVAA cache refresh cycle is to be suspended if there is no cache activity for the specified number of minutes. The cache refresh cycle will be restarted as soon as cache activity is resumed. "Cache activity" is defined as any request that queries or modifies data in the SVAA cache, such as `displaydevice`, `startdrain`, or `refreshcache`; commands such as `queryversion` and `setadministratorgroup` do not affect SVAA cache and therefore do not affect this option.

Enter the number of minutes; entry can be any integer between 1 and 10080 (seven days). You must specify a value; there is no default.

If you do not specify this parameter, SVAA server cache refreshes will never be suspended for this instance of the server.

`-webport` *port_number*

Optional parameter. Specifies the TCP/IP port number to be used for communications between the SVAA server and the WBI. This entry overwrites the port number in the Windows Registry, making it the new default WBI port number for this SVAA server.

Note: This entry must not be the same as the port number of any active SVAA servers (SVAA server port numbers are defined in the Windows Registry and can be overridden by the `-port` parameter).

Enter any integer between 1024 and 64567.

Note: If the WBI was previously disabled for this SVAA server, specifying this parameter re-enables it. The WBI will remain enabled until explicitly disabled.

snap

Name `snap` — create a SnapShot image of a SCSI partition, striped or spanned volume (created on Windows NT 4 only), or entire physical drive on an SVA subsystem.

Abbreviation None.

Synopsis `snap -source` *device_id* `-target` *device_id*

Description This command performs a SnapShot copy of one SCSI device to another. Both SCSI devices, hereafter called objects, must be on the same SVA subsystem and must be the same size. Additionally, for striped and spanned volumes, all their members must be the same size.

SnapShot for Windows can duplicate any of the following objects:

- Entire SCSI partitions created with Disk Management within Computer Management on Windows; these are identified by drive letters.
- Striped and spanned volumes created on Windows NT 4; identified by drive letters.
- Entire physical drives; identified by Windows Device Manager drive names.
- Windows 2000/2003 Dynamic Disk volumes.

Note: SnapShot for Windows 2003 is supported by SVAA Patch 20 and higher.

SnapShot cannot distinguish the contents of a SCSI device; it simply copies the device as a whole. There is no support for snapping a file within a file system.

SnapShot currently does not support Veritas Logical Volume Manager (LVM) on Windows.

Caution: Attempting to use SnapShot on unsupported objects may have unpredictable results.

Before issuing this command, you should make sure the target device is not being accessed by any applications and that it has no open files. If there are open files on the device, SVAA will attempt to close them; if it is unable to close them, the request will fail. Also, it is recommended, but not required, that the source device have no open files.

Caution: Having open files on the source device during the snap may result in inconsistent, or even corrupted, data on the target. After the snap, always run `chkdsk` on the target device to ensure data integrity of the copy.

Caution: There are significant data loss considerations when running this command. A SnapShot copy to a target object that contains data will irrevocably

erase that data. It is strongly recommended that you read Chapter 4., "SnapShot" of the *SVA Administrator for Windows User's Guide* before using this command.

Note: `Administrator` is subject to the same permission checks as all other users. If for some reason `Administrator` does not have the required access permissions to the source and/or target object(s), the snap will fail and an error message will be issued.

Note: You can snap to a PPRC primary volume, but not to a PPRC secondary volume.

Security

Requires read access to the source object and read/write access to the target object named in the request.

Options

`-source` *device_id*

Identifies the source object, the one to be copied. *device_id* can be any of the following:

- An entire SCSI partition, identified by a drive letter (such as `F:`)
- A striped and spanned volume, identified by a drive letter (such as `h:`)
- An entire physical drive, identified by the physical drive name from Windows Device Manager (such as `PhysicalDrive1`)
- A Windows 2000 Dynamic Disk volume

Entry is not case-sensitive for drive letters. Entry is case-sensitive for physical drives. Wildcards are not supported.

Note: The source and target devices must be of the same type. For example, if the source device is a drive letter, the target must also be a drive letter, not a physical drive.

Note: The snap source can be a PPRC (peer-to-peer remote copy) primary volume, but not a PPRC secondary volume. The snap source cannot be a PPRC bridge volume.

`-target` *device_id*

Identifies the target object, the one receiving the copy. *device_id* can be any of the following:

- An entire SCSI partition, identified by a drive letter (such as `F:`)
- A striped and spanned volume, identified by a drive letter (such as `h:`)
- An entire physical drive, identified by the physical drive name from Windows Device Manager (such as `PhysicalDrive1`)
- A Windows 2000 Dynamic Disk volume

Entry is not case-sensitive for drive letters. Entry is case-sensitive for physical drives. Wildcards are not supported.

Note: The source and target devices must be of the same type. For example, if the source device is a physical drive name, the target must also be a physical drive name, not a drive letter.

Note: The snap target can be a PPRC primary volume, but not a PPRC secondary volume, nor a PPRC bridge volume.

Examples

Example 1 (snapping a drive letter):

```
SIB> snap -source j: -target s:
```

Initiates a snap of drive letter J: onto drive letter S:.

Example 2 (snapping a physical drive name):

```
SIB> snap -source PhysicalDrive0 -target PhysicalDrive1
```

Initiates a snap of all of physical drive 0 onto physical drive 1.

See Also

None.

startclienttrace

Name `startclienttrace` — activate trace logging on the client side.

Abbreviation None.

Synopsis `startclienttrace [-class` *class_name*`...] [-format` *format_name*`] [-level all|`*level_name*`...] [-logfile` *file_name*`]`

Description This command activates trace logging on the CLI for diagnostic purposes. The trace process stays active until it is stopped with the `stopclienttrace` command or the CLI is terminated with the `exit` command.

Note: Normally, you would use this command only under the direction of StorageTek Software Support.

By default, the trace output is sent to the defined standard output device (`stdout`) on the terminal on which the CLI is running. You can optionally use the `-logfile` parameter to direct the output to a different location.

Note: Only one trace process can be running at a time. If you want to initiate another trace, you must first stop the current trace process with the `stopclienttrace` command, then start a new one.

Security All users allowed.

Options `-class` *class_name*`...`

Optional parameter. Specifies the fully qualified class name to be traced.

Entry is case-sensitive. You can use ranges and lists. (See the `sibadmin` command reference for rules about entering string ranges.) The `*` wildcard is supported (but not the `?` wildcard). The default value is `*`.

Note: Specifying "*" causes all classes starting with "com.storagetek" to be traced.

`-format` *format_name*

Optional parameter. Specifies the way trace entries are written. The default value is `full`.

Only one format can be specified. Valid formats are as follows:

- `basic`—adds a timestamp prefix to each trace entry.
- `noprefix`—writes only the trace itself.
- `thread`—adds a timestamp prefix and thread information to each trace entry.
- `method`—adds a timestamp prefix and method name to each trace entry.

– `callstack`—adds a timestamp prefix and callstack information to each trace entry.

– `object`—adds a timestamp prefix and object name information to each trace entry.

– `full`—writes all possible information for each trace entry (equivalent to `callstack` + `method` + `object` + `thread`).

`-level all|`*level_name*`[,`*level_name*`]`

Optional parameter. Specifies the trace level to be activated. The default value is `all`.

Entering `all` or no value activates all tracing levels. This entry is not case-sensitive.

Entering *level_name* activates the specified trace level(s). You can enter a list of one or more trace levels. The * wildcard is supported (but not the ? wildcard).

Note: The * wildcard works recursively through the class tree.

Valid trace levels are as follows:

– `class`—class-level actions.

– `component`—product component interactions.

– `detail`—extra information.

– `entryexit`—method entries and exits.

– `error`—errors from which it is possible to recover and function properly.

– `method`—the execution process of a method.

– `severe`—errors that seriously affect system functionality.

– `warning`—conditions where there is no obvious error, but the state is abnormal.

`-logfile` *file_name*

Optional parameter. Specifies the full pathname of the file to which you want the trace output directed. The file must be on a local drive, as the CLI cannot access any files on network drives. If the file already exists and you have write permission to the file, it is overwritten without a warning.

If you do not enter a full pathname, the file is created in the current folder (the one in which the CLI was started).

If you do not use this parameter, the output is sent to the defined standard output device (`stdout`) on the terminal on which the CLI is running.

Examples

Example 1:

```
SIB> startclienttrace
-class com.storagetek.blackhawk.application.cli.data.Server
-level severe, detail -logfile C:\temp\sibadminclitrace.log
```

Activates service and detail logging on the com.storagetek.blackhawk.application.cli.data.Server class, and directs the output to the specified file.

Example 2:

```
SIB> startclienttrace -class * -level detail
```

Activates detail-level logging on all classes whose names start with "com.storagetek", and sends the output to the defined standard output device on the client.

See Also

```
queryclienttrace
startservertrace
stopclienttrace
```

startdrain

Name `startdrain` — initiate a drain of one or more drive modules.

Abbreviation None.

Synopsis

```
startdrain -subsys subsys_name
-aryid array_id|-dmod u.t.s[,u.t.s]|-tray u.t
```

Description This command initiates a drain of one or more drive modules. A drain moves data off one or more drive modules to other drive modules in order to free the space on the source drives.

You can specify the drive modules individually, by array, or by tray. When the designated drives have been successfully drained (that is, all data has been moved), they are placed in the MAT partition.

Note: Before you begin the drain, the number of drive modules in the MAT and Spares partitions combined must be equal to or greater than the number of drives to be drained (in order to accept the drained data). For this reason, you cannot drain the last array, drive module, or tray.

Security Requires write access to a privileged ECAM device.

Options `-subsys` *subsys_name*

Specifies the name of the SVA subsystem that is to perform the drain.You can specify only one subsystem name. Entry is case-sensitive, and wildcards are not supported.

Note: You must use one and only one of the following parameters: `-aryid`, `-dmod`, or `-tray`.

`-aryid` *array_id*

Specifies the array to be drained. An *array_id* is an integer from `0` to `7`. You can specify only one array.

`-dmod` *u.t.s*`[,`*u.t.s*`]`

Specifies one or more individual drive modules to be drained. Each *u.t.s* (three digits separated by periods) identifies a specific drive module, where:

– *u* identifies the disk array unit (0 to 3)

– *t* identifies the tray (0 to 3)

– *s* identifies the slot (0 to 7)

You can specify a list of drive modules, but not a range. Wildcards are not supported.

When you specify a drain of individual drive modules, the SVA subsystem must have at least as many spares available as there are drives to be drained. If there are not enough available spares, the command is rejected.

`-tray` *u.t*

Specifies the tray to be drained. You can specify only one tray. *u.t* (two digits separated by a period) identifies a specific tray, where:

– *u* identifies the disk array unit (0 to 3)

– *t* identifies the tray (0 to 3)

Examples

Example 1:

```
SIB> startdrain -subsys ABC1 -dmod 0.1.7,0.0.4
SIB9808I: Drain initiated on subsystem (Process Id 123). Check
Report Events for details.
```

Drains drive modules 0.1.7 and 0.0.4 in subsystem ABC1.

Example 2:

```
SIB> startdrain -subsys ABC2 -aryid 1
SIB9808I: Drain initiated on subsystem (Process Id 594). Check
Report Events for details.
```

Drains all of disk array 1 in subsystem ABC2.

See Also

`displayarray`
`displaydrivemodule`
`reportevents`

startservertrace

Name `startservertrace` — activate trace logging on the server side.

Abbreviation None.

Synopsis

`startservertrace [-class` *class_name*`...] [-format` *format_name*`] [-level all|`*level_name*`...] [-logfile` *file_name*`]`

Description This command activates trace logging on the server side for diagnostic purposes. The trace process stays active until it is stopped with the `stopservertrace` command or the server is shut down with the SVAA Manager.

Note: Normally, you would use this command only under the direction of StorageTek Software Support.

By default, the trace output is sent to the defined standard output device (`stdout`) on the terminal on which the CLI is running. You can optionally use the `-logfile` parameter to direct the output to a different location.

Note: Only one trace process can be running at a time. If you want to initiate another trace, you must first stop the current trace process with the `stopservertrace` command, then start a new one.

Security Requires SVAA administrator privileges.

Options

`-class` *class_name*`...`

Optional parameter. Specifies the fully qualified class name to be traced.

Entry is case-sensitive. You can use ranges and lists. (See the `sibadmin` command reference for rules about entering string ranges.) The * wildcard is supported (but not the ? wildcard). The default value is *.

Note: Specifying "*" causes all classes starting with "com.storagetek" to be traced.

`-format` *format_name*

Optional parameter. Specifies the way trace entries are written. The default value is `full`.

Only one format can be specified. Valid formats are as follows:

- `basic`—adds a timestamp prefix to each trace entry.
- `noprefix`—writes only the trace itself.
- `thread`—adds a timestamp prefix and thread information to each trace entry.
- `method`—adds a timestamp prefix and method name to each trace entry.

- `callstack`—adds a timestamp prefix and callstack information to each trace entry.
- `object`—adds a timestamp prefix and object name information to each trace entry.
- `full`—writes all possible information for each trace entry (equivalent to `callstack` + `method` + `object` + `thread`).

`-level all|`*level_name*`[,`*level_name*`]`

Optional parameter. Specifies the trace level to be activated. The default value is `all`.

Entering `all` activates all tracing levels. This entry is not case-sensitive.

Entering *level_name* activates the specified trace level(s). You can enter a list of one or more trace levels. The `*` wildcard is supported (but not the `?` wildcard).

Note: The `*` wildcard works recursively through the class tree.

Valid trace levels are as follows:

- `class`—class-level actions.
- `component`—product component interactions.
- `detail`—extra information.
- `entryexit`—method entries and exits.
- `error`—errors from which it is possible to recover and function properly.
- `method`—the execution process of a method.
- `severe`—errors that seriously affect system functionality.
- `warning`—conditions where there is no obvious error, but the state is abnormal.

`-logfile` *file_name*

Optional parameter. Specifies the full pathname of the server file to which you want the trace output directed. The file must be on a local drive, as the SVAA server cannot access any files on network drives. If the file already exists and you have write permission to the file, it is overwritten without a warning.

If you do not enter a full pathname, the file is created in the current folder (the one in which the SVAA server was started).

If you do not use this parameter, the output is sent to the defined standard output device (`stdout`) on the terminal on which the CLI is running.

Examples

Example 1:

```
SIB> startservertrace
-class com.storagetek.blackhawk.request.Request,
com.storagetek.blackhawk.request.admin.AddServer
-level severe, detail -logfile sibservertrace.log
```

Activates severe- and detail-level logging on the com.storagetek.blackhawk.request.Request and com.storagetek.blackhawk.request.admin.AddServer classes, and directs the output to the specified file (which is located in the folder from which the server was started, since a full pathname was not given).

Example 2:

```
SIB> startservertrace -class *
-level detail,error,warning,severe
```

Activates detail-, error-, warning-, and severe-level logging on all classes whose names start with "com.storagetek", and sends the output to the defined standard output device on the server.

See Also

`queryservertrace`
`sibadmin`
`stopservertrace`

stopclienttrace

Name `stopclienttrace` — deactivate a CLI trace process.

Abbreviation None.

Synopsis `stopclienttrace [-id` *trace_id*`]`

Description This command deactivates a trace process on the CLI. By default, the command stops the current client trace. Optionally, you can specify the ID of the trace process you want to stop.

Note: Normally, you would use this command only under the direction of StorageTek Software Support.

Security All users allowed.

Options `-id` *trace_id*

Optional parameter. Specifies the identifier of the trace process to be stopped. You can enter only one value at a time, and wildcards are not supported.

Note: Use the `queryclienttrace` command to display active trace IDs.

Examples **Example 1:**

```
SIB> stopclienttrace
```

Stops the current CLI trace process.

Example 2:

```
SIB> stopclienttrace -id Trace#24
```

Stops trace process Trace#24 on the CLI.

See Also

`queryclienttrace`
`sibadmin`
`stopservertrace`

stopservertrace

Name `stopservertrace` — deactivate a server trace process.

Abbreviation None.

Synopsis `stopservertrace [-id` *trace_id*`]`

Description This command deactivates the specified trace process on the SVAA server. By default, the command stops the current server trace. Optionally, you can specify the ID of the trace process you want to stop

Note: Normally, you would use this command only under the direction of StorageTek Software Support.

Security Requires SVAA administrator privileges.

Options `-id` *trace_id*

Optional parameter. Specifies the identifier of the trace process to be stopped. You can enter only one value at a time, and wildcards are not supported.

Note: Use the `queryservertrace` command to display active trace IDs.

Examples **Example 1:**

```
SIB> stopservertrace
```

Stops the current server trace process.

Example 2:

```
SIB> stopservertrace -id Trace#19
```

Stops trace process Trace#19 on the server.

See Also

`queryservertrace`
`startservertrace`
`stopclienttrace`

Appendix A. Drive Module Status Codes

Figure A-1, “Drive module status transitions” illustrates how the status of a drive module is updated by various SVA subsystem activities.

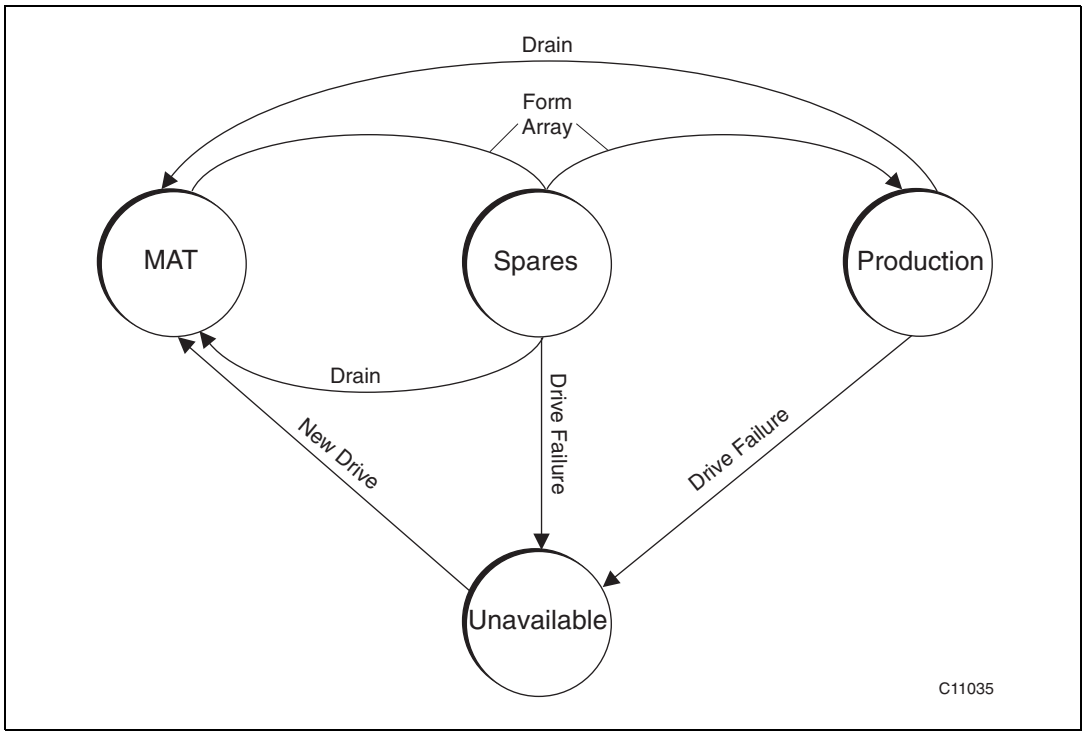


Figure A-1 Drive module status transitions

Note: The transition from MAT to Production by the Form Array function is actually a two-step process: first the drives are temporarily moved to Spares, then they are moved to Production. This is done automatically, and is therefore usually transparent to the user.

Table A-1, “Drive module status descriptions” lists and describes the specific statuses that a drive module can acquire. Status is indicated by a two-character code:

- The first character identifies the partition with which the drive is associated;
- The second character shows the current state of the drive.

Status codes are displayed either as two characters or as two characters separated by a period. For example, the status of a drive module associated with the Production partition in the active state is “PA” or “P.A”.

Table A-1 Drive module status descriptions Page 1 of 2

Drive Module Status	Status Code	Description
MAT Partition		
MAT:Active	M.A (MA)	Drive module is active and available.
MAT:Fenced	M.F (MF)	Drive module has been fenced for diagnostics for one of the following reasons: • New drive module pre-acceptance test (capacity upgrade). • Guided FRU replacement (drive module removal).
Production Partition		
Production:Active	P.A (PA)	Drive module is a member of an array that is associated with the Production partition.
Production:Copy (receiving drain data)	P.C (PC)	Drive module is receiving data from the drain of a single drive module in the Production partition.
Production:Draining	P.D (PD)	Drive module is being drained.
Production:Initialize array	P.I (PI)	Drive module is part of an array initialization process.
Production:Pending drain (multiple device drain)	P.P (PP)	Drive module is waiting to be drained. The drain cannot begin for one of the following reasons: • A drive module reconstruction is in progress. • Another drive in the array is being drained. • The number of available spares is inadequate (this can occur if the number of spares was reduced after the drain request was accepted).
Production:Reconstruction	P.R (PR)	Drive module is being used to reconstruct data for a Production partition drive module that has failed.
Spares Partition		
Spares:Active	S.A (SA)	Drive module is available for forming arrays, for data reconstruction, or for receiving data from a drain of a single drive module.
Spares:Draining	S.D (SD)	Drive module is currently being drained.
Spares:Pending drain	S.P (SP)	Drive module is waiting to be drained, pending completion of a periodic drive test.
Spares:Reserved	S.R (SR)	Drive module is reserved for diagnostics.

Table A-1 Drive module status descriptions Page 2 of 2

Drive Module Status	Status Code	Description
Unavailable Partition		
Unavailable:Broken	U.B (UB)	Drive module in slot is broken.
Unavailable:Isolated	U.I (UI)	Drive module is broken and isolated, but the SSA path is still operational.
Unavailable:No active drive module	U.N (UN)	No active drive module is sensed in slot, or slot has not been installed.
Unavailable:No slot installed	U.S (US)	Drive slot is not installed.

Appendix B. Tracing

SVAA provides separate trace functions for the server and the CLI. You can use these functions to help StorageTek Software Support diagnose and resolve problems.

Note: Normally, you would use tracing only under the direction of a StorageTek Software Support representative.

Starting a Trace

SVAA server tracing and CLI tracing work in essentially the same way. You turn on SVAA server tracing with the `startservertrace` command, and you turn on CLI tracing with the `startclienttrace` command. With either command, you specify the Java class to be traced and the severity level of the events to be included. Your StorageTek Software Support representative should give you instructions on what to specify.

Note: Only one server trace process and/or one CLI trace process can be running at a time. While the process is running you may notice a significant degradation in system performance.

Trace Log File

Once you have turned on tracing, records of all events you have requested are written to a trace log file. You can optionally specify a trace log file name at the time you start the trace process. The file must be on a local drive, as the SVAA server cannot access any files on network drives. If you do not specify a file name, the trace output is sent to the defined standard output device (`stdout`).

Depending on the level of tracing and the events being traced, the trace log file can quickly become very large. You should monitor the size of the file as it grows to make sure it does not run out of room. If the trace log file does become full, a message is written in the SVAA server log file indicating that the trace log is full (see “SVAA Server Log,” on page 20 for details).

Monitoring a Trace

You can monitor the progress of active tracing processes through either the `queryservertrace` or `queryclienttrace` command. Either query shows the classes being traced, the trace levels, the log file where the output is being written, and the trace process ID.

Stopping a Trace

Tracing continues until the trace process is stopped. Methods of stopping a trace process depend on the type of tracing being performed:

- For server tracing, you can use the `stopservertrace` command to stop the specific trace process, or you can use the SVAA Manager program to shut down the SVAA server.
- For CLI tracing, you can use the `stopclienttrace` command to stop the specific trace process, or you can use the `exit` command to terminate the CLI.

To use either stop trace command, you must specify the trace process ID which is displayed with the `queryservertrace` or `queryclienttrace` command.

Once you have stopped the trace process, your StorageTek Software Support representative may want you to view the trace log file header to verify some information. Then they will give you instructions on sending the file to StorageTek Software Support for evaluation.

Appendix C. Interpreting SVAA Displays and Queries

displayarray

Field	Definition	Calculation
ARRAY	Arrays specified in the request.	None
CONFIG	Number of drives in the array.	None
FORMATION IN PROGRESS	Indicates whether the array is currently being formed.	None
SPARE(S) NEEDED	Indicates whether the array needs spares. A “Yes” entry is probably due to a drive reconstruction, drive drain, or array configuration in progress.	None
TWO DRIVE FAILURE	Indicates whether two drives in the array have failed.	None
GROUP DRAIN STATUS	Indicates whether there is a drain pending or in process for the array. If “Pending,” one or more drives in the array need to be drained. Possible values: • In Progress • Pending • N.A.	None
---DEVICE STATUSES---		
INITIALIZING	Indicates whether the array is initializing.	None
DRAINING	Indicates whether there is a drain pending or in progress for the array. Possible values: • In Progress • Pending • N.A.	None

Field	Definition	Calculation
RECONSTRUCTING	Indicates whether there is a reconstruction pending or in progress for the array. Possible values: • In Progress • Pending • N.A.	None

displaydevice

Field	Definition	Calculation
Subsystem Name:	SVA subsystem name specified in the request.	None
Subsystem Model:	Model number of the subsystem.	None
Frame Serial Number:	Serial number of the subsystem.	None
Site Name:	Name of the site.	None
Site Location No.:	StorageTek-assigned site location number.	None
Logical Devices:	Total number of logical devices in the subsystem.	None
Functional Capacity:	Total capacity for the subsystem specified in the request. Displayed only if the -subsysfcapa parameter was used on the command.	None
Device Information:		
FDID	FDID specified in the request.	None
FDID NAME	Name assigned to the device.	None
TYPE	Indicates whether the device is SCSI or CKD.	None
CAPACITY (GB)	Total device capacity, in GB.	None
---SCSI---		
DOMAIN TARGET LUN	Domain, target, and LUN for the device. If the device is not SCSI, entry is a dash (–).	None
INDEX	SCSI larger LUN index for the device (0 indicates the parent device; entry >0 indicates a child device). Displayed only if the -functional parameter was used on the command.	None
ENBLD	Indicates whether SCSI I/O access is enabled.	None
R/W	Indicates whether SCSI read/write access is enabled.	None

Field	Definition	Calculation
ACCESS	SCSI Interface Accessibility Count. The number of subsystem I/O interfaces with SCSI addressing entries that map to this FDID.	None
CACHE	Indicates whether cache is enabled.	None
---CKD---		
ENBLD	Indicates whether CKD I/O access is enabled.	None
R/W	Indicates whether CKD read/write access is enabled.	None
PRVLG ECAM	Indicates whether this is a privileged ECAM device.	None
PPRC STATE	Indicates the type of PPRC volume and its status. Possible types: • primary—primary volume of a PPRC pair. • secondary—secondary volume of a PPRC pair. • data—data bridge volume of a PPRC bridge pair. • status—status bridge volume of a PPRC bridge pair. • blank—not a member of a PPRC pair. Possible statuses: • duplex—PPRC pair is fully synchronized. • pending—PPRC pair is in the process of being synchronized; data updates made to the primary volume are being written to the secondary volume.	None

Field	Definition	Calculation
	• simplex—used for data or status bridge volumes only; volume is not a member of a PPRC bridge pair. • suspended—PPRC pair is suspended; transfer of updates from the primary to the secondary volume has been halted.	
Device Composition: (displayed only if the -detail parameter was used on the command)		
FDID	FDID of the parent (index 0) device.	None
CHILD FDIDS	Child FDIDs that make up the parent FDID, separated by commas. If the device is not a SCSI larger LUN, entry is a dash (–).	None
Device Capacity: (displayed only if the -detail parameter was used on the command)		
FDID	FDID specified in the request.	None
FUNCTIONAL CAP(GB)		
AVAILABLE	Capacity available on the device, in GB.	None
STORED	Capacity of data stored on the device, in GB.	**For SCSI devices:** logical block size * logical blocks per track * number of mapped tracks **For CKD devices:** bytes per track * number of mapped tracks
PHYSICAL CAP USED (GB)		
SHARED	Physical capacity of shared data on the device, in GB.	TOTAL PHYSICAL CAP USED + UNIQUE PHYSICAL CAP USED
UNIQUE	Physical capacity of unique data on the device, in GB.	None
TOTAL	Total capacity used on the device, in GB.	None

Field	Definition	Calculation
PHYSICAL CAP		
CYLS	Cylinder capacity of the device.	None
LBLKS	Logical block size of the device.	None
LBPT	Logical blocks per track for the device.	None
COMP RATIO	Calculated compression ratio for the device (entry is rounded to one decimal point).	functional capacity stored / total physical capacity used
Avg Comp Ratio	Average compression ratio for all specified devices. Displayed only if the -detail parameter was used on the command and the * wildcard is used to specify all devices on the subsystem.	None
Sys Compr Ratio	Compression ratio for the entire SVA subsystem. Displayed only if the -detail parameter was used on the command and the * wildcard is used to specify all devices on the subsystem.	Tot. functional capacity stored / net capacity load (in bytes)
PPRC Information (displayed only if the -pprcinfo parameter was used on the command)		
FDID	FDID that is part of a PPRC pair.	None
---PRIMARY---		
ID	ID of the primary PPRC volume. Defaults to the lower byte of the primary PPRC volume's FDID.	None
SSID	SSID of the primary SVA subsystem where the primary volume exists.	None
SERIAL NUM	Serial number of the primary SVA subsystem where the primary volume exists.	None
---SECONDARY---		
ID	Target ID assigned to the secondary PPRC volume.	None
SSID	SSID of the secondary SVA subsystem where the secondary volume exists.	None

Field	Definition	Calculation
SERIAL NUM	Serial number of the secondary SVA subsystem where the secondary volume exists.	None
FIRST CYL NOT SYNC	PPRC first cylinder not synchronized.	
LAST CYL NOT SYNC	PPRC last cylinder not synchronized.	
STATE DESC	Current state of the PPRC pair.	
QUERY FLAGS	Indicates exceptional conditions.	None

displaydrivemodule

Field	Description	Calculation
Subsystem Name:	SVA subsystem name specified in the request.	None
Subsystem Model:	Model number of the subsystem.	None
Frame Serial Number:	Serial number of the subsystem.	None
Site Name:	Name of the site.	None
Site Location No.:	StorageTek-assigned site location number.	None
Capacity:	Total SVA subsystem capacity, in GB.	None
-DEVICE LOCATION-UNIT TRAY SLOT	The unit, tray, and slot location of the drive module within the SVA subsystem.	None
SERIAL NUMBER	Serial number of the drive module.	None
DEVICE STATUS	Two-letter status code of the drive module. See Appendix A., “Drive Module Status Codes” for possible values.	None
ARRAY	Array ID to which the drive module is assigned.	None
CAPACITY (GB)	Total capacity of the drive module, in GB.	None
---DEVICE TYPE---		
CLASS	Device class—indicates the type of drive. Possible values: • ESDI • SCSI-strapped • SCSI-unstrapped • 4GB • 9GB • 18GB • 36GB	None

Field	Description	Calculation
SUB-CLASS	Device subclass—indicates the capacity of the drive. Possible values: • 1.6GB, ESDI, SCSI-strapped • 2.0GB, SCSI-unstrapped • 4GB • 9GB • 18GB • 36GB	None

displayiointerface

Field	Description	Calculation
Subsystem Name:	SVA subsystem name specified in the request.	None
Subsystem Model:	Model number of the subsystem.	None
Frame Serial Number:	Serial number of the subsystem.	None
Site Name:	Name of the site.	None
Site Location No.:	StorageTek-assigned site location number.	None
No. of Interfaces:	Total number of I/O interfaces installed on the SVA subsystem.	None
IFID	I/O interface specified in the request.	None
NAME	Name assigned to the I/O interface.	None
STATUS	Current status of the I/O interface. Possible values: • ENABLED • DISABLED	None
TYPE	Type of I/O interface. Possible values: • FIBRE • PARALLEL (OEMI) • SCSI • SERIAL (ESCON)	None
SPEED	Transfer rate of the I/O interface, in MB per second. Possible values: • 3.0 • 4.5 • 20.0 • 40.0 • 100.0	None

Field	Description	Calculation
RANGE	Number of addresses on the I/O interface. Displayed for serial, parallel, or SCSI interfaces only. Possible values: • for parallel: 8, 16, 32, 64, 128, 256 • for serial (ESCON): 8, 16, 32, 64, 128, 256, 512, 1024 • for SCSI: 0–120	None
BASE	Lowest interface address on the I/O interface.	None
BFDID	Base FDID (functional device ID) for the I/O interface; identifies the path between the base address and the functional device. Used for serial, parallel, or SCSI interfaces only.	None
LINK USAGE	Used only if this I/O interface is defined as a PPRC path. Possible values: • PRIMARY–This SVA subsystem is the primary subsystem for the PPRC path. • SECONDARY—This SVA subsystem is the secondary subsystem for the PPRC path. • LOCAL—This is a Power PPRC direct path. • REMOTE (WAN)—This is a Power PPRC WAN path. • blank—This is not a PPRC path.	None
WORLD WIDE NAME	Unique fibre channel world-wide node name assigned to the I/O interface. Displayed for FC-PLDA interfaces only.	None

Field	Description	Calculation
DOMAIN	Domain assigned to the I/O interface. Displayed for FC-PLDA interfaces only.	None
ADDRESS	Fibre channel address assigned to the I/O interface. Displayed for FC-PLDA interfaces only.	None
PORT	Unique fibre channel port assigned to the I/O interface. Displayed for FC-PLDA interfaces only.	None
PPRC PATH Information: (displayed only if the -pprcinfo parameter is used on the command)		
IFID	I/O interface ID specified in the command.	None
INTERNAL IFID	I/O interface ID internal to the SVA subsystem.	None
PRIMARY SSID	SSID (subsystem ID) of the PPRC path's primary SVA subsystem.	None
SECONDARY VCU	Virtual control unit (VCU) number on the PPRC path's secondary SVA subsystem.	None
SECONDARY SSID	SSID (subsystem ID) of the PPRC path's secondary SVA subsystem.	None
SECONDARY SSNAME	Name of the PPRC path's secondary SVA subsystem.	None
SECONDARY SERIAL NUM	Serial number of the PPRC path's secondary SVA subsystem.	None
LINK DEST	Destination logical address. Entry is 00, unless the connection is going through an ESCON director.	None
STATUS	Current status of the PPRC path. Possible values: • Established • Blank, if not established	None
FLAGS	Indicates exceptional conditions.	None

displayncaplodpct

Field	Description	Calculation
SUBSYS NAME	SVA subsystem name specified in the request.	None
DISK ARRAY CAPACITY(GB)	Total subsystem capacity, in GB.	None
NET CAPACITY LOAD (GB)	Total subsystem space that is used (not free), in GB.	total subsystem capacity - free subsystem capacity
NCL %	Percentage of subsystem space that is used (not free).	((total subsystem capacity - free subsystem capacity) / total subsystem capacity) * 100
COLLECTED FREE SPACE %	Percentage of array cylinders in the SVA subsystem that are free (that is, the total space that can be written to).	collected free subsys. capacity / total subsystem capacity
UNCOLLECTED FREE SPACE %	Percentage of space occupied when a functional track has been rewritten to a new location in the SVA subsystem.	(free subsys capacity - collected free subsys capacity) / total subsys capacity

displayserver

Field	Description	Calculation
Version:	Current release level of the SVAA server.	None
Description:	User-assigned description of the server.	None
Trace Status:	Indicates whether tracing is currently enabled.	None
Installed Features:	List of installed SVAA features.	None
Maintenance Level:	Current SVAA server patch level installed.	None
Host:		
Name:	User-assigned name for the SVAA server.	None
TCP/IP Port:	Unique TCP/IP port number assigned to this SVAA server.	None
Host OS Level:	Current host operating system release level.	None

displaysubsystem

Subsystem Name:	SVA subsystem name specified in the request.	None
Subsystem Model:	Model number of the subsystem.	None
Frame Serial Number:	Serial number of the subsystem.	None
Site Name:	Name of the site.	None
Site Location No.:	StorageTek-assigned site location number.	None
General Information:		
Subsystem Identifiers:	List of (up to 16) SSIDs assigned to the subsystem.	None
Number of Spares:	Number of spare drive modules available on the subsystem.	None
Number of Arrays:	Number of subsystem arrays that have been configured.	None
Default Array Size:	Default array size assigned to the subsystem.	None
Number of:		
SCSI Domains:	Maximum number of SCSI domains supported for addressing by this SVA subsystem. Possible values: 0–16.	None
SCSI Targets per Domain:	Maximum number of SCSI targets supported per domain address by this SVA subsystem. Possible values: 0–16.	None
SCSI LUNs per Target:	Maximum number of SCSI LUNs supported per target address by this SVA subsystem. Possible values: • 0 • 8 • 256	None
Licensed Functional Devices:	Maximum number of functional devices licensed to this SVA subsystem. Possible values: 256–65,535.	None

Cache Size:	Base cache size, in MB.	None
NVS Size:	Non-volatile storage capacity, in MB.	None
Maximum Number of:		
Channel Interface Supported:	Maximum number of channel interfaces the SVA subsystem can support.	None
Physical Devices Supported:	Maximum number of drive modules the SVA subsystem can support.	None
Virtual Devices Supported:	Maximum number of functional devices the SVA subsystem can support.	None
Arrays Supported:	Maximum number of arrays the SVA subsystem can support.	None
Physical Capacity Control Limit:	Physical capacity control limit for the SVA subsystem, in GB. An entry of 0 indicates the subsystem has no limit.	None
Virtual Volumes per VCU/SSID:	Number of functional devices per VCU/SSID. Possible values: 64 or 256.	None
SCSI SIMS Status:	Indicates whether the SVA subsystem passes SIM messages to a SCSI-attached host. Possible values: • Disabled • Enabled	None
Remote Operator Panel Status:	Indicates whether a remote operator panel is enabled. Possible values: • Disabled • Enabled	None
Functional Capacity:	Total capacity of functional devices on the SVA subsystem, in GB.	None
Installed Features/Options:	List of features installed on the SVA subsystem.	None

Revision Level Information:		
Component Id:	SVA subsystem software FRU ID. Up to eight entries are possible.	None
Level:	Revision level of the component.	None

queryadministratorgroup

Field	Description	None
`SVAA administrator group:`	Current SVAA administrator group. Blank if none is assigned.	None
`SVAA PPRC administrator group:`	Current PPRC administrator group. Blank if none is assigned.	None

querycache

Field	Description	Calculation
SUBSYSTEM	SVA subsystem name specified in the request.	None
CACHE	Cache type. Possible values: • array • functional • interface • physical • subsystem	None
AREA	Cache area. Possible values: • performance • properties • space utilization • status	None
REFRESH RATE (SECS)	Current refresh rate, in seconds.	None
ELEMENTS	Total number of elements in this cache.	None

queryclienttrace

Field	Description	Calculation
`Trace side:`	Indicates whether the trace is being performed on the client (CLI) or the SVAA server.	None
`Identifier:`	System-generated trace ID.	None
`Level(s):`	Trace levels specified in the `startclienttrace` command.	None
`Log file:`	File to which the trace log is being written.	None
`Traced class(es):`	Classes specified in the `startclienttrace` command.	None

queryclientlogentries

Field	Description	Calculation
	Date when the log entry was generated.	None
	Time when the log entry was generated.	None
	Text of the log entry.	None
.	Severity level of the log entry.	None

queryclienttrace

Field	Description	Calculation
`Trace side:`	Indicates whether the trace is being performed on the client (CLI) or the SVAA server.	None
`Identifier:`	System-generated trace ID.	None
`Level(s):`	Trace levels specified in the `startclienttrace` command.	None
`Log file:`	File to which the trace log is being written.	None
`Traced class(es):`	Classes specified in the `startclienttrace` command.	None

queryconnections

Field	Description	Calculation
ID	System-generated connection ID.	None
USERID	Userid connected.	None
HOST	Host the user is logged into.	None
PERM	Permission level of the user. Displayed only if the -detail parameter was used on the command.	None
LOGTIME	Date and time when the user logged onto the SVAA server. Displayed only if the -detail parameter was used on the command.	None

querypermissions

Field	Description	Calculation
USERID	Userid specified in the request. Displayed only if the -user parameter was used on the command.	None
GROUPID	Group ID specified in the request. Displayed only if the -group parameter was used on the command.	None
PERM	Permission level(s) assigned to the user or group.	None

queryrequests

Field	Description	Calculation
REQUEST ID	Identifier assigned to the requesttcr.	None
USER	Userid that initiated the request.	None
CONNECTION ID	System-generated connection ID for the request.	None
HOST	Host from which the request was initiated.	None
STARTTIME	Date and time when the request was initiated.	None

querysecuritymode

Field	Description	Calculation
`SVAA security mode:`	SVAA security mode currently in effect.	None

queryserverlogentries

Field	Description	Calculation
	Date when the log entry was generated.	None
	Time when the log entry was generated.	None
	Text of the log entry.	None
	Severity level of the log entry.	None

queryservertrace

Field	Description	Calculation
`Trace side:`	Indicates whether the trace is being performed on the client (CLI) or the SVAA server.	None
`Identifier:`	System-generated trace ID.	None
`Level(s):`	Trace levels specified in the `startservertrace` command.	None
`Log file:`	File to which the trace log is being written.	None
`Traced class(es):`	Classes specified in the `startservertrace` command.	None

querysubsystem

Field	Description	Calculation
	List of SVA subsystems accessible to the local SVAA server.	None

queryversion

Field	Description	Calculation
	Release and patch level of the SVAA server software installed on this server.	None

reportevents

Field	Description	Calculation
Subsystem Name:	SVA subsystem name specified in the request.	None
Subsystem Model:	Model number of the subsystem.	None
Frame Serial Number:	Serial number of the subsystem.	None
Site Name:	Name of the site.	None
Site Location No.:	StorageTek-assigned site location number.	None
Drain Identifier:	System-assigned ID for this drain event.	None
Initiating Node ID:	Host from which the drain was initiated.	None
Started:	Date and time when the drain was initiated.	None
Initiating User ID:	Userid that initiated the drain.	None
Completed:	Date and time when the drain was completed. Blank if the drain is ongoing.	None
Completion Node ID:	Host node ID to which notice of drain completion is to be sent (used for S/390 hosts only).	None
Percent Complete:	Drain percentage completed.	None
User ID:	Userid to which notice of drain completion is to be sent (used for S/390 hosts only).	None
U.T.S	Unit, tray, and slot of the drive module being drained.	None
Serial Number	Serial number of the drive module.	None
Status	Status of the drive module.	None

Glossary

A

array—A group of drive modules used collectively to achieve data redundancy and/or improved performance. An array consists of either 7 or 15 drive modules (15 only for V960 or V2X subsystems). There can be up to 8 arrays in a subsystem.

array device—The disk devices that are logically grouped together when a `formarray` command is issued at the local operator panel or from SVAA.

B

back-end storage—The data storage portion of a storage system. In the Shared Virtual Array, this is the disk array on the subsystem.

C

cache—Solid-state, random-access memory that is located in a controller. The cache retains frequently used data for faster access. In the Shared Virtual Array, all data access is through cache.

channel interface—See "I/O interface".

CKD device—A storage device that uses the "count-key-data (CKD)" format.

CLI (command line interface)—In SVAA, a user interface on the Windows host system used to issue commands for configuring and monitoring the SVAA server and the SVA subsystem.

client—the CLI (command line interface) is a client of the SVAA server.

count-key-data (CKD)—A recording format that writes variable-length records. Each record consists of 1) a count field, which specifies the length of the (optional) key field and the data field of the record, 2) the (optional) key field, and 3) a data field. The first record on each track contains a fourth field, home address. Contrast with "fixed-block architecture (FBA)".

D

disk array—The logical grouping of drive modules on a Shared Virtual Array.

domain—An SVA subsystem addressing scheme, prefixed to SCSI target and LUN addresses, that extends the number of addressable devices from SCSI-attached hosts.

drain—The Shared Virtual Array process that gradually moves data stored on a drive module or a disk array to other devices. Drains allow for the non-disruptive de-installation of a device.

drive module—One of the inexpensive disks comprising the storage in an SVA subsystem. A physical device.

dynamic configuration—A Shared Virtual Array feature that allows the I/O interfaces and up to 1024 (in the case of V960 or earlier subsystems) or 4096 (in the case of V2X subsystems) functional devices to be defined and/or altered. The functional configuration of a Shared Virtual Array can be determined by user requirements, rather than available physical devices.

E

ECAM (extended control and monitoring facility)—The communications protocol that permits communication between SVAA and the SVA subsystem. See also "ECAM device" and "privileged ECAM device".

ECAM device—A functional device over which ECAM messages are exchanged between the Shared Virtual Array disk array controller and the SVAA server.

ESCON—Enterprise Systems Connection. An IBM standard for connecting peripheral devices to a host system via fibre-optic cables.

ESCON channel—A channel that uses fiber-optic (ESCON) cables to transmit data between the host and the disk array controller.

F

FDID—Functional device ID. The ID for a functional device as it is known to the SVA subsystem and the SVAA server. FDIDs range from 0 to 3FF, hexadecimal, for V960 and earlier subsystems; 0 to FFF, hexadecimal, for V2X subsystems. See “functional device”.

fixed-block architecture (FBA)—A recording format in which every track of the device is formatted with a fixed number of fixed-length records (generally called sectors), each of which contains an identifier (ID) field and a data field. Contrast with “count-key-data (CKD)”.

functional—The term used to describe the Shared Virtual Array system interface as viewed by the host, application, and users. This interface appears as a SCSI or 3990-3 system interface.

functional device—The disk device image viewed by the host operating system.

I

index 0 device—The parent device in a SCSI larger LUN device. Child devices are devices 1–n.

initiator—A device that begins a SCSI transaction by issuing a command to another device (the target) to perform a task. Typically, a SCSI host adaptor is the initiator. Contrast with “target”.

I/O interface—The Shared Virtual Array circuitry that attaches to the host system. Can be fibre or ESCON.

K

known SVA subsystem—A subsystem to which the local SVAA server has access.

L

larger LUN (LLUN) device—See “SCSI larger LUN (LLUN) device”.

local operator panel (LOP)— A user interface, located on the front door of the SVA subsystem, used to control and configure the SVA subsystem.

logical unit number (LUN)—A method to expand the number of devices on a SCSI bus. LUNs address up to eight devices at each SCSI ID.

LOP—See “local operator panel (LOP)”.

LUN—See “logical unit number (LUN)”.

M

MAT partition—The SVA subsystem partition consisting of drive modules that are not yet available for storing user data. Drive modules are automatically members of the MAT partition when they are first physically inserted in the SVA subsystem or when they have been drained of data.

N

NCL—See “net capacity load (NCL)”.

net capacity load (NCL)—A percentage of the total number of sectors used to store user data. It is based on physical capacity used. This number is 2KB times the number of physical sectors actually used to store user data, not including redundancy data.

P

parallel channel—A channel that uses bus-and-tag cables to transmit data between the mainframe and the disk array controller.

partition—In the SVA subsystem, the logical separation of devices, arrays, or groups of arrays to allow different modes of operation. The SVAA for Windows server supports the following partitions: MAT, Production, Spares, and Unavailable.

physical device—See “drive module”.

point-in-time reports—SVA subsystem reports generated through the SVAA WBI.

PPRC bridge pair—A primary bridge volume and its associated secondary bridge volume. There are both PPRC data bridge pairs and PPRC status bridge pairs.

PPRC data bridge volume—Used by Power PPRC as a staging area for PPRC data.

PPRC link—The physical ESCON connection between two SVA subsystems.

PPRC pair—A primary volume and its associated secondary volume. PPRC pairs are established and de-established using SVAA commands.

PPRC path—The logical connection between two virtual control units (VCUs) used by PPRC.

PPRC primary volume—A device that is being mirrored under PPRC control. A primary volume can be copied to only one secondary volume.

PPRC secondary volume—A device that is receiving mirrored data from a primary volume.

PPRC status bridge volume—Used by Power PPRC for acknowledgments that PPRC data has been successfully transmitted and received.

privileged ECAM device—One of the only devices that SVAA can use to send messages to the SVA subsystem to request a change in the subsystem's state. Such messages include those that alter the subsystem configuration or start a drain.

At least one privileged ECAM device must be defined in each SVA subsystem.

Production partition—The SVA subsystem partition consisting of drive modules used for storing and retrieving user data.

S

SCSI—See "Small Computer Systems Interface (SCSI)".

SCSI bus—A pathway for data that conforms to the SCSI standard.

SCSI device—A device that conforms to the SCSI standard.

SCSI ID—The unique address of a SCSI device. SCSI IDs range from 0 to 7 for 8-bit systems, 0 to 15 for 16-bit systems, and 0 to 31 for 32-bit systems.

SCSI larger LUN (LLUN) device—A device consisting of one or more functional devices, viewed by the host system as a single logical device. SCSI larger LUN devices can be virtually any size.

serial channel—See "ESCON channel".

Shared Virtual Array (SVA)—An online, random access disk array storage system composed of disk storage and control unit combined into a single frame.

Shared Virtual Array Administrator (SVAA)—A server-based product providing SVA subsystem configuration and administration functions. Runs on UNIX, Windows, and S/390 platforms.

Shared Virtual Array Console (SVAC)—A GUI interface that connects to an SVAA server and provides SVA subsystem configuration, administration, and reporting functions.

slot—The physical location of a Shared Virtual Array drive module.

Small Computer Systems Interface (SCSI)— An I/O interface with a standard, device-independent protocol that allows many different peripheral devices to be attached to a host system.

space release—An SVAA facility that informs an SVA subsystem that the space in a SCSI partition can be released because the information in the partition is no longer needed.

Spares partition—The SVA subsystem partition consisting of drive modules that are physically installed but not logically associated with an array. Drive modules in the Spares partition are used to form production arrays, to reconstruct failed disks, and to receive data from drive modules being drained.

subsystem—A secondary or subordinate system, usually capable of operating independently of a controlling system.

SVA—See "Shared Virtual Array (SVA)".

SVA subsystem—See "Shared Virtual Array (SVA)".

SVAA—See "Shared Virtual Array Administrator (SVAA)".

SVAA server—See "Shared Virtual Array Administrator (SVAA)".

SVAA server cache—A cache of information maintained by the SVAA server for each SVA subsystem to which it has access. Includes information about the overall SVA subsystem, its I/O interfaces, physical devices, and functional devices.

SVAC—See "Shared Virtual Array Console (SVAC)".

T

target—A SCSI device that executes a command from another device (the initiator) to perform a task. Typically, a SCSI peripheral device is the target. Contrast with "initiator".

tray—The physical packaging of eight drive modules in a Shared Virtual Array.

U

Unavailable partition—The SVA subsystem partition consisting of drive modules that are not available for use in an array. This includes drive modules that have failed or that are not yet physically installed.

unknown SVA subsystem—A subsystem to which the local SVAA server does not have access.

V

virtual storage architecture—The innovative storage architecture used by Shared Virtual Array storage to extend the capabilities of traditional online storage.

VSAM volume data set (VVDS)—Mainframe term referring to the volume data set that describes the characteristics of VSAM (virtual storage access method) or SMS (storage management subsystem) managed data sets residing on the volume. There is one, and only one, VVDS for each volume containing either SMS-managed data sets or VSAM data sets cataloged in an ICF (integrated catalog facility) catalog.

volume table of contents (VTOC)—Mainframe term referring to the volume data set that describes the characteristics of each data set stored and the space remaining available for use on the volume. There is one VTOC per volume.

W

WBI—Graphical user interface (GUI) to the SVAA server that can be accessed through a Web browser.

Index

A

C

D

E

F

G

H

I

L

Q

R

S

T

U

V

W

NEED MORE INFORMATION?

www.storagetek.com

ABOUT STORAGETEK

Storage Technology Corporation (NYSE: STK) is a $2 billion global company that enables businesses, through its information lifecycle management strategy, to align the cost of storage with the value of information. The company's innovative storage solutions manage the complexity and growth of information, lower costs, improve efficiency and protect investments. For more information, visit www.storagetek.com, or call 1.800.275.4785 or 01.303.673.2800.

WORLD HEADQUARTERS

Storage Technology Corporation
One StorageTek Drive
Louisville, Colorado 80028 USA
1.800.678.4430 or 01.303.673.4430

StorageTek equipment is manufactured from new parts, or new and used parts. In some cases, StorageTek equipment may not be new and may have been previously installed. Regardless, StorageTek's standard warranty terms apply, unless the equipment is specifically identified by StorageTek as "used" or "refurbished."

Replacement parts provided under warranty or any service offering may be either new or equivalent-to-new, at StorageTek's option. Specifications/features may change without notice.