Part Number: MP4003C

9500
Shared Virtual Array

Planning, Implementation and Usage

StorageTek
June, 2000

Information contained in this publication is subject to change. In the event of changes, the publication will be revised. Comments concerning its contents should be directed to the address shown below.

StorageTek
One StorageTek Drive
Louisville, Colorado
80028-2201

Contents

This page intentionally left blank.

Preface

Notices

United States FCC Compliance Statement

The following is the compliance statement from the Federal Communications Commission:

Note: This equipment has been tested and found to comply to the limits for Class A digital devices pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his or her own expense.

Some of the cables used to connect peripherals must be shielded and grounded as described in the installation manual. Operation of this equipment with the required cables that are not shielded and correctly grounded may result in interference to radio and TV reception.

Changes or modifications not expressly approved by StorageTek could void the user's authority to operate the equipment.

Industry Canada Compliance Statement

This digital apparatus does not exceed the Class B limits for radio noise emissions from digital apparatus set out in the radio interference regulations of the Canadian Department of Communications.

Le présent appareil numerique német pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe B prescrites dans le Reglément sur le brouillage radioélectrique édicté par le ministére des Communications du Canada.

Japanese Compliance Statement

The following is the compliance statement from Japan:

この装置は、クラスＡ情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。 VCCI-A

Note: This equipment is in the Class A category information technology equipment based on the rules of Voluntary Control Council For Interference by Information Technology Equipment (VCCI). When used in a residential area, radio interference may be caused. In this case, user may be required to take appropriate corrective actions.

Consequently, when used in residential area or in an adjacent area thereto, radio interference may be caused to radios and TV receivers, etc. Read the instructions for correct handling.

Taiwan Warning Label Statement

The following is the warning label statement from Taiwan, R.O.C.:

警告使用者：這是甲類的資訊產品，在居住的環境中使用時，可能會造成射頻干擾，在這種情況下，使用者會被要求採取某些適當的對策

About This Book

This book is designed to help you tailor a successful Shared Virtual Array (SVA) conversion and installation plan for your environment. It provides specific recommendations for configuring and using an SVA subsystem in different operating environments. It does not explicitly address physical planning considerations, but instead focuses on software support and utilization of the subsystem. Refer to the 9500 *Shared Virtual Array Introduction* for information about SVA features and functionality and to the *9500 Shared Virtual Array Physical Planning Guide* for physical planning and hardware support information.

Who Should Read This Book

The audience for this book includes system programmers, storage administrators, and performance and capacity analysts who have a basic understanding of Shared Virtual Array features and functionality.

Do You Have The Complete Manual?

This book consists of page 1 through page 156.

Note: The last two pages are the reader comment form and its mailer. These pages may not be there if someone sent the comment form to StorageTek.

Trademarks

Related Documents

The documents listed below comprise the complete SVA 9500 set. See "Viewing and Ordering Documents" (below) to obtain documents through available distribution channels.

Shared Virtual Array (SVA) Subsystem

Documents below are available online, on CD-ROM, and as bound books.

- *9500 Shared Virtual Array Introduction (MP4001B).*
- *9500 Shared Virtual Array Operation and Recovery (MP4002B)*
- *9500 Shared Virtual Array Planning, Implementation and Usage (MP4003B)*
- *9500 Shared Virtual Array Physical Planning (MP4004B)*
- *9500 Shared Virtual Array Reference (MP4005B)*
- *9500 Shared Virtual Array System Assurance (MP4006B)*
- *Peer to Peer Remote Copy Configuration Guide (MP4007A)*

Shared Virtual Array Administrator (SVAA) for OS/390

Documents below are available online and on CD-ROM.

- *SVAA for OS/390 Configuration and Administration (PN 3112905xx)*
- *SVAA for OS/390 Reporting (PN 3112906xx)*
- *SVAA for OS/390 Messages and Codes (PN 3112907xx)*
- *SVAA for OS/390 Installation, Customization, and Maintenance (PN 3112908xx)*

SnapShot for OS/390

Documents below are available online and on CD-ROM.

- *SnapShot for OS/390 User's Guide (PN 3112912xx)*
- *SnapShot for OS/390 Installation, Customization, and Maintenance (PN 3112913xx)*

Shared Virtual Array Administrator (SVAA) for Solaris

Documents below are available online and on CD-ROM.

- *SVAA for Solaris User's Guide (PN 3112909xx)*
- *SVAA for Solaris Messages (PN 3112910xx)*
- *SVAA for Solaris Installation (PN 3112911xx)*
- *SVAA for Solaris Quick Start Guide (PN 3112971xx)*

SnapShot for Solaris

Documents below are available online and on CD-ROM.

- *SnapShot for Solaris User's Guide (PN 3112914xx)*
- *SnapShot for Solaris Quick Start Guide (PN 3112915xx)*

Shared Virtual Array Console (SVAC) for Windows NT

The document below is available online and on CD-ROM.

- *SVAC for Windows NT Quick Start Guide (PN 3112993xx)*

Viewing and Ordering Documents

Viewing the Documents Online

SVA 9500 documents can be viewed (and printed on your printer - these are PDF files) at the Customer Resource Center website at:

`http://www.support.storagetek.com/wwcss/SilverStream/Pages/pgCRCHome.html`

Click on ‘Disk Subsystems’, then ‘Docs’ under the 9500 section.

Note: A password is required. You may obtain the password from a StorageTek marketing representative.

Ordering Documents

SVA 9500 documents are available on CD-ROM, and bound book. Consult a StorageTek marketing representative to order the various manuals relating to the 9500.

CD-ROM

- Customer hardware documents: a CD-ROM of SVA 9500 customer documents is available by requesting the *9500 Customer Documentation, PN MP-9500x*.
- Software documents: a CD-ROM of SVA 9500 software documents is available by requesting *SVA Software Publications, PN 311295301-xx*.

Bound Books

Individual bound books of SVA 9500 documents are available through Software Manufacturing Distribution (SMD); request by document title and/ or part number.

Other Documents

The following IBM documents may also assist you in using SVA 9500:

- *Planning For IBM Remote Copy SG24-2595-xx*
- *Remote Copy Administrator’s Guide and Reference SC35-0169-xx*

History of Changes

Rev A - Initial release November 1999.

Rev B - First reissue. December 1999.

- Changed the page numbering scheme to match that used by the PDF files.
- Enhance the PPRC lockout drawing.

Rev C - Second reissue. June 2000.

- Changed channel matirx to reflect ICF cards.
- Added ICF HBA and SCSI HBA tables to Chapter 5.
- Other minor changes and updates.

Chapter 1 Configuration Planning

This chapter discusses the Shared Virtual Array (SVA) facilities that must be carefully planned prior to subsystem installation.

Selecting an SVA Subsystem Configuration

The process of selecting a DASD hardware configuration has always been complex. Factors such as I/O rates, cache hit rates, the amount of data to be stored, and response times influence the decision-making process. Typically, storage managers make decisions that optimize a specific configuration to meet either capacity requirements or performance requirements.

Anticipate your data activity when planning the SVA subsystem configuration. The Direct Access Device Activity Report shows device utilization for each volume. When migrating smaller capacity volumes to a single volume on the SVA, it is recommended that the combined device utilization of the two source volumes not exceed 40%. If the Direct Access Device Activity Report shows a non-zero value in the device reserve percentage field (for any of the source volumes), the device utilization should be calculated as follows:

Dev utilization % = SIO/SSCH rate X (Pend time + Disc time + Conn time)/10

Note: Peak access density is measured over the peak hour of a normal business day.

Physical Considerations

Installing a Shared Virtual Array subsystem requires only a few physical planning decisions, such as channel attachment options, redundant power supplies, and extended operator panel attachment. Information about each of these physical planning decisions follows.

These physical planning considerations are discussed during the system assurance process. For complete SVA physical planning requirements, refer to the *9500 Shared Virtual Array Physical Planning Guide*.

For open system attachments, refer to Appendix D “SCSI Open System Attachment” on page 117.

Virtual Device Considerations

Because of the different device addressing structures in a 9500 (1024 vs. 256 available device addresses), when migrating data to this model from some earlier models[1] you must make a concurrent IOCP GEN change. If you do not make the required IOCP GEN change, you will not be able to locate all 256 devices previously used.

1. SVA 9393-T82, as well as early RAMAC Virtual Array models, were limited to 256 virtual device addresses.

Channel Attachment Options

SVA attaches to all IBM 370-equivalent data streaming channel architectures, including IBM or compatible 30XX, 43XX, 9370, and ES/9000 series CPUs. SVA supports:

- SCSI channels, via the SVA Extended SCSI Attach (XSA) Feature, at rates of up-to 20 megabytes per second per fibre connection.
- ESCON-compatible FX channels with data transfer rates of 20 megabytes per second.
- Fibre channel (long wave, short wave, and copper).

Channel transfers are never synchronized with array transfers. Therefore, the timing problems associated with channel extenders for traditional DASD are not a consideration for an SVA subsystem.

Redundant Power Supplies

The Shared Virtual Array design provides comprehensive fault tolerance, high availability, and hardware failure isolation. For true AC redundancy, the two AC power supply systems should receive power from two independent power circuits, with independent circuit breakers that are not likely to fail simultaneously. Independent power circuits preclude a single point of failure for the subsystem's power sources.

Refer to the *Shared Virtual Array Physical Planning Guide* for detailed information on power configuration planning.

Functional Considerations

The Shared Virtual Array's architecture also requires some functional configuration considerations, which are described in the following sections.

Note: You can perform the configuration tasks and sub-tasks from the local operator panel or from the host using the Shared Virtual Array Administrator (SVAA) product or its predecessor, the Iceberg Extended Facilities Product (IXFP). Because of its flexibility, we recommend performing most of the configuration sub-tasks using SVAA / IXFP configuration services.

Functional Configuration

Before configuring an SVA subsystem from the host operating system, you must first establish a minimum functional configuration from the local operator panel.

The initial SVA subsystem configuration is identified during the system assurance process. Then, as part of the subsystem installation, a StorageTek CSE sets up the minimum subsystem configuration. (Refer to the *Shared Virtual Array Operation and Recovery* for more information about the initial subsystem configuration.) Next, a trained customer representative or StorageTek representative modifies the minimum configuration, which includes defining the functional configuration of the subsystem.

The SVA 9500 allows the user to define:

- up to 1024 functional 3380-J, 3380-K, 3390-3 volumes, or
- up to 341 3390-9 volumes

regardless of the physical capacity of the attached disk arrays.

Note: For SCSI attach, only 3390-3 and 3390-9 devices are supported.

The only restriction is that the physical capacity of the attached disk arrays supports the actual net capacity load (NCL) of the subsystem. (The NCL is the physical disk array capacity required to store all of the compressed functional tracks actually occupied by data in a functional configuration.)

Functional 3380 or 3390 volumes may be intermixed freely in an SVA subsystem configuration; however:

- 3380-compatibility mode for 3390 volumes is not supported
- SYSGEN, IOCP, and other host system constraints do apply.

Note: Implementing 3380 and 3390 functional volumes requires the same operating system maintenance level required to implement actual 3380 and 3390 volumes, and 4-digit addressing.

Missing-Interrupt Handler (MIH)

Although SVAA adjusts the missing-interrupt handler (MIH) for SVA devices automatically, it is important that you know the minimum value to which you can safely set the MIH manually.

SVA's internal error recovery procedures require that the MIH time-out value be set to five minutes (the default MIH time-out value is 15 seconds); however, the time-out value can vary from one SVA subsystem to another. Known I/O timer or MIH values are:

- **For OS/390:** If installing SVAA, set the MIH to four minutes twenty seconds.
- **For Solaris:** If installing SVAA, set the I/O timer to 270 seconds.
- ***For MVS*:** If installing SVAA, you can elect (by default) to have SVAA set the MIH time-out value automatically.
- ***For VM*:** You must set the MIH time-out value manually; you cannot elect to have SVAA set the MIH time-out value automatically. See your CSE for the appropriate value.

Note: For systems not listed, set the I/O timer or MIH to 270 seconds.

This change forestalls the invocation of the ERPs for the missing interrupt handler in normal error recovery situations.

Note: Adjust applications running on your system that detect missing interrupts and that are independent of the system MIH setting to allow a five-minute missing interrupt time-out value.

Disk Array Definition

In a Shared Virtual Array, 15 or 7 physical devices are organized into a logical group. Within the group, user data is recorded on identically addressed tracks on all but two of the devices. The identically addressed tracks on the other two devices are used to store the two levels of redundancy data generated by the subsystem.

Note: The redundancy data is dynamically placed across the devices in the array, so that all of the devices in the array contain user data and redundancy data.

There are two configuration options for a Shared Virtual Array: a 15-device array or a 7-device array.

- In a 15-device array, which is referred to as a 13+2+1 array, one physical device in the array is reserved as a spare device and is globally available to the subsystem.
- In a 7-device array, which is referred to as a 5+2+1 array, one physical device in the array is reserved as a spare device and is globally available to the subsystem.

Both configurations support dual redundancy.

The recommended and default disk array configuration is the 15-device array option. This configuration provides the maximum data capacity with an adequate number of spare drives for data recovery.

While it is not recommended, the SVA also supports 5+2+1 and 5+2 redundancy groups (five data drives + two redundancy/parity drives + one spare). This configuration provides improved storage capacity granularity.

Array Partitions

Part of the SVA configuration process includes assigning physical devices to an array partition. A physical device may be associated with one of four partitions:

Spares Partition

Physical devices may be assigned to the Spares partition. Drives in the Spares partition are used to form test and production arrays, to reconstruct failed disks, and to receive data from a draining device.

Production Partition

Physical devices in the Production partition comprise the arrays that store user data. These arrays are formed from the physical devices in the Spares partition.

Unavailable Partition

The Unavailable partition includes any device slots that have not had devices installed, any powered off devices, or any broken devices. The subsystem manages the Unavailable partition.

Service Configuration Options

StorageTek offers several different service options to consider when planning a Shared Virtual Array installation. These options are discussed in the following sections.

ServiceTek

ServiceTek is an optional service facility that automatically alerts the Customer Service Remote Center (CSRC) of an SVA exception condition. ServiceTek alerts identify when an SVA subsystem has failed, is about to fail, is running with degraded performance, or has error or statistical information to down-load.

Communication between the subsystem and the CSRC is sent via a modem or modem-MARS (machine-activated routing switch) connection. A modem is used to connect to a single subsystem. A modem and MARS is used to connect to multiple SVA subsystems. A MARS device monitors and responds to the incoming signals from the attached subsystems and allows connection to a modem for calls to the CSRC.

Performance Monitoring/Predictive Maintenance (PM2)

StorageTek’s Performance Monitoring and Predictive Maintenance (PM2) is an optional program that allows the user to print reports that include SIM information. PM2 provides performance and error reports for maintenance history, daily reporting, and monthly statistics. PM2 reports also aid the CSE in troubleshooting subsystem problems by correlating SIM and the ServiceTek information with the error and event logs.

Program for Online System Testing (POST)

StorageTek’s Program for Online System Testing (POST), another optional service program, offers an additional diagnostic tool that tests and verifies the reading or writing to functional volumes. POST reports all errors as they are reported in the sense data, and includes all SIM information in both online and stand-alone environments.

HCD/IOCP Definition for ESCON Channel Configuration

In general, an IOCP for a Shared Virtual Array configuration with ESCON channels has few differences with an equivalent parallel channel configuration. With minor modifications, the parallel channel IOCP can be made to work in the ESCON environment. Some of the main items of consideration are:

- Changes to the channel TYPE keyword in the CHPID statement to reflect usage of an ESCON channel path (TYPE=CNC).
- Usage of the PROTOCOL and SHARED keywords of the CNTLUNIT macro instruction are no longer necessary with ESCON-attached control units.

- The UNITADD keyword is required on both the CNTLUNIT and IODEVICE macro instructions with ESCON control units.
- The CUADD keyword is required on the CNTLUNIT macro instruction for Shared Virtual Array in the ESCON environment.
- The TIMEOUT keyword on the IODEVICE macro instruction is not used in ESCON attached devices.
- Control unit status verification should be enabled for Shared Virtual Array via the STADET keyword of the IODEVICE macro instruction (STADET=Y).
- ESCON Director usage - ESCD SWITCH numbers are required on the PATH keyword of the CHPID statement, along with port addresses for the LINK keyword of the CNTLUNIT macro instruction.
- In the EMIF environment, the PARTITION keyword of the CHPID statement is used to establish path access lists and candidate lists for partitions. The Access and Candidate lists are used to control shared and reconfigurable ESCON channels.

CUADD Keyword

The CUADD=address specifies the logical address for the control unit. The address is a one-digit hexadecimal number in the range 0-3. The address allows a CNC or CTC channel path to be specified on multiple CNTLUNIT macroinstructions for a given path (chpid.link or switch.link). For Shared Virtual Array, this results in two CNTLUNIT macro instructions for each control unit image. This will allow access to all virtual volumes from the designated paths. The CUADD keyword is required for Shared Virtual Array ESCON attachment.

UNITADD Keyword

UNITADD=Address specifies the unit address that is transmitted on the channel path to select the I/O device.

The UNITADD keyword on the CNTLUNIT macro instruction is usually specified as "00" for an attachment to the ESCON channel. SVA addresses are ‘trapped’ on a (maximum) 256 address boundary. The UNITADD for the CNTLUNIT macro instruction must be coded as UNITADD=((00,256)) to trap the maximum number of addresses. The number ‘256’ specifies that 256 sequential unit addresses are recognized by the Control Unit. Always specify the full range of devices.

If the UNITADD keyword is not specified in the IODEVICE macro instruction, the last two hexadecimal digits in the ADDRESS parameter specify the unit address that is transmitted on the channel path to select the device.

For the IODEVICE macro instruction, the UNITADD keyword must be coded as UNITADD=00 to begin each 256-address range. Multiple IODEVICE macros may be coded to define 3380 and 3390 devices or other attributes. The UNITADD parameter must be two digits within the hexadecimal range of 00-FF for each set of 256 addresses.

ADDRESS Keyword

The ADDRESS=(address,number) specifies a 1-4 digit hexadecimal number in the range 0000-FFFF. This assignment is the beginning of the device range as seen by the host system. The number parameter specifies the number of sequential devices (1-256).

CHIPID Statement

In the EMIF environment, the SHARED keyword has been added to allow sharing of channel paths between LPARs. Usage of the SHARED keyword is restricted to channel types of CNC or CTC. The PARTITION keyword identifies LPARs that are on the access list and which LPARs are on the candidate list. The format of the PARTITION keyword is: ...PARTITION=((access list), (candidate list)), SHARED. Access lists are used to denote paths that are shared at Power On Reset (POR) time, while the Candidate list is used to indicate that a particular path is reconfigurable to the specified LPAR after the initial POR has occurred.

Recommendations for ESCON HCD/IOCP

The following are recommendations for ESCON IOCP statements:

- FEATURE=SHARED on the IODEVICE statement is recommended to be coded for proper functioning of the RESERVE/RELEASE mechanism if devices are shared between LPARs or processors.
- For configuration of 1024 devices, each CUADD statement can address up to 256 devices. For example, UNITADD=((00,256)).

  CUADD=0 accesses the first set of 256 devices

  CUADD=1 accesses the second set of 256 devices

  CUADD=2 accesses the third set of 256 devices

  CUADD=3 accesses the fourth set of 256 devices
- Always use the UNITADD=00 statement for the IODEVICE.
- To eliminate any HCD restrictions, do not specify the model on the 3990 unit parameter for CNTLUNIT.
- Each PATH parameter in the CNTLUNIT macro instruction must match the channels attached to a cluster in the Physical Configuration. This allows for proper path rotation.
- The PARTITION parameter may be used on the IODEVICE statement to restrict access to devices by partition in the EMIF environment.

Recommendations for ESCON Channel Configurations

The following are recommendations for installing ESCON-attached Shared Virtual Array subsystems:

- Use only the default channel configuration of base=0, range=1024, and BFDID=0. These constructs are associated with trapping specific addresses on parallel (OEMI) channels. An ESCON channel of a 3990 control unit image can only receive 1024 addresses at a maximum. These configuration parameters (base, range, BFDID) are not required to limit the number of addresses trapped by an ESCON channel.
- Eight physical connections via four ESCON channel cards represent the ideal configuration for most applications. Additional channel cards may be necessary if an ESCON director is not available and additional channel connections are required. SVA may deal with more contention when more than eight channels are direct-connected. This may result in CU Busy situations which requires Control Unit initiated re-selections via request-in processing. The alternative is using an ESCON director. Channel contention now occurs at the director level via Port Busy. This process takes less than one hundred microseconds and requires no Shared Virtual Array resources.
- A maximum of eight logical or physical paths to a device can be configured from each operating system or LPAR.
- The eight-path group is a host processor limit established by IBM. HCD/IOCP error messages will be encountered if this limit is exceeded.

Chapter 2 Capacity and Performance

The Shared Virtual Array (SVA) offers advanced capabilities that provide efficient use of storage capacity and maximize subsystem throughput. This chapter discusses some of the capacity and performance factors that must be understood to achieve the full benefit of the SVA subsystem. This chapter also includes some associated concepts, as well as some traditional capacity management concepts.

Capacity and Performance Planning

The three basic steps in capacity and performance planning are as follows.

1. Use the appropriate measurement tools to obtain data capacity and performance characteristics.
2. Decide what data should reside on the SVA for the purpose of optimizing subsystem content.
3. Acquire and install the appropriate SVA subsystem(s) based on installation needs and capacity, performance, and effectiveness requirements.

The preceding steps, although not applicable to every situation, support the selection of an SVA configuration that is appropriate for a specific environment and workload. They assist in the successful transition to StorageTek's virtual storage architecture.

The following table shows the minimum recommended cache size for number and type of configured devices.

Table 2-1 Minimum Effective Cache Size

Configured Devices	Minimum Effective Cache Size
256 3390 Mod 3	1.5 GB
256 3390 Mod 9	1.5 GB
1024 3390 Mod 3	2.0 GB
341 3390 Mod 9	2.0 GB

Table 2-2 on page 22 shows the relative amounts of physical cache to effective cache.

Capacity Factors

The primary goal in selecting an SVA configuration is to maximize the value of Extended Capacity. The major capacity factors to consider, which are discussed in the following sections, are the subsystem's physical and functional capacity, data compression and data compaction effectiveness, and unallocated and unused space management.

Table 2-2 9500 System Volatile Cache Capacities

Physical User-Available Cache	Effective User-Available Cache (with 2:1 compression)
1024 MB	2048 MB
1152 MB	2304 MB
1280 MB	2560 MB
1408 MB	2816 MB
1536 MB	3072 MB
1664 MB	3328 MB
1792 MB	3584 MB
1920 MB	3840 MB
2048 MB	4096 MB
2176 MB	4352 MB
2304 MB	4608 MB
2432 MB	4864 MB
2560 MB	5120 MB
2688 MB	5376 MB
2816 MB	5632 MB
2944 MB	5888 MB
3072 MB	6144 MB

Note: 2:1 compression of 1024 MB to 3072 MB of physical customer cache provides 2048 MB to 6144 MB of effective customer cache.

Physical Capacity

In SVA, physical devices are organized into 15- or 7-device arrays (refer to "Disk Array Definition" on page 16). In capacity planning, note that the 15-device array option is the recommended and default disk array configuration.

Table 2-3 shows the physical capacity of different SVA configurations using different numbers of the recommended 15- and 7-device arrays for a 9500 (using 9GB SSA drives).

For simplicity, the performance information provided throughout this manual is based on 15-device arrays.

Table 2-3 9500 Subsystem Array Capacities

Equivalent Subsystem Capacity(1)	**Array Type(s)**	**No. of Arrays, by Type(s)**	**Total No. of Data Drives**	**Total No. of Redundancy / Parity Drives**	**Total No. of Spare Drives**	**Total No. of All Drives in the Subsystem(2)**
422 GB	13+2	1	13	2	1	16
845 GB	13+2	2	26	4	2	32
1267 GB(3)	13+2	3	39	6	3	48(4)
1689 GB(3)	13+2	4	52	8	4	64(4)

Note: (1) Equivalent subsystem capacity = number of data drives (n) x formatted capacity (8.51 GB) x typical compression ratio (4:1)
(2) Total of all data drives + redundancy/parity drives + spare drives (1 spare for 422 GB configuration, 2 for all others
(3) Capacity orderable only as an upgrade to an existing subsystem (162 MB/5+2 or 422 MB/13+2, as applicable)
(4) Configurations of >32 drives must use the 3390-9 device support, 4 array DCPSs
(5) Configurations of 5+2 arrays are not recommended because they degrade subsystem performance.

Functional Capacity

The type of volumes defined in the SVA functional configuration determines the functional capacity of the SVA subsystem. As stated previously, the SVA user may configure 1024 functional 3380 or 3390 volumes (or up to 341 3390-mod 9 volumes) for a 9500 with maximum capacity. Refer to “Functional Configuration” on page 14 for more information about the SVA functional configuration and its restrictions.

The functional configuration (the type, number, and use of functional volumes) has an impact on subsystem performance. For example, you can minimize the queue lengths within the host by defining more functional volumes.

Because of their potential significance, functional capacity performance considerations are discussed separately in “Performance Factors” on page 30.

Data Compression and Compaction

Data compression and compaction are standard SVA features that are very important when considering SVA capacity. Compression effectiveness depends upon the characteristics of the data being compressed. Studies of data-processing installations have shown that the average SVA data compression compaction ratio is approximately 0.32 (i.e., 100 GB of data occupies 32 GB of storage capacity or 0.32 x 100 = 32). The following lists two examples of SVA’s compression ratio:

- Temporary work space usually has a compression ratio of 0.27, which is more effective than the average.

- DFSMShsm host-compressed data has a compression ratio of 0.59, which is less effective than the average.

SVA's compression and compaction features enable effective use of physical storage capacity. A good compression and compaction ratio (e.g., a value such as 0.25) indicates that the SVA subsystem can store 300 gigabytes of equivalent traditional DASD capacity in less than 100 gigabytes of physical SVA storage.

Effective compression and compaction also tends to improve subsystem performance. Since all source data is automatically compressed upon entering the SVA subsystem and is not decompressed until exiting the subsystem, all operations and data transfers within the SVA subsystem are performed with compressed data. Given the same amount of cache and an identical workload as a 3990-3 subsystem, the SVA cache contains more data records; hence, there is a greater chance for a cache hit. A smaller cache size will usually produce an equal cache hit.

Unallocated Space

Generally, when source data is initially moved onto an SVA subsystem, unallocated and unused space is not moved and is not stored by SVA. In normal loading circumstances, only allocated and used space occupies physical capacity on an SVA subsystem. Therefore, a utility that copies only used data does not store unused space in SVA. However, if the utility used to move or copy data writes full tracks, allocated or not, then unused space is stored in SVA.

For example, moving a 630 megabyte DASD volume that is 50 percent allocated onto SVA requires only 315 megabytes of physical capacity, without taking compression or compaction into consideration. When compressed and compacted at the standard rate (0.32), the net physical storage consumption, or net capacity load (NCL) on SVA, is approximately 100 megabytes (i.e., 0.32 times 315 megabytes).

The overall storage consumption of data on SVA is a product of the allocation percentage and the compression-compaction index for the stored data, times the volume of data:

```
Storage= (allocation X (compression- X (total source consumption)percent-
age)compaction space in MB  index)or GB
```

Therefore, the physical space occupied by a data set on SVA can be significantly less than the functional or uncompressed space it occupied.

Studies of the DASD industry have shown an average space allocation in the 60 to 70% range. Recent studies of several installations indicate that average allocations may be approaching 75%, perhaps due to the beneficial influence of increasing System Managed Storage (SMS) usage within the industry.

For planning purposes, identifying unallocated space enables you to accurately size the physical storage capacity that you require. Allocation percentage is also important in assessing the potential to exploit extended capacity. Finally, the amount of unallocated or free space within an SVA subsystem may also have a significant performance impact (refer to "Performance Factors" on page 30).

Deleted Data Space Release

DDSR is a function of SVAA (host software) that assists the SVA in managing free space. When the operating system deletes a data set, the volume table of contents (VTOC) is updated, the data set is deleted, and the space becomes a free extent on the volume. On a traditional DASD device, physical tracks reside on the physical volume and cannot be used by any other volume, so there is no reason to reclaim these tracks. In the SVA subsystem, the functional data is dynamically mapped and never goes to the same place on the physical disks. Updating the VTOC still satisfies the operating system's requirements, and from a functional volume view, these tracks are available for reallocation. However, if the data on the physical track is not current, the SVA has the ability to reclaim this space and make it available to the entire subsystem, not just the one functional volume. DDSR exists because the deleting of an extent does not affect the tracks that contain the data of the deleted extent. The only updated track is the VTOC record, which is the only track that is released to free-space collection.

MVS systems delete a data set by removing it from the volume table of contents (VTOC). VM systems delete a minidisk by removing the entry from the directory. In either case, the data set or minidisk is no longer accessible to the host, but it continues to occupy physical storage capacity until it is overwritten by another file.

To reclaim the data tracks in the deleted extent, the host tells the SVA subsystem to release the physical sectors that hold the data that the host operating system now considers free space. This is precisely what the DDSR function does. A request is sent to the subsystem, which causes the subsystem to release the functional tracks that held the data that was deleted by the host. When these sectors are marked as free by the microcode, the change in NCL is immediately reflected. Because of the increase in collectable array cylinders, free-space collection begins collecting the array cylinders that contained these tracks and increases the amount of collected free space. Due to the limitations of free-space collection, the collected free space may not be reflected immediately, but will change over time as free-space collection does its work.

In SVA, if "deleted" data space is not explicitly released (or freed), then additional physical capacity may be required to compensate and to provide an adequate free space pool. "Performance Factors" discusses related performance considerations.

Space Release in VM

In VM, the user issues the RELEASE MINIDISK sub-command to initiate the release of the physical disk array capacity associated with a specific functional device address.

Refer to the *IXFP Configuration and Administration* for more information about DDSR and DDSR sub-command.

DDSR in MVS and OS/390

In MVS, SVAA/IXFP obtains control from the DASDM user exit IGGPRE00 during data set scratch or partial release functions. If the extent being deleted resides on an SVA volume, then the Dynamic DDSR function performs space release.

Interval DDSR is an alternative to Dynamic DDSR. With Interval DDSR, the user may specify the volumes and/or subsystems on which to run DDSR and at what time intervals.

Depending upon how many extents are released at one time, Interval DDSR may require more SVA subsystem resources than Dynamic DDSR and may impact other subsystem operations.

The following recommendations can be used as a starting point for implementing DDSR:

- Run Dynamic DDSR continuously.
- Consider running Interval DDSR periodically (for example, once a week) in the following situations.
  - After full volume restores
  - After re-initializing volumes
  - After a system outage in a shared DASD environment.
- To avoid unnecessary overhead activity, do not run Dynamic DDSR and Interval DDSR against the same volume at the same time. It may be advisable, however, to execute Interval DDSR to release space that may not have been freed if Dynamic DDSR was not continuously running on one or more hosts.
- If a volume is controlled by Dynamic DDSR, the Dynamic DDSR intercept should be activated on all hosts that have access to that volume, and the volume selection lists for the intercept should include that volume on all hosts.
- If a volume is controlled by Interval DDSR, the volume should be controlled by only one interval task across all hosts that have access to that volume.

Refer to the *SVAA User's Guide for OS/390* or the *IXFP Configuration and Administration*, as appropriate, for more information about DDSR and the DDSR sub-commands and parameters.

Space Release in Open Systems

In an Open Systems Solaris environment, SCSI partitions can be designated as free space and reclaimed. Traditional DDSR does not apply in the Open Systems environment.

Refer to the *SVAA for Solaris User's Guide* for more information.

Net Capacity Load

Net Capacity Load (NCL) represents the actual physical disk array capacity used to store the compressed functional tracks which reside on the SVA subsystem. NCL is presented in two formats: gigabytes (GB) and percent (%). NCL GB reports the actual physical disk array space, in gigabytes, used to store the compressed functional tracks. NCL % is the percent of the total configured physical disk array capacity used to store the compressed functional tracks. NCL increases as new data is written to the subsystem. NCL decreases as data is deleted from the subsystem by DDSR. NCL will increase or decrease when data on the subsystem is updated based on how the compressed updated data compares in size to the compressed original data. It is normal for NCL to fluctuate during operations as data is written and as space is released by DDSR.

Calculating Net Capacity Load

The NCL% is calculated by dividing the NCL GB by the total configured physical disk array capacity in gigabytes, and then multiplying by 100.

NCL% = (NCL GB / Total Physical Capacity GB) x 100

Managing Net Capacity Load

To effectively administer most user workloads, StorageTek recommends a maximum NCL percentage of 75%. At or below this NCL percentage, Free Space Collection is capable of efficiently collecting the uncollected free space generated by most user workloads without impacting performance. When the NCL percentage increases to beyond the 75% range, the update write and write/delete content in the user workload becomes a factor. User workloads with a high update write and/or write/delete content will generate enough uncollected free space to force Free Space Collection to use more subsystem resources and potentially degrade performance. User workloads with a low update write and write/delete content can operate a higher than 75% NCL percentage without impact to performance due to Free Space Collection.

Several methods can be used to manage the subsystem as the NCL percentage increases beyond 75%. These methods are listed below.

1. Add more physical capacity to the subsystem by configuring another disk array.
2. Move some data to another SVA subsystem with a lower NCL percentage.

3. Reduce the update write and write/delete content of the user workload by swapping data between another SVA subsystem.
4. Install an additional SVA subsystem.

Collected Free Space

In an SVA subsystem, data written to an array is always written to a free array cylinder. (Functional volumes and functional tracks are the host view of the SVA subsystem. Arrays, array tracks, and array cylinders are the SVA view of the physical storage devices.) These array cylinders are placed in a free cylinder list and are available to be selected and used for writing either new data or updated data to the subsystem. Collected free space is not part of the net capacity load.

The disk array controller only schedules write operations to free array cylinders. Data are written to these array cylinders as records are updated or created on functional volumes. SVA must recycle array cylinders as data is updated or it would eventually run out of free array cylinders. To recycle free array cylinders, it collects all current data from an allocated array cylinder and rewrites it to a new free array cylinder called a destage array cylinder.

How Free-Space Collection Works

The SVA Controller performs free space collection as a background task. The free space collection facility uses the free space list to identify which array cylinders have the most free space.

As SVA destages data to the disk arrays, it updates the free space list. As allocated array cylinders age, they migrate from 100% current data towards 100% free. If the free space collection facility determines that there are enough collectible array cylinders to justify collection, it opens a destage array cylinder. (This is a different array cylinder than is used for normal write processing.) It then begins copying functional tracks from the array cylinders with the most free space to the destage cylinder. Each time a destage cylinder is filled, FSC closes the current destage cylinder and adds all of the collected array cylinders to the free-cylinder list. FSC then checks to see whether there are still enough collectable array cylinders to continue doing work. If there are, FSC opens another destage cylinder. If not, then free space collection "sleeps" until the next time tracks are released. FSC then checks to see if there are enough collectible array cylinders and, if so, begins collecting again.

Free space collection affects the amount of collected free space displayed in SVAA/IXFP reports and displays, but does not affect the NCL. NCL is modified by changing the number of stored sectors in the array cylinders and reflects stored data rather than free space, and so NCL is unaffected by free-space collection.

Shared Virtual Array SIM Messages

SVA reports a shortage of free space when there is less than 10% collected free space by issuing a service information message (SIM). The SIM contains a REFCODE, the first two bytes of which are the fault symptom code '3E41' that indicates low on capacity. The fault symptom code '3E40' indicates that the back-end capacity has been exhausted. At the same time, SVA changes its thresholds to allow FSC to do more work. This allows free space collection routines to collect free space from more array cylinders for writing new data. As SVA approaches 0% collected free space (CFS), the free space collection routine is allowed access to more array cylinders as it tries to collect all free space remaining in the subsystem.

If you receive this low-on-capacity SIM, take the following steps:

1. Monitor the FSC using SVAA/IXFP Space Utilization reports, PM2 reports, or SVAA/IXFP and SVA local operator panel.
2. Identify which files or functional device can be migrated to another storage subsystem or backed up to tape.
3. Initiate the migration of the files or functional device to another storage subsystem or back up the files or functional device to tape.
4. Delete the files or functional device. (To delete a functional device, first vary the functional device offline to all attached hosts.)
5. Wait for free space collection and DDSR to return the capacity to the available cylinders pool.

Note: When a file or a functional device is deleted, the disk array capacity that it occupied may not become available for several minutes to several hours.

If SVA has no more array cylinders to allocate, it no longer accepts any commands that require a write operation, including commands that are issued to browse, scratch, or delete a dataset. Such commands are rejected with INTERVENTION REQUIRED sense data.

If you receive an out-of-capacity SIM, take the following steps:

First, check for uncollected free space. If there is uncollected free space, free space collection will collect it and the Intervention Required state will be reset. Reduce the update write and write/delete content of the user workload and then follow the low-on-capacity procedure. If no uncollected free space exists:

1. Migrate data to archives.
2. Delete volumes with temporary or old data currently not in use.
3. Add storage capacity to increase the collected free space to acceptable levels. Acceptable levels would be considered 15-20% collected free space.

Note: At this point, deleting files is not sufficient; you must delete a functional device.

4. Initiate the migration of the functional device to another storage subsystem or back up the functional device to tape.
5. Vary the functional device offline to all attached hosts.
6. Delete the functional device.
7. Wait for free space collection and DDSR to return the capacity to the available cylinders pool.

Note: When a functional device is deleted, the disk array capacity that it occupied may not become available for several minutes to several hours.

If SVA has no more array cylinders to allocate, it no longer accepts any commands that require a write operation including commands that are issued in an attempt to browse, scratch, or delete a data set. Such commands are rejected with INTERVENTION REQUIRED sense data.

The subsystem will cancel the INTERVENTION REQUIRED condition when enough user capacity is available to sustain write activity.

Another strategy for managing a subsystem's CFS is to:

- Designate one functional device in the Production partition as a work volume.
- This functional device should contain an expendable data set (e.g., a work volume of temporary data) that occupies at least 64 megabytes of capacity.
- Designate one functional device *with low write activity* as a privileged ECAM device.
- This is the ECAM device that will accept the write operation and allow you to delete the work volume.

Note: StorageTek recommends operating SVA at 75% NCL or less. This does not mean that there should be 25% CFS. Free space is divided into collected and uncollected free space. At 75% NCL, there will always be 5-10%. UFS, so 15-20% CFS should be considered normal operating levels.

Performance Factors

SVA offers a unique, advanced architecture that affects both the storage and the processing of data. Along with performance and capacity management concerns, throughput and performance features, such as dynamic mapping, fault tolerance, and dynamic configuration must be considered.

SVA also introduces some non-traditional concepts and factors that influence performance. Because data is compressed and compacted before it is stored, simple throughput (i.e., I/Os per second) is no longer the predominant factor to consider in capacity and performance planning.

Throughput relative to storage capacity or access density is an equally significant factor. Also, net capacity load becomes important.

Keep volumes that are not in use off-line to avoid unnecessary activity on the subsystem.

Optimizing Subsystem Content

SVA offers opportunities for improved performance and increased capacity for well over 90 percent of the online data in most environments. Because of SVA's caching, high bandwidth back-end data transfer, and parallelism, performance is generally excellent at high throughput levels. Extended capacity provides the capability to maximize storage capacity; however, there are some physical throughput limitations.

Although SVA supports any type of data or workload, methods exist that improve the overall capacity and performance effectiveness of installed subsystems. The purpose of these methods is twofold:

- To limit the content of an SVA subsystem to that data for which it is best suited.
- To place the remaining data on an appropriate device-type for it.

One method limits SVA data that does not appear to perform well in either normal cache or DASD fast write operations.

Another method limits data with very high access density. Very high access density makes it more difficult to make effective use of extended capacity.

Improving Performance

Five main subsystem design methods provide for or improve SVA's performance. They are:

1. Exploit the performance-enhancing features of SVA.

   Defining 256 or 1024 UCBs can flatten skew across functional volumes. Defining the appropriate functional volume types can reduce contention within volumes.

2. Use index VTOCx to improve VTOC access.
3. Incorporate performance-enhancing hardware options, such as additional host channel connections or larger cache sizes.
4. Exclude or remove data from SVA that does not gain specific performance advantage from cache or DASD fast write.

These options must be individually assessed using the appropriate performance measurement tools. SVAA/IXFP subsystem reports help significantly in this area.

Increasing Capacity

Clarify your need for additional capacity. The most common reason is a need for more storage capacity for data. Sometimes, however, you may wish to increase capacity to achieve a higher level of throughput or I/O rate. Either of these situations is best addressed by improving performance, which also tends to increase usable storage capacity.

Optimizing Extended Capacity

Capacity utilization is addressed by a different set of methods and configuration options than are performance issues. There are two primary ways to provide or increase the potential for using extended capacity and to enhance its usefulness. They are:

1. Improve the cache effectiveness of the data in the SVA subsystem.
2. Reduce the access density of the data being placed on the SVA.

   This method enables the effective utilization of larger subsystems at a higher net capacity load, enabling more effective use of extended capacity (up to the 75% recommended NCL limit).

Self-Tuning Operation

Dynamic Mapping allows the SVA subsystem to be self-tuning. In a conventional DASD subsystem, user data and I/O activity might normally be concentrated on one physical storage device or actuator. However, because data space and files are spread throughout an SVA disk array, I/O activity is dispersed across several physical devices. Therefore, SVA's multi-path access to back-end storage and its high degree of parallelism can be exploited to maximize data transfer bandwidth.

Media Maintenance, Track Recovery, and Drive Reconstruction

Media maintenance and functional track recovery have minimal performance impact on the entire subsystem. Overall influence from single drive reconstruction, however, varies depending on the number of arrays in the subsystem, the existing net capacity load, current throughput levels, cache effectiveness of the workload, and the target elapsed time for reconstruction to complete. As a general guideline, single drive reconstruction, with an expected reconstruction time of two hours or less, degrades the average I/O service time by less than one or two milliseconds. If a second physical device fails in the same array during drive reconstruction, the drive reconstruction task becomes a high-priority task that normally completes in five to ten minutes, but may require more resources.

UCB Queuing and Functional Volumes

Because Dynamic Configuration allows you to define a subsystem with up to 1024 volumes, host queueing for access to functional volumes may be significantly reduced or, in many cases, eliminated. Contention for reserved volumes in a multi-host environment may also be reduced.

A performance advantage is gained by defining and using as many functional volumes in a configuration as possible. An SVA subsystem user should make as much use of this feature as appropriate to balance functional volume activity and reduce single UCB contention.

Dynamic configuration also supports the definition of functional volumes ranging in device type and size from a single capacity 3380 device to a 3390-9 device. This capability can be used to define volumes just large enough to contain selected files, thereby preventing too many allocations, and the associated contention, on the same volume. This feature can also be used to define very large public or storage pools (volumes), which can minimize “out of space” abends and reduce volume fragmentation.

Access Density and Throughput

Access density and throughput are closely-related concepts. Access density measures throughput in relation to the storage space occupied by the data that is sustaining the I/O activity. While throughput is measured as I/Os per second (IO/SEC), access density is measured as I/Os per second per gigabyte of storage space (IO/SEC/GB).

A common measure of access density is physical access density, which is the throughput of the data in relation to the physical space available to, or reserved for, the data. Unallocated or unused physical space within the corresponding storage area, whether it be a single DASD volume, a storage device pool, or an entire subsystem, is considered when computing the access density.

Given similar throughput rates, an SVA subsystem that contains data with a low access density can store more data than the same SVA subsystem with a high access density. For additional information on access density, throughput rates and capacity, refer to “Optimizing Subsystem Content.”

Channel Configuration

Channel configuration can be a significant performance factor on an SVA subsystem. Modeling data indicates that defining 1024 functional volumes to an SVA subsystem significantly improves the performance of a TSO workload. TSO workloads are characterized by a very high cache hit rate. A 25% improvement in TSO response time, primarily due to reduced channel utilization, was observed in modeling when an SVA was configured with eight channels rather than four.

Conversely, batch workloads are not significantly affected by increasing the number of channels from four to eight, because batch workloads have much lower cache hit rates than do TSO workloads. In a batch processing environment, cache is much more of a bottleneck than is channel utilization, so additional channels minimally affect response times.

Other Performance Factors

Several additional factors, which are listed below, significantly influence SVA's performance. Although these performance factors also affect traditional DASD subsystem performance, their significance is often somewhat different for SVA, because of SVA's architectural differences.

1. Block size of the data - Larger blocks may lengthen I/O service time. In SVA, this influence is minimized since data is always compressed and the physical transfer of blocks occurs only between the cache and the host CPU. Additionally, SVA always stages a full compressed track to cache.
2. Locality of reference - Poor locality of reference (LOR), associated with source data for a cached subsystem, usually indicates a potential for poor cache-hit performance. In SVA, this potential is minimized, because the data is compressed in cache, which enables a more effective use of an equal cache capacity or equal cache effectiveness with less cache capacity.
3. DASD skew - DASD skew is a measure of I/O concentration on a few volumes of a subsystem. It has a potentially severe performance impact on traditional DASD subsystems. However, because SVA's standard self-tuning operation disperses functional tracks throughout a subsystem, DASD skew is often reduced at the physical device access level. Also, because SVA makes it easy to define additional functional volumes and spread the data across more volumes, IOS queuing in the host computer systems is relieved.
4. Read/write ratio - Because all write operations in SVA are DASD fast writes, this factor has a minimal performance impact. However, this ratio can affect the level of NCL that an SVA subsystem can reasonably sustain. SVA uses much of its free space to support normal update and write activity. If the update or write activity in a subsystem is minimal, then an NCL approaching 100% is achievable without severe performance degradation.

SVA is more tolerant of data with poor cache characteristics than traditional cached DASD subsystems. It benefits from the use of:

- Actuator level buffers (ALB)
- Higher back-end bandwidth
- Extensive parallelism
- Compressed data for both internal data transfers and cache storage.

RMF Performance Metrics

SVA operates differently than does the 3990, which creates differences in the RMF data produced when the same workload is run in these two DASD subsystems. The following sections describe some of these differences.

Pend time versus disconnect time

When the front-end data paths of a DASD controller are heavily utilized, additional I/O requests are more likely to be accepted and queued in an SVA subsystem than in a 3990 subsystem.

In a 3990 subsystem, when a channel attempts selection and both paths of the multi-path storage director are busy, the subsystem presents Control Unit (CU) Busy. If it is configured correctly, the SSCH may be retried over a path connected to the other storage cluster, where it may or may not also encounter CU Busy. The 3990 supports a total of four concurrently connected channel interfaces. A director path remains busy during data transfer over that path. When an I/O request is delayed because of CU Busy, the delay time appears in RMF reports as pend time.

Although SVA also has 16 front-end data paths to cache, it has eight functional code processors, all of which can be concurrently processing a connected channel program. An SVA subsystem can concurrently process up to 16 connected channel programs. When a selection request arrives at an SVA storage cluster, it receives CU Busy only if none of the eight microprocessors in the cluster is available to support the connection request.

Whereas a 3990 would be more likely to present CU Busy, an SVA subsystem would be more likely to accept initial selection, process the Define Extent and Locate Record commands, and disconnect while waiting for a front-end data path to become available. When running the same workload on the two subsystems, RMF will report a smaller than expected increase in the average pend time for SSCH (based on the 3990). At the same time, RMF will report a larger-than-expected increase in the average disconnect time for a SSCH (based on the 3990).

No fast write bypass

In an SVA subsystem, all write operations pass through cache; accessing DASD directly is not an option. When NVS in an SVA subsystem is full, the Controller disconnects from the channel attempting to write and reconnects later when the NVS full condition has been relieved. SVA never increments the statistical counters of fast write bypass operations.

Measurement Tools

The preceding information focused on capacity and performance, the physical nature of the source data that is being considered for placement on SVA, and how SVA is likely to operate relative to the source data.

The following sections discuss measurement tools that are available to collect information and to assess its significance in an existing environment.

MVS Tools

In an MVS/ESA environment, the following standard tools are available to collect key planning statistics about data usage:

Resource Measurement Facility (RMF)

The most appropriate RMF report is the Direct Access Device Activity Report. For this report, statistics are gathered for one or more peak processing periods (e.g., prime shift) with DINTV(100) to capture the peak-hour DASD activity rate. With the peak-hour I/O rate value and knowledge of the existing DASD configuration, peak-hour access density can be determined. The potential or need for functional device re-configuration to exploit SVA's unique architecture may be determined using the same report.

System Management Facility (SMF)

Numerous data reduction and analysis packages are available to aid in the analysis of SMF data. SMF data is often most useful as a long term (i.e., days or weeks) resource measurement tool, but it can also be used to support specific data set or data pool analysis, where RMF cannot.

IDCAMS Listdata

Can provide real time cache statistics for the subsystem or device.

VM Tools

In any VM environment, the standard VM/Monitor can be used to collect resource utilization and capacity planning data at user-specified intervals over desired time periods. Different packages, depending on the level of VM system in use, are available for the reduction and reporting of this data.

- Under VM/ESA, the Virtual Machine Processor Resource Facility (VM/PRF) can be used to analyze and produce standard reports on the collected data.
- The Virtual Machine Performance Planning Facility (VM/PPF), which is a combination of VM/PRF with a capacity planning capability, can be used for longer term analysis.
- The Real Time Monitor/Systems Facility (RTM/SF) can be used for short-term, dynamic monitoring and analysis.

SVAA Subsystem Reports

The Shared Virtual Array Administrator (SVAA) offers a subsystem reporting software package. SVAA reports are primarily designed to analyze SVA subsystem performance and capacity utilization. Additionally, a subset of SVAA reports can be produced for existing 3990-3 (or equivalent) cache subsystems. In particular, several variations of the SVAA cache effectiveness reports can be generated to aid in the evaluation of potential cache effectiveness for source data currently residing on a cache controller.

Capacity and Performance Planning Recommendations

Capacity and performance planning for SVA can be straightforward if you understand the capacity and performance factors and appropriate methods and tools that influence an SVA subsystem.

Use the following basic steps for initial planning and continuing control.

1. Decide what data is to be placed on SVA.

   This may include specific files, specific storage pools, specific existing volumes, or even the totality of current on-line storage.

2. Use available measurement tools (e.g., RMF and SMF, VM/Monitor, etc.) to obtain the capacity and performance characteristics of this source data.

   Most notably, this should include space allocated, space available, space used, throughput rates, compression and compaction potential, space needed (if different from space available), and some measure of cache effectiveness (e.g., Locality of Reference). Refer to “Measurement Tools” on page 36.

3. Use the methods described in “Selecting an SVA Subsystem Configuration” on page 13 to select, acquire, and install the appropriate SVA subsystem(s).

4. Implement advanced SVA capabilities and features (e.g., extended channel connectivity, extra spare disks, DDSR, 1024 UCBs, variable device types, etc.) to maximize configuration reliability, availability, and performance, as well as to monitor and control continuing operations.

5. The recommended overall approach to SVA configuration planning and design is to implement a moderate balance between capacity- and performance-effectiveness, thereby reaping the benefits of both. An alternative method to achieving the same result is to separate the source data between capacity-effective and performance-effective subsystems.

   Capacity-effective subsystems are most suitable for static (limited growth) data. Performance-effective subsystems are best for dynamic (active growth) data.

If properly followed, these steps enable the selection of an appropriate SVA configuration(s) for a specific environment and workload. They also support a successful transition to StorageTek's virtual storage architecture. However, priorities and workload may change over time, and some adjustments may then be necessary.

Chapter 3 SVA For MVS or OS/390 Use

Installing Shared Virtual Array

After installation, as with any other I/O device, a Shared Virtual Array (SVA) subsystem must be defined to MVS or OS/390 and the channel subsystem. The process for defining devices may vary, depending on the level of the operating system. This chapter describes the special software considerations for defining the SVA subsystem to an operating system. Functional disk initialization using ICKDSF is also discussed.

Minimum Program Levels

The operating system and corresponding DFSMSdfp release levels required for SVA implementation is dependent on the functional device types used. Minimum prerequisite host software levels are listed in Table 3-1. This provides a ***baseline*** for additional PTFs and levels of software applications required for implementation of a Shared Virtual Array system.

Table 3-2 on page 40 and Table 3-3 on page 42 show current levels (as of document publication date) of additional required software.

These requirements are all subject to change.

Table 3-1 Minimum Host Software Requirements

Operating System	Version and Release Level
OS/390	Version 1, Release 2
MVS/ESA	Version 5, Release 2.2
VM/ESA	Version 2, Release 2.3

The host operating system must be able to support a 3990 storage control unit and the functional device types being defined. Before installing the initial SVA subsystem, install and verify all of the current maintenance recommended in support of the 3990, 3380 and 3390 volumes, and 4-digit addressing. The most current levels of software maintenance are highly recommended.

Table 3-2 Required IBM Software Support Level

IBM OS/390 Software Support Maintenance
DFSMS 1.5 = FMID HDZ1**1E0**
DFSMS 1.4 = FMID HDZ1**1D0**
DFSMS 1.3 = FMID HDZ1**1C0**
DFSMS 1.2 = FMID HDZ1**1B0**

	APARs	**PTFs**
OS/390, DFSMS		
	OW26616	
		1D0 + UW39622 1C0 + UW37783 1B0 + UW37782
	OW32197	
		1D0 + UW90553 1C0 + UW90552 1B0 + UW47173
DEVSERV		
	OW27859	
		1D0 + UW90553 1C0 + UW90552 1B0 + UW40316
DFSMSdss		
	II11078 VTOC information	
	OW41368	
		1E0 + UW64100 1D0 + UW64099 1C0 + UW64098 1B0 + UW64097
	OW41482	
		1E0 + UW64755 1D0 + UW64754 1C0 + UW64753

Table 3-2 Required IBM Software Support Level

DFSMSdss (PPRC and SnapShot)		
	OW37501	
		1E0 + UW59054 1D0 + UW59053 1C0 + UW59052
	OW38194	
		1E0 + UW66061 1D0 + UW59079 1C0 + UW59078
	OW38623	
		1E0 + UW59083 1D0 + UW59082 1C0 + UW59081
	OW39651	
		1E0 + UW63033 1D0 + UW63032 1C0 + UW63031
DFSMSsdm (PPRC)		
	OW35368	
		1E0 + UW54434 1D0 + UW54433 1C0 + UW54432 1B0 + UW54431
RMF		
	OW31700	
		HRM6604 + UW90450 HRM6603 + UW90449 HRM6602 + UW90448 HRM5520 + UW90447
ICKDSF 1.16.0		
	PQ02288 (9393) PQ20390 (PPRC)	
		EDU1G01 + UQ90018

Table 3-3 Required StorageTek Software Levels

	APARs	PTFs
IXFP/MVS V2R1		
	OW33715	L170637
	OW36501	L170662
	OW36513	L170672
	OW37650	L170686
	OW38337	L170696
	OW39374	L170735
	OW40796	L170741
	OW40850	L170744
SNAPSHOT/MVS V1R2		
	OW36507	L170663
	OW38367	L170698
IXFP/VM V2R1		
	VM62024	L170685
	VM62227	L170736

Configuration Considerations

The SVA subsystem does not have physical disk volumes in the traditional sense; however, the operating system does access disk-resident data as though it were on traditional DASD. Data volumes, as seen by the operating system, are functional volumes in SVA terms. (Refer to the *9500 Shared Virtual Array Introduction* for a more detailed discussion of functional volumes.)

From an operating system perspective, functional volumes are defined just as traditional DASD is defined: IODEVICE macro definitions describe all of the functional devices that are to be accessed by the operating system.

The SVA subsystem should be defined as four 3990 storage controls with a full complement of 3380 and/or 3390 devices.

There are two distinct advantages to defining the maximum available number of functional volumes in MVS or OS/390 (this example assumes a maximum functional devices):

- If fewer than 1024 functional volumes are initially configured on the SVA subsystem, more may be added (i.e., defined) at a later date without an IPL (assuming that the operating system and the IOCP are configured for 1024 volumes). The new volumes may then be varied online from MVS or OS/390 after being defined to the SVA subsystem.

- If properly implemented, defining the maximum number of volumes allowable on the subsystem reduces the I/O delay at the host due to UCB queuing.

The major disadvantage to defining 1024 devices for an SVA subsystem is the amount of host virtual storage required to support 1024 UCBs.

The implications of defining 1024 devices should be evaluated for each specific site. The industry trend has been toward systems with increasing amounts of memory, and the benefits of improved I/O response probably outweigh the penalty paid in virtual storage used.

Using RMF to study and understand the UCB queuing in the current storage subsystem will help to determine the need for additional volumes. Similarly, storage utilization reports can help to determine possible storage constraints.

SVA allows you to configure functional volumes from the local operator panel or at the host console through IXFP (batch and ISPF).

Preparing the Hardware

The use of the ICKDSF format command to initialize SVA functional volumes in MVS or OS/390 is identical to the process used for traditional DASD volumes. Due to differences in the SVA architecture over conventional DASD, there are some differences in the recommendations for formatting.

Cache Operation

Although the cache operation on an SVA subsystem is essentially the same as that of a 3990, SVA's response to some of the 3990 commands varies from the 3990 response. The differences are described below.

Dual Copy

SVA does not support the 3990 dual copy feature, because the feature is not necessary in SVA. In response to dual copy commands such as Establish, Terminate, or Suspend Duplex Pair, SVA presents status and sense data indicating that the device does not support dual copy.

DASD Fast Write And Cache Fast Write

DASD fast write (DFW) and cache fast write (CFW) are both fully supported by the SVA subsystem. All data transfers from the cache to the disk array are an asynchronous background task.

The implementation of cache fast write in the SVA subsystem is slightly different from the 3990-3. SVA acknowledges and records cache fast write commands, but all writes and data transfers in an SVA subsystem are DASD fast writes. Unlike the 3990-3, the data is written to both cache and to nonvolatile storage. This process ensures that in any circumstance, there is a secure copy of the data.

IDCAMS

The *IBM 3990 Storage Control Planning, Installation, and Storage Administration Guide* documents the cache functions of IDCAMS. The similarities and differences in SVA's responses to IDCAMS cache control commands are documented here.

LISTDATA

The LISTDATA command provides activity and status reports about cache, including:

Subsystem counters report

This report generates a report on subsystem counters at the device, subsystem, or system level by specifying DEVICE, SUBSYSTEM, or ALL, respectively. Device counters are for the functional devices defined with SVA subsystems.

Remote access code

IDCAMS can be used to generate a remote support access code for the SVA subsystem.

Pinned track to data set cross reference report

Because SVA does not experience pinned data, this report always indicates that no data is pinned. SVA's fault-tolerant design allows the subsystem to recover from both cache and NVS failures.

Device status report

This report describes the device status with the channel connection address and the director-to-device connection address for each device. SVA supports this function, and "cache inactive" is reported for cache status requests for devices that have caching turned off, even though SVA subsystems continue to cache all data.

SETCACHE

The SETCACHE command has several functions. In SVA, all directives for volumes operate on functional volumes.

The SETCACHE command:

- Makes functional volumes eligible for caching operations

  In an SVA subsystem, all data is cached. Setting cache off for a device minimizes the cache storage available for that volume by changing the cache management algorithms. Modified tracks are immediately queued for de-staging, and unmodified tracks are discarded when other tracks need the space they occupy.

- Makes subsystem storage available or unavailable to the subsystem for caching

This option performs the same function as turning cache off for a device or for all 64 devices attached to a functional storage control. The subsystem returns “cache inactive” for queries of its status. When storage becomes available, all normal SVA caching algorithms become active.

- Makes cache storage unavailable when cache is in a pending state

  This operation is identical to that described for the 3990-3.

- Makes NVS available or unavailable to the subsystem

  Making NVS unavailable to the subsystem is the same as making cache unavailable to the subsystem. NVS algorithms for affected volumes minimize the amount of NVS available for those volumes. If NVS is unavailable, inquiries about NVS return “NVS inactive.” Normal NVS processing resumes when NVS becomes available to the subsystem.

- Activates or deactivates DASD fast write for a functional device

  SVA performs all writes as DASD fast writes. If DASD fast write is deactivated for a functional device, queries for that device indicate that DASD fast write is inactive. However, DASD fast write for that device is still active, but the caching algorithm is changed. The cache frames occupied by a non-cached volume are placed low on the least-recently used (LRU) list and are preferentially selected for writing to the disk array when write processing is being performed.

- Makes cache fast write available or unavailable

  SVA supports cache fast write. Commands managing cache fast write respond as documented by StorageTek.

- De-stages modified data

  Commands to destage modified data are processed as documented by StorageTek.

- Discards pinned data

  Because SVA does not experience pinned data, all commands that manage pinned data receive a “no data pinned” response.

- Re-initializes the subsystem and device status to INITIAL INSTALLATION DEFAULT VALUES

  This process is performed as documented for the 3990-3 except as it concerns pinned data and duplex pairs. SVA does not experience pinned data, so data cannot be lost, and SVA does not support dual copy commands

Exploiting IXFP

IXFP provides an additional level of cache control for a Shared Virtual Array subsystem beyond that provided by the local operator panel. In addition, IXFP maximizes the benefits of SVA's virtual storage architecture by allowing interactive control of activities such as subsystem administration and reporting.

IXFP has additional reporting capability for SVA and other 3990-compatible cache controllers. IXFP can be used to produce cache performance reports, including information on cache hit ratios and cache occupancy. The information from these reports can be used to effectively manage cache utilization.

There may be some differences between the data produced by IXFP reports and the data produced in the IDCAMS LISTDATA output. For compatibility reasons, the SVA counts certain I/Os as non-cached in response to LISTDATA. Therefore, the IXFP reports reflect actual SVA performance more accurately.

The IXFP facility is described in more detail in the *Shared Virtual Array Introduction*, and the *IXFP Configuration and Administration* manual.

MVS or OS/390 Tools

In an MVS or OS/390 environment, the following standard tools are normally available to collect key planning statistics about data usage:

- **Resource Measurement Facility (RMF)**

  The most appropriate RMF report is the Direct Access Device Activity Report. For this report, statistics are gathered for one or more peak processing periods (e.g., prime shift) with DINTV (100) to capture the peak-hour DASD activity rate. With the peak-hour I/O rate value and knowledge of the existing DASD configuration, peak-hour access density can be determined. The potential or need for functional device re-configuration to exploit SVA's unique architecture may be determined using the same report.

- **System Management Facility (SMF)**

  Numerous data reduction and analysis packages are available to aid in the analysis of SMF data. SMF data is often most useful as a long-term (i.e., days or weeks) resource measurement tool, but it can also be used to support specific data set or data pool analysis, where RMF cannot.

- **IDCAMS Listdata**

  Can provide real time cache statistics for the subsystem or device.

Storage Management in MVS or OS/390

Special Utilities - Device Support Facilities

Device Support Facilities (ICKDSF) is a software utility that performs volume initialization and disk media maintenance. Volume initialization prepares a disk volume for use by the operating system, while disk media maintenance provides a means to detect and correct problems with disk surface defects.

Use ICKDSF to initialize SVA functional volumes, either as 3380s or 3390s. With traditional DASD subsystems, incorrect placement of the volume table of contents (VTOC) can cause excessive disk head movement. Excessive disk head movement causes I/O performance degradation. Because an SVA subsystem spreads data across multiple volumes in the disk array, excessive head movement is minimized. With an SVA subsystem, there are no performance penalties for VTOC placement; however, the VTOC should be placed at the beginning of the functional volume to preserve contiguous free space. Only an ICKDSF MINIMAL INIT is necessary and recommended for SVA.

Disk errors that cause an interruption in data availability for traditional DASD subsystems are eliminated with an SVA subsystem. SVA can automatically correct media errors and dynamically reconstruct a failed disk drive. It is never necessary to use ICKDSF for media maintenance on SVA functional volumes. Refer to the *Shared Virtual Array Reference* for further information on SVA drive reconstruction and automatic media maintenance.

The following table provides a list of ICKDSF commands that are supported by an SVA subsystem:

Table 3-4 ICKDSF Commands Supported by SVA in MVS or OS/390

ICKDSF Commands	**Restrictions**
AIXVOL with no surface checking functions	READCHECK is not supported
ANALYZE SCAN function only	DRIVETEST is not supported
BUILDIX	None
CPVOLUME with no surface checking functions	READCHECK is not supported
INIT minimal initialization	CHECK and VALIDATE are not supported
REFORMAT	None

Data Recovery

SVA recovers from most hardware failures automatically. However, as with any other DASD subsystem, data corruption caused by application programs or user errors must be corrected by the customer.

As a precaution when correcting VTOC errors on an SVA functional volume, suspend Deleted Data Space Release (DDSR) until the VTOC errors have been corrected.

Data Migration Planning

Migrating data to an SVA subsystem is similar to migrating data to any other form of disk storage. The SVA subsystem is designed for 3990 compatibility. As insurance against accidental data loss and to expedite the migration process, develop and document a detailed migration plan prior to moving data. Make backups prior to moving any data to guard against accidental loss during migration. Also, secure a backup and recovery plan as part of the complete migration plan.

Migration Strategies

The simplest migration method is a direct volume copy from the source volumes to the target functional volumes. This strategy is also the most expedient and similar method to existing migration plans. However, this method does not attempt to exploit any of the special features of the SVA subsystem.

The SVA subsystem features that are automatically available, regardless of the migration strategy used, include:

- Fault tolerance - the data is protected from hardware failures.
- Data compression - the data on an SVA subsystem is compressed upon entering the SVA Controller. This strategy reduces the amount of cache and physical disk space required to store the data.
- Data compaction - inter-record gaps are removed, further decreasing the amount of space required to store the data. In addition, RPS miss is eliminated.

When using the volume copy method, be aware that some volume copy utilities copy track images, which includes residual data in unused and un-allocated space on the track. SVA treats the images as valid data, compressing and storing un-allocated space during the copy process. Therefore, use utilities (or copy options) that recognize the end of data, and copy only active data. This reduces the net capacity load on the SVA subsystem following a data migration. If storing the unused portion of track images is unavoidable, the net capacity load can be reduced to actual data allocation after migration by using the DDSR feature of IXFP.

DFSMSdss is a commonly-used data migration tool.

DFSMSdss is perhaps the simplest and most direct volume copy method. Using this package is exemplified by the following full-volume copy statement:

```
COPY FUL1 INDD(ddn1) OUTDD(ddn2) COPYVOLID
```

The values for ddn1 and ddn2 identify DD statements that define the source and target DASD volumes. DFSMSdss defaults to copying only the used space for allocated data sets and nothing for empty data sets. COPYVOLID can be used to copy the VOLSER from the input volume to the output volume (it is required for a full-volume copy of an SMS-managed input volume). At the end of the copy operation, before the volume serial number on the receiving volume is altered, the operator is notified and the operating system initiates a demount of the volume.

Examine Data Access Characteristics

A more comprehensive process for migrating data to an SVA subsystem involves studying and understanding the I/O activity of the data prior to selection and movement. RMF reports can be examined to find the I/O activity by volume. Volumes should then be ranked by the amount of activity they support, giving the most active volumes preference for movement to the SVA subsystem.

The same method can be used to select volumes for caching, if it has been determined that only selected volumes should be specified as cached volumes. Specify volumes with high I/O rates over volumes with low I/O rates to take advantage of improved performance from caching.

The I/O activity profile indicates if specific volumes have excessive I/O activity and queuing or reserves in a shared DASD environment. Those volumes can be analyzed further to determine the most effective way to spread the data across two or more functional volumes. Using more functional volumes to separate concurrently-accessed data sets minimizes CB queuing and volume reserves and improves the performance of the I/O subsystem. Single host queuing is reported as the IOSQ in RMF. Reserve delay is reported as PEND in RMF.

Minimizing SVA's net capacity load is an additional benefit of this method of data migration. Data movement tools that work at the data set level do not move unused allocations. In fact, they can reduce the size of over-allocated data sets, if desired.

Preparing for the Movement of Data

The major considerations for selecting data for migration involve the trade-offs between the desired performance level and the data capacity of the subsystem. Remember that the fault-tolerant design of the SVA subsystem protects the data stored on it. This may be the most important consideration when selecting data for migration.

It is particularly important to understand the I/O activity of application data to select candidates that will realize the greatest benefit from migration to an SVA (refer to "Examine Data Access Characteristics" on page 49). Give consideration to the potential benefit of spreading the data across more functional volumes.

Due to the disruptive nature of migrating system data sets, they should be the most carefully planned. Careful planning minimizes the time required to perform the migration and minimizes the risk of an error occurring during the migration process.

Placing Data Sets on an SVA Subsystem

Data Set Segregation

Placement of data on functional volumes is less critical than data placement has been on traditional DASD volumes. Traditional volumes have one actuator to transfer data to and from a volume. This creates data transfer bottlenecks, which adversely impact performance. Spreading active data sets across volumes has been the usual method for addressing this problem.

In an SVA subsystem, functional volumes are associated with multiple physical actuators. The data for any functional volume can potentially be on any or all physical devices in the subsystem. Data transfers from the disk arrays to the Controller occur on several paths simultaneously. This design minimizes the I/O bottlenecks within the subsystem and reduces (but does not eliminate) the concern for data set placement on particular functional volumes.

An I/O queuing study performed in the planning stage (refer to "Examine Data Access Characteristics" on page 49) may indicate that there is a need to separate certain data sets because of concurrent activity to a volume. Such problems are indicated by excessive IOSQ time or PEND time on a device or UCB. Further investigation of the data set activity on that volume will indicate which data sets should be spread across multiple UCBs. The SVA subsystem's ability to support up to 1024 functional volumes allows for much more data set separation than has previously been feasible.

System Data Sets

Data sets in this category include operating system data sets, catalogs, paging data sets and spool data sets. All of these data sets benefit from the fault-tolerant design of the SVA subsystem. Dual redundancy provides system data with protection from hardware errors that are a serious concern with traditional disk media.

IPL volumes gain little in performance, except during the initial program load process where sequential loading of the control program dominates the I/O activity. SVA performs sequential read ahead when sequential processing is detected, ensuring that the next data requested is available in cache; therefore, the operation occurs at channel rates, not device rates.

Catalog placement on an SVA is less critical than on traditional DASD. To minimize contention for volume access, catalogs are commonly isolated onto volumes, which have very little other activity. Functional volumes in an SVA have multiple access paths because they are mapped onto multiple physical disks. Given that, it is still important to monitor and control the amount of activity to the functional volume because of the possibility of UCB queuing. UCB queuing should be kept to a minimum on catalog volumes to avoid a system-wide bottleneck, but higher overall activity can be supported on SVA functional volumes than can be supported on traditional volumes.

JES2 and JES3 spool volumes benefit greatly when they are placed on an SVA subsystem. Much of the data that is read and written to spool volumes is processed sequentially, benefitting from cache access, especially for read processing. The longer channel command word (CCW) chains that are used to read and write data to the spool data sets benefit from DASD fast write. The checkpoint data sets, in particular, benefit from DASD fast write because of good locality of reference. Fault tolerance and dual redundancy may also eliminate the need for duplexing the checkpoint data set, reducing I/Os for checkpoint updates.

Page and Swap data sets perform on an SVA subsystem much the same way that they currently do on a 3990 subsystem. The data access on paging data sets is random and involves relatively small quantities of data (single 4K blocks). This type of data access does not benefit greatly from the sequential pre-fetch, DASD fast write, and caching features of SVA. However, page data sets still benefit from dual redundancy in an environment where availability is of prime importance.

Access to high activity load libraries is much improved (unless Virtual Fetch is used) on SVA. The directory of high activity load libraries are likely to remain in cache.

Application Data Sets

All application data sets benefit from residing on an SVA subsystem, some more so than others. Dual redundancy provides a high degree of continuous availability for all data sets without the overhead and cost of dual copy. In addition, most data set activity benefits from the advanced caching features.

Work and temporary data sets, if active enough, may never be written to the functional volumes to which they are directed. This can have a strong positive effect on throughput of tasks that use temporary data sets. Work data sets are usually accessed sequentially. Data that is written to the functional volumes are retrieved very quickly because of sequential pre-fetch.

VSAM data set definitions and attributes should be reviewed when moving VSAM data to an SVA subsystem. Some of the performance options that are used on traditional DASD are no longer needed in an SVA environment, though they do still work. These options specifically include REPLICATE and IMBED.

The REPLICATE option is a VSAM performance solution directed at the traditional DASD architecture. It attempts to minimize the effects of rotational delay on data transfer. Data transfer from an SVA disk begins immediately once the correct track is located, eliminating RPS miss, and minimizing rotational delay. In addition, a read request for a particular record causes the entire functional track containing that record to be staged into cache. Although there is little to no advantage to using REPLICATE and IMBED on an SVA subsystem, there may be some advantage to avoiding it. Maintaining the index as a discrete data component increases the probability of a cache hit. If the index is imbedded in the data or is replicated, the probability of a cache hit is reduced for both the data and the index. In addition, REPLICATE increases the net capacity load of the subsystem.

Space Management

With an SVA subsystem, the physical disk array space is managed automatically. However, DASD space management tasks used to manage traditional DASD space should also be used to manage SVA's functional volumes. The necessity to perform space management tasks does not differ with an SVA subsystem. Tasks such as volume de-fragmentation, idle space release, data set reorganizations, etc. are still important to manage an SVA subsystem's functional space. For additional information on how data is stored in an SVA subsystem, refer to the *Shared Virtual Array Reference.*

Volume De-fragmentation

As data sets are allocated and deleted, the DASD space available for data set allocations becomes fragmented. Even though there may be adequate space available for new allocations on a fragmented DASD volume, that space is often not available in sufficient contiguous amounts. In a traditional DASD subsystem, management of space fragmentation often means the physical movement of the data to consolidate the available free space. Typically, the storage administrator executes a special utility that performs the consolidation of the fragmented space.

With an SVA subsystem, the functional volume VTOC does not reflect the space available in the disk array but rather the space available on the functional volume. As data sets are allocated and deleted, the functional volume becomes fragmented. This fragmentation can cause "X37" type abends. Functional volume fragmentation can be minimized by defining 256 maximum size functional volumes. This allows data set allocations to occur in larger contiguous areas, thereby reducing the necessity of de-fragmentation operations. However, functional volume fragmentation can occur and a de-fragmentation utility should be executed to consolidate the fragmented functional volume.

For a discussion on how data is stored on an SVA subsystem, refer to the *Shared Virtual Array Reference.*

For a discussion on how data is stored on an SVA subsystem, refer to the *Shared Virtual Array Reference.*

Data Archival Strategies

There are many storage management utilities that allow for the migration of inactive data from primary DASD storage to a secondary storage medium. Typically, there are two levels of storage for less active data. Commonly, the first level of storage is DASD and the second is tape. These levels are often referred to as ML1 and ML2. Data migrations often use a host resident data compression routine to provide a more effective utilization of the secondary storage medium.

Small Data Set Packing (SDSP) is a method employed by DFSMShsm that increases the effective utilization of ML1 volumes. The SDSP function stores user data sets as logical records within VSAM key-sequenced data sets (KSDS), thereby allowing more data sets to be stored per track than with the standard DASD allocation unit. Those data sets that are not migrated into an SDSP data set are stored on ML1 DASD space allocated in tracks. Migrated data sets stored in an SDSP data set are stored in space increments of 2 KB. an SVA subsystem automatically stores all data in 2 KB space increments. Using SDSP on an SVA subsystem is inefficient and costly in terms of the DFSMShsm overhead (compression, I/O to search for candidate small data sets, maintenance of the SDSP data set, I/O to the DFSMShsm control data sets, etc.). *The use of SDSP with a Shared Virtual Array subsystem has no resultant benefit and is not recommended.* The elimination of SDSP provides some reduction in CPU cycles and DASD I/O.

SVA provides increased data availability and data compression/compaction outboard of the host. The following recommendations establish an inactive data set migration strategy that takes advantage of SVA's characteristics:

- Do not use SDSP for data migrations to an SVA subsystem ML1 pool.
- Do not migrate inactive data sets residing on SVA ML0 volumes to an ML1 pool.

DFSMS, CACHE console command

Automated storage management eliminates the requirement to manage data at the volume level. With DFSMS, management of data can be accomplished at the data set level to achieve availability and performance objectives. The SVA subsystem presents an alternative to traditional DASD storage subsystems that exceeds today's availability and performance objectives in a cost-effective manner. This portion of the manual addresses some considerations for an SVA subsystem in a DFSMS environment and explores ways in which DFSMS can exploit an SVA subsystem.

Data Class

The data class construct defines allocation attributes for a data set. an SVA subsystem can emulate most 3380- and/or 3390- type DASD devices. With an SVA subsystem, data class can be used as though the data set allocation were to traditional DASD. However, fewer data class constructs are necessary, because SVA allows larger allocations for all files.

Storage Class

The storage class construct describes the service levels for a data set. For the data set to be managed by DFSMS, a storage class must be assigned to the data set. The storage class construct identifies the data set requirements for availability, guaranteed space and performance. Critical data that requires continuous availability should be placed on an SVA subsystem.

SVA does not support dual copy commands. The use of dual copy is not recommended with SVA subsystems because it uses storage resources inefficiently with no resultant benefit. Careful consideration should be given to designing storage class and storage group automatic class selection (ACS) logic to prevent dual copy allocations on an SVA subsystem. Data sets with the Availability parameter set to Continuous should be directed to a storage group containing a 3990 with dual copy enabled. Failure to properly structure the storage group ACS logic causes the data set allocation to fail.

When the DFSMS is active, caching at the data set level can occur dynamically with the 3990. There are three classifications a data set may have for dynamic caching:

- NEVER-CACHE
- MAY-CACHE
- MUST-CACHE

Data sets are classified based on their millisecond response (MSR) values defined in the storage class.

Table 3-5 Dynamic Data Set Caching Classifications

Operation	MSR=999 BIAS=Blank R,W	MSR=blank BIAS=Blank R,W	25=<MSR<999 BIAS=Blank R,W	MSR=blank BIAS=W	MSR<25 BIAS=Blank R
Cache Read	NEVER-CACHE	MAY-CACHE	MAY-CACHE	MUST-CACHE	MUST-CACHE
DASD-Fast Write	NEVER-CACHE	MAY-CACHE	MAY-CACHE	MUST-CACHE	MUST-CACHE

At data set open time, cache usage is determined by the MSR value defined in the storage class. Data sets classified as MAY-CACHE use 3990 cache, if cache is not over-utilized. MUST-CACHE data sets always use cache. Data sets classified as NEVER-CACHE never use cache in a 3990 subsystem. DFSMSdfp periodically collects 3990 subsystem information (DINTERVAL default value is 150 seconds) to ascertain the utilization of cache and NVS. If the 3990 cache or NVS is over-utilized, DFSMSdfp progressively restricts MAY-CACHE data sets from cache. Conversely, if cache and NVS are under-utilized, DFSMSdfp allows previously restricted MAY-CACHE data sets to use cache.

To monitor cache utilization in a DFSMS environment, the DFSMS, CACHE console command can be used. This command provides information for each 3990 cache controller that has at least one DFSMS-managed volume attached to it. The information provided includes:

- The number of DFSMS-managed volumes attached to cache
- The percent of MAY-CACHE reads allowed to use cache
- The percent of MAY-CACHE writes that are allowed to use the DASD fast write feature
- The read hit ratio expressed as a percentage
- The number of DASD fast write waits per minute.

Because the SVA subsystem is a cache subsystem and all writes are DASD fast writes, the storage class construct can be simplified for data residing on an SVA subsystem. For the majority of data residing on an SVA subsystem, the MSR values and bias values should be left blank. For data sets that are MUST-CACHE, the MSR value should be less than 25 milliseconds. The caching algorithms of the SVA subsystem are modified to avoid unnecessary over-usage of cache memory for those data sets that are classified as NEVER-CACHE. When caching is restricted by DFSMSdfp, the MAY-CACHE data sets inhibit CACHE LOAD SET in the DEFINE EXTENT CCW SET to prevent the data set from being loaded into cache. With an SVA subsystem, the restriction of cache by DFSMSdfp results in the modification of the caching algorithms for the MAY-CACHE data sets to provide performance similar to non-cache. To exploit SVA subsystem caching, all functional volumes should have cache turned on. This allows those data sets that are MAY CACHE to utilize the SVA subsystem cache more effectively. As with the 3990-3, using the DFSMS CACHE command provides insight into the utilization of the SVA subsystem cache in a DFSMS environment.

Management Class

The management class construct identifies a data set’s expiration, retention and migration criteria. DFSMShsm is a storage management software utility that performs data migration. The use of DFHSM can be significantly enhanced by an SVA subsystem. The following discussion refers to DFSMS, but the same concepts are applicable to other data management products.

With traditional DASD subsystems, DFHSM uses a significant amount of processing resources during the data migration and recall process. When DFHSM data compression is active, CPU cycles are expended to compress and decompress data sets. DASD space consumption increases due to the addition of ML1 volumes. I/O rates increase due to the data migration/recall process. Lastly, access to migrated data is slowed as a result of the recall process.

With an SVA subsystem, there is a significant potential to reduce the amount of resources that DFHSM requires. Because an SVA subsystem compresses and compacts data outboard of the host, an ML1 pool should be defined to an SVA subsystem. Establishing a DFHSM ML1 pool comprised exclusively of SVA functional volumes allows DFSMShsm data compression to be disabled on the host. Data migrations from traditional DASD subsystems to an SVA ML1 pool can occur without the associated DFSMShsm compression overhead. DFSMShsm data compression for ML2 can be eliminated. The elimination of DFSMShsm data compression for ML1 and ML2 minimizes the CPU cycles consumed by DFSMShsm data migration/recall and backup/recovery operations.

The DFSMShsm command to disable host compression for ML1 and ML2 is:

```
SETSYS COMPACT(NONE)
```

To further exploit an SVA subsystem, DFSMShsm data migrations to ML1 should be eliminated for data sets residing on an SVA subsystem. CPU cycles consumed by DFSMShsm are further reduced by the elimination of the ML1 migration process for data sets residing on an SVA subsystem. To implement this strategy, set the LEVEL 1 DAYS NON-USAGE parameter in the management class to 0 for the data sets residing in the storage group consisting of the SVA subsystem. Because of SVA's compaction and compression, it is unnecessary to migrate data to ML1 to increase primary DASD storage capacity.

Storage Group

In typical DFSMS implementations, storage groups are mapped across various DASD devices to optimize availability and device performance. The fault tolerance and redundancy of all data on an SVA subsystem makes dual copy unnecessary. With an SVA subsystem, the process can be reduced to two steps:

1. Define a storage group with SVA functional volumes.
2. Design ACS routines that direct both critical and noncritical data to the storage group containing the SVA functional volumes.

When defining storage groups for an SVA subsystem, segregation of data based on availability requirements is no longer necessary.

There are two common approaches to configuring storage groups for performance: horizontal and vertical mapping.

The horizontal approach simply maps a storage group to a number of different device types with varying performance characteristics.

The vertical approach maps a storage group to devices that are similar and have similar performance characteristics. The horizontal approach is common when the DASD is expected to be under-utilized or where a major portion of the DASD is attached to cache controllers. The vertical approach is used mainly where there is a limited number of cache controllers available.

One objective of DFSMS is to reduce the number of storage groups or DASD pools that a storage administrator manages. an SVA subsystem with extended capacity and Dynamic configuration should be defined with

256 maximum size functional volumes. With data dispersed across all 256 functional volumes, UCB queueing is reduced and out-of-space conditions are be minimized. Extended capacity and Dynamic configuration allows for the establishment of a single storage group comprised of SVA functional volumes. The creation of a single storage group for data sets residing on an SVA subsystem provides increased data availability and performance while reducing the number of storage groups the storage administrator manages. When multiple storage groups must be defined to an SVA subsystem, the horizontal approach should be used to take advantage of the SVA subsystem's parallelism in channel operations and to maximize dispersal of data across functional volumes. Volumes that are not typically managed by DFSMS (IPL volumes, ML1 volumes, etc.) can co-exist with a storage group defined to an SVA subsystem. Because an SVA subsystem provides fault tolerance and redundancy, as well as exceptional performance characteristics, the need to segregate data based on availability and performance requirements is virtually eliminated.

Volume Conversion

Volume conversion with an SVA subsystem is accomplished in the same manner as with a traditional DASD subsystem.

This page intentionally left blank

Chapter 4 SVA in a VM Environment

The Shared Virtual Array (SVA) subsystem is fully compatible with the VM/ESA operating systems. It functions as up to four 3990-3 storage control units and requires no special software. However, there are some special considerations in defining and allocating an SVA subsystem under VM.

Installing Shared Virtual Array for VM

After installation, as with any other I/O device, an SVA subsystem must be defined to the VM operating system in the same manner as other subsystems. Special software considerations for defining the SVA subsystem to the VM operating system, along with a list of supported releases of VM, are provided in this section. Functional disk initialization using ICKDSF is also discussed.

Minimum VM Program Levels

The VM operating system release level required for SVA implementation is dependent on the functional device types used. The minimum prerequisite host software level required to support 3380 functional volumes is:

VM/ESA Version 1, Release 2.1.

The minimum prerequisite host software level required to support 3390 functional volumes is:

VM/ESA Version 1, Release 2.1.

The SVA subsystem requires no additional host software support to emulate the 3990-3. The host operating system, however, must be able to support a 3990-3 storage control unit and the functional device types being defined. Before installing the initial SVA subsystem, install and verify all of the current maintenance recommended in support of the 3990-3, 3380 and 3390 volumes. Check with the support center for the current PSP buckets.

SYSGEN Considerations

The SVA subsystem does not have physical disk volumes in the traditional sense; however, the operating system does access disk-resident data as though it were on traditional DASD. Data volumes, as seen by the operating system, are functional volumes in SVA terms. (Refer to the *Shared Virtual Array Introduction* for a more detailed discussion of functional volumes).

From an operating system perspective, functional volumes are defined just as traditional DASD is defined in the real I/O (RIO) definitions. In the VM/ESA environment, the RDEV macro is used to define a real device block for a functional volume.

You may define a maximum of 1024 functional devices. The primary advantage to defining 1024 functional volumes to VM is that if fewer than

1024 functional volumes are initially configured on the SVA subsystem, more may be added (i.e., defined) at a later date without an IPL. New volumes may be varied online after they are defined to the SVA subsystem.

The primary disadvantage to defining 256 functional devices to VM is the added amount of real storage required for real control blocks for each functional device. The amount of storage required to support the control blocks for 256 real device definitions is:

VM/ESA 102,400 (X'19000') bytes

You should understand these implications for your specific environment. The industry trend has been toward systems with increasing amounts of memory, and the benefits of improved I/O response probably outweigh the cost of used host storage.

For all VM environments, the Shared Virtual Array subsystem should be defined as four 3990-3 storage control units with a full complement of 3380 or 3390-type devices. These VM definitions are made via a real I/O (RIO) regeneration.

Preparing the Hardware

The use of the ICKDSF format command to initialize SVA functional volumes in VM is identical to the process used for traditional DASD volumes. Due to differences in the SVA architecture over conventional DASD, there are some differences in the recommendations for formatting. Refer to "Preparing a Functional Volume" for a full discussion on initializing functional volumes using ICKDSF.

VM Cache Operation

DASD fast write (DFW) and cache fast write (CFW) are both fully supported by the SVA subsystem. All writes in an SVA subsystem are DASD fast writes. All data transfers from the cache to the disk array are performed asynchronously in the background.

The implementation of cache fast write in the SVA subsystem is slightly different from the 3990-3. The SVA subsystem acknowledges and records cache fast write commands. All data transfers are performed exactly the same as for DASD fast write. The SVA subsystem indicates a successful write before the data is transferred to the disk array. Unlike the 3990, data is written to both cache and 0 nonvolatile storage. This ensures that there will always be a secure copy of the data under any circumstance.

The CP SET CACHE command is used to activate or deactivate caching by functional device address or by entire subsystem. From the VM perspective, the CP QUERY CACHE command is used to display the caching status of the SVA subsystem.

Note: The SVA subsystem always operates in a caching mode, even if the response from the QUERY CACHE command indicates that caching is off.

Exploiting Iceberg Extended Facilities Product (IXFP) in VM

IXFP is an optional software package that enhances configuration and storage management operations of the SVA subsystem in a VM environment while providing an additional level of cache control. Included in these capabilities are some standard functions such as enabling and disabling DASD fast write (DFW).

In addition, IXFP maximizes the benefits of StorageTek's virtual storage architecture by allowing interactive control of activities such as subsystem administration and reporting.

IXFP has additional reporting capability for SVA and other 3990-compatible cache controllers. IXFP can be used to produce cache performance reports, including information on cache hit ratios and cache occupancy. The information from these reports can be used to effectively manage cache utilization.

The IXFP facilities are described in more detail in the *Shared Virtual Array Introduction* and *IXFP Configuration and Administration*, manuals.

VM Tools

In any VM environment, the standard VM/Monitor can be used to collect resource utilization and capacity planning data at user-specified intervals over desired time periods. Different packages, depending on the level of the VM system in use, are available for the reduction and reporting of this data.

Under VM/ESA, the Virtual Machine Processor Resource Facility (VM/PRF) can be used to analyze and produce standard reports on the collected data.

The Virtual Machine Performance Planning Facility (VM/PPF), which is a combination of VM/PRF with a capacity planning capability, can be used for longer term analysis.

The Real Time Monitor/Systems Facility (RTM/SF) can be used for short-term, dynamic monitoring and analysis.

Storage Management in VM

In VM, a Shared Virtual Array functional device may be initialized and prepared for CP use in either system stand-alone mode (when the software tools used for volume initialization and preparation run on the CPU without benefit of an operating system) or under the control of a VM.

Note: In VM, a prerequisite to SVA functional volume initialization is the definition of the real addresses for the functional volumes HCPRIO (for VM/ESA). The current CP nucleus must reflect these new device addresses prior to software initialization.

Special Utilities - Device Support Facilities

Device Support Facilities (ICKDSF) is an IBM software utility that performs functional volume initialization and disk media maintenance. Volume initialization prepares a functional volume for VM operating system usage. However, the use of ICKDSF for media maintenance on SVA functional volumes does not perform the desired service.

Disk errors that cause an interruption in data availability for traditional DASD are minimized with an SVA subsystem. SVA can automatically correct media errors and dynamically reconstruct a failed disk drive. Refer to the *Shared Virtual Array Reference* for further information on SVA drive reconstruction and automatic media maintenance.

Only an ICKDSF MINIMAL INIT is necessary and recommended for SVA. Refer to “Special Utilities--Device Support Facilities” for a list of the commands that are supported and are not supported by an SVA subsystem.

Defining and Preparing a Functional Volume

Functional volumes that contain a CP system residence volume, or VM paging, spooling, dump, directory, or temporary disk space, must be defined and initialized as CP-owned volumes. A CP-owned volume must contain a CP allocation table with assignments for these system functions at cylinder 0, record 4. Because the CP system references its DASD locations by 4K (4096) byte pages of contiguous disk storage, a special process is required to prepare these functional volumes.

The process to define and initialize a functional volume as a CP-owned volume may be completed by either using the ICKDSF CPVOLUME command or by using the CP format/allocate program from a VM. The steps used to prepare a functional volume as CP-owned by using the CP format/allocate program from VM/ESA are:

1. Use either the CMS command--CPFMTXA--or the CP command--CPFORMAT--to format and allocate the functional volume.
2. Define the CP volume in the SYSCPVOL list located in HCPSYS.
3. Reassemble HCPSYS, then generate the CP nucleus.

All the cylinders of a CP-owned functional volume are not required to be formatted with CP DASD pages; only the cylinders used by CP require formatting. However, cylinder 0 must always be formatted before any other cylinders on a functional volume are formatted.

Data Recovery

SVA recovers from most hardware failures automatically. However, as with any other DASD subsystem, data corruption caused by application programs or user errors must be corrected by the customer.

Data Migration Planning

Migrating data to an SVA subsystem is similar to migrating data to any other form of disk storage. The SVA subsystem is designed for reliability and you should encounter no problems during the data migration process. To enhance the integrity and speed of the migration process, create and document a detailed migration plan prior to any movement of data. Some key components of a complete VM data migration process include:

- A report documenting the current allocations residing on DASD
- A report documenting the utilization rates of the allocated minidisks
- A complete tape backup of data from DASD
- A report documenting the converted allocations from one device type to a different device type emulated by the SVA subsystem (if required)
- A plan to archive/off load any data determined to be not in use
- A description of the impact of data migration from the end-user's perspective
- A report documenting the I/O workload skew across the DASD devices containing data to be migrated to an SVA subsystem
- A plan to recreate the data residing on the original disk in the event that a process back-out is required.

Migration Strategies

The simplest migration method is a direct volume copy from the source volumes to the target functional devices. This strategy is also the most expedient and similar method to existing migration plans. However, this method does not attempt to exploit any of the special features of the SVA subsystem.

The SVA subsystem features that are automatically available, regardless of the migration strategy used, include:

- Fault tolerance - the data is protected from hardware failures.
- Data compression - the data on an SVA subsystem is compressed upon entering the SVA Controller. This strategy reduces the amount of cache and physical space required to store the data.
- Data compaction - inter-record gaps are removed, further decreasing the amount of space required to store the data. Additionally, RPS miss is eliminated.

Minidisk Migration Strategies

Several software tools may be used to perform the actual data migration process in a VM environment:

CMDISK - A DIRMAINT command that moves the data residing on a minidisk from one device type to the same or another device type. DIRMAINT is an IBM licensed program product.

COPYFILE - A CMS command used to copy files residing on a minidisk from one device type to the same or another device type.

SPTAPE - A CP command used to dump spool files to tape, and load spool files from tape to a functional device.

While each of these software tools has different characteristics, the following steps can be identified as required for a complete migration of a DASD volume to an SVA functional volume:

1. Confirm that a valid VOLSER has been written to each functional volume
2. MDISK statements in the VM directory reflect allocations on the functional volume. Actual data is migrated to the minidisks on the functional volume(s).

While these software tools (with the exception of SPTAPE) use a disk-to-disk copy process, it is also possible to use a disk-to-tape-to-disk migration process. Regardless of which process is used for data migration, it is recommended that a reliable backup copy of the data to be migrated be maintained until the migration process is complete and minidisk integrity has been validated.

The selection of the software tool to use for data migration depends on the environment in which SVA is being implemented. Consult the table below to determine which tools will optimize the migration process in your environment:

Table 4-1 Migration Process Tools

ACTION	CMDISK	COPY-FILE	DDR	SPTAPE
Used to migrate CMS formatted minidisk data	Yes	Yes	Yes	No
Allows for altered minidisk allocation sizes between DASD and SVA	Yes	Yes	No	Yes
Automatically creates CMS format results	Yes	No	Yes	No
Automatically updates VM directory with functional volume minidisk volumes	Yes	No	No	No
Copies a physical device volser to functional volume	No	No	Yes	No
Migrates data between unlike device types (such as 3380 to 3390)	Yes	Yes	No	Yes
Migrates more than one minidisk per command input	Yes	No	Yes	Yes

Migration Using DIRMAINT CMDISK

DIRMAINT CMDISK (change minidisk) is a DIRMAINT command that moves the data residing on a minidisk from one device type to the same or another device type. (DIRMAINT is an IBM licensed program product.) When the DIRM CMDISK command is issued, the following tasks are performed:

1. A new minidisk allocation, on a functional volume, for the CMDISK cylinder specifications is automatically created using a dummy virtual address for the DATAMOVE VM.
2. The user's existing minidisk is automatically reassigned to the DATAMOVE VM at another temporary address.
3. The new minidisk allocation on the functional volume is automatically formatted with the CMS FORMAT command.
4. All the files on the existing minidisk allocation are copied to the new minidisk allocation on the functional volume.
5. The space allocated for the existing minidisk allocation is returned to the general pool of available minidisk space, and the new minidisk allocation on the functional volume is reassigned as the existing minidisk for the target userid.

Migration Using COPYFILE

The CMS COPYFILE command can migrate a copy of the CMS files residing on a single minidisk allocation to another minidisk allocation. With the COPYFILE command, the input and target minidisks may reside on different device types (such as 3380 to 3390). For further information on the CMS COPYFILE command format, consult the appropriate CMS command reference document for your operating system environment.

Migration Using DDR

The DASD Dump Restore program can be used to perform a physical copy of a DASD volume to a functional volume. The DDR program can be executed stand-alone on the CPU (without benefit of the VM operating system) or under the control of a VM.

DDR requires the input and target volumes to be the same device type (such as 3380 to 3380). For this reason, an SVA functional volume must be defined as a 3380 to receive migrated data from a 3380 volume via DDR. Correspondingly, an SVA functional volume must be defined as a 3390 to receive migrated data from a 3390 device via DDR. No other device configurations of data migration are supported with DDR.

A series of DDR commands used to perform the physical migration of a volume can be placed in a control file. The control file name can then be specified in the DDR command to automatically execute the DDR statements in the control file. If a disk-to-tape-to-disk migration strategy is desired, DDR can facilitate such a strategy, limited only by the device type constraints previously described.

Migration Using SPTAPE

The SPTAPE command is used to move spool files from disk to tape, and load spool files from tape to disk. The input and target device types may differ. Spool files migrated to another device are assigned a new spoolid to avoid a duplication conflict within the VM spooling system. For further information on the CP SPTAPE command format, consult the appropriate CP command reference document for your operating system environment.

Examine Data Access Characteristics

A more comprehensive process for migrating data to an SVA subsystem involves studying and understanding several activity and utilization characteristics of the current DASD configuration. These characteristics include:

I/O workload skew - Minimizing the workload skew provides a performance benefit by reducing the I/O contention for a functional volume. Determine the percentage of I/O activity submitted to each volume over several time intervals. I/O data saved from the online VM monitor can be used to analyze workload skew by device. Volumes should then be ordered by the amount of activity they currently support, with the most active volumes given the highest priority for spreading across multiple functional devices.

Seek skew - Understanding the seek skew assists in determining appropriate volumes for caching. Minimal seek skew offers the highest probability of a cache hit and therefore better performance. Determine the percentage of zero seek skew on a volume and minidisk basis. One strategy to consider is migrating minidisks with high zero seek skew rates to SVA functional volumes designated as cache volumes.

Capacity utilization - By evenly distributing minidisk allocations across functional volumes, greater environmental flexibility is offered to the process of adding new minidisks to an SVA subsystem. Determine the percentage of the existing DASD capacity that is currently allocated, as well as the aggregate utilizations of the existing allocations. Data migration should be considered as an opportunity to review the current utilization of existing allocations and to modify these allocations based upon under- or over-utilization.

Using more functional volumes allows the reduction of access delay due to device queueing and improves the performance of the I/O subsystem. Understanding the I/O activity of the volumes to be migrated to an SVA subsystem maximizes the benefits of data migration.

Preparing for Data Migration

The major considerations for selecting data for migration involve the trade-offs between the desired performance level and the data capacity of the subsystem. Remember that the fault tolerant design of an SVA subsystem protects data stored on it. This may be the most important consideration when selecting data for migration.

Data movement should be planned to minimize the risk of data corruption or loss. The highest chance for data loss to occur is through human error or misunderstanding.

The first stage of migration should emphasize data that benefits significantly when placed on an SVA subsystem, but for which impact is low if unexpected problems are encountered during the migration process. Data in this category includes general user minidisks and relatively inactive system data, such as CP and CMS source and aux file minidisks. These minidisks are most easily managed by systems programming personnel.

The next group of data to be moved may be application minidisks. It is particularly important to understand the I/O activity of these minidisks to select candidates that will realize the greatest benefit from migrating to an SVA subsystem (refer to “Examine Data Access Characteristics”). Application data can be the most sensitive data from the users’ perspective. It realizes the greatest performance benefit from movement to an SVA subsystem.

Due to the disruptive nature of migrating system minidisks, these migrations should be the most carefully planned. Such added consideration minimizes the time required to perform the migration and minimizes the risk of an error occurring during the migration process.

Some of the most active system minidisks benefit the most from migration to SVA. The MAINT minidisks linked to by VMs (190, 19D, 19E, etc.) show marked performance improvement from the fault tolerant aspects of the SVA subsystem.

Placing Minidisks on an SVA Subsystem

The placement of minidisks on functional volumes under SVA is less critical than data placement on traditional DASD. This is because traditional DASD has a single actuator between the stored data and the control unit, creating data transfer bottlenecks that adversely impact performance. Spreading active minidisks across volumes is the usual method for addressing this problem with traditional DASD.

With an SVA subsystem, functional volumes are associated with multiple paths and actuators. The data transfer from the disk arrays to the Controller occurs on several paths simultaneously. The data for any functional volume can potentially be on any or all physical volumes on the array. This design minimizes the I/O bottlenecks within the subsystem, and reduces the concern for minidisk placement on particular volumes.

The I/O queueing study performed in the planning stage (refer to “Examine Data Access Characteristics”) may have indicated a need to separate certain minidisks because of concurrent activity to a volume. A problem condition may be demonstrated by a repeated occurrence of device busy for I/O targeting specific minidisks. Additional investigation of the minidisk activity on the volumes to be migrated will indicate which (if any) minidisks should be allocated to different functional volumes. The SVA subsystem’s ability to define and support 1024 functional volumes allows for a high level of flexibility in minidisk distribution.

IPL Volumes

Minidisks containing IPLable data, such as MAINT 190/19E, have good performance characteristics because of SVA's sequential detection and pre-staging. System areas such as the CP directory and saved system areas also benefit in this manner.

Common CMS Minidisks Linked by other VMs

Minidisks that are linked at VM logon by more than one VM, such as MAINT 19D, PROFS, or language compilers, are excellent candidates for disk caching.

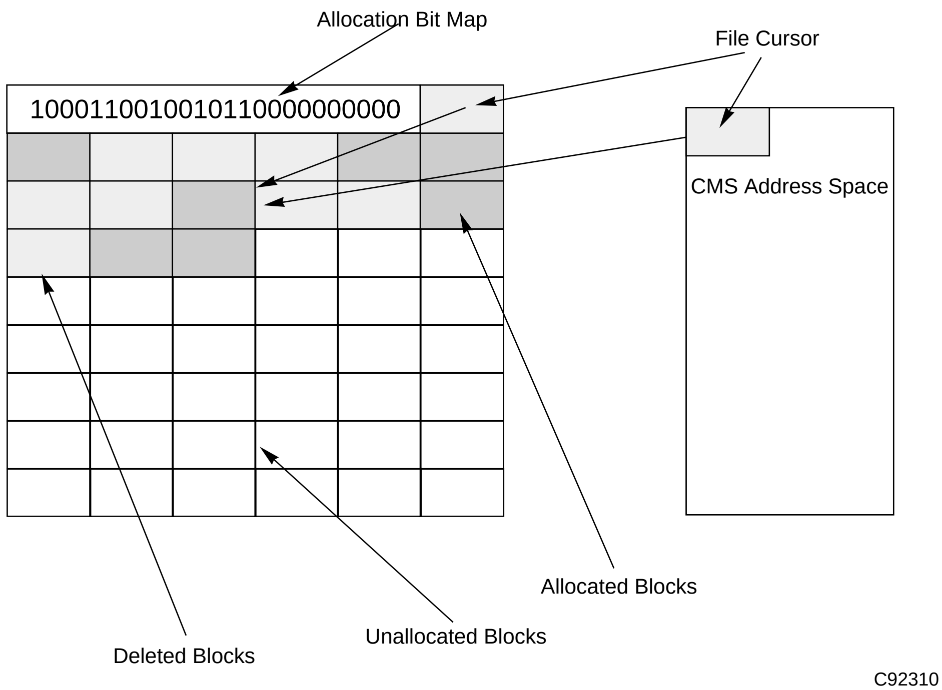


Figure 4-1

The CMS minidisk format traditionally consists of 4KB blocks that are written to the minidisk space when the volume is CMS formatted. The virgin tracks of the read DASD device are sectioned off by a relative cylinder allocation in the VM directory, maintained by MAINT. Link to this minidisk as a virtual device address and format this virtual device using the CMS format command. This command results in 4K block records being written to the cylinders defined. Thereafter, writing to a CMS minidisk is done using only update write commands. The CMS 4K block structure is not altered by writing to the disk. A file cursor moves through the blocks and points to the next available block for allocation. In VM/ESA, this cursor does not reset when files are deleted, so new blocks will be written until all of the blocks in the entire CMS minidisk are used. Then, the cursor resets to the top and begins rewriting blocks that have become available by deleting files in the minidisk.

IXFP Development has taken several steps to improve space management in the VM environment. They are as follows:

1. Release MiniDISK

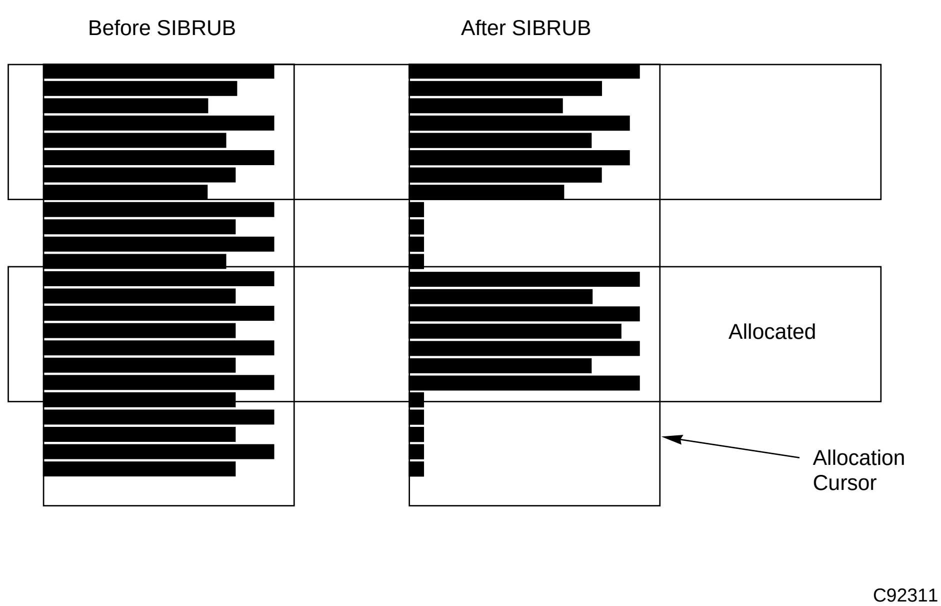
   You may have multiple concurrent Release sub-commands being executed to the same or different devices. Each command is a unique task and should complete normally. The VM sub-commands performs no VTOC mapping.

   The Release MiniDISK sub-commands queries the real device and will be provided with the cylinder boundaries for that disk. The FROM and FOR option are not needed, but could be used if the link was a full-pack minidisk. Use caution, because this command will revoke access to data.

2. SIBRUB Utility

   The SIBRUB Utility helps SVA perform space management by allowing the blocks that CMS does not use to be rewritten with blanks. Blanks compress at a rate of about 25:1, so when these blocks are rewritten to SVA, they will take a minimum amount of backend space. It does not use binary zeros, because VM detects that and will not actually do the write, but will instead mark the block and leave the data. Blank characters actually get written so the resulting compression and dynamic allocation in SVA will release 90 to 95% of the backend space being used to store the un-allocated data. This reduces the physical capacity for the CMS minidisk and will reduce SVA's net capacity load (NCL).

Rewriting Unused Blocks



You must have R/W access to release the minidisk. Unused blocks are reformatted with blanks and require fewer sectors to store the data. Used blocks are left untouched so data IS NOT destroyed on the minidisk. Periodic use of SIBRUB reduces the physical capacity required in a VM environment. Volumes that have much update activity will benefit the most from this function. Idle volume will show little benefit.

The SIBRUB utility rewrites un-allocated blocks with blanks. SIBRUB rewrites functional track records that contain CMS file data but are not currently allocated. The resulting compression of data for these functional tracks replaces the full tracks with very compressed tracks. SVA's free space collection (FSC) routines can then collect this space and make the space available to the SVA subsystem free space pool for new data storage.

3. SIBVMCUR Utility

   The SIBVMCUR utility is a CMS nucleus extension that causes a CMS allocation cursor to reset any time a CMS ERASE command is issued to a minidisk. This causes CMS to reuse blocks just deleted, rather than move forward in the CMS minidisk until all blocks are consumed.

4. Query MiniDISK Function

   The Query MiniDISK function allows monitoring of the CMS minidisk space utilization. You require authority to perform DIAGNOSE E4 Functional 1 and 3; you require the DEVMAINT option in the user's directory entry to perform the function to other userids.

   You can query any minidisk that is linked if you have class 'B' authority to use DIAGNOSE E4. You may query another user's minidisk using the USERID parm.

   You must be logged on if the DIRECTORY option is set to NO, but if the DIRECTORY option is set to YES, then DEVMAINT authority is required to query the directory entry for you to find the MDISK definition and query the space usage on that minidisk.

   The output message shows the real device address, the SVA subsystem name and FDID, the cylinder allocation and physical space used, as well as the compression ratio for that data.

   Query MiniDISK is only valid for SVA functional volumes and cannot be used to query space on traditional DASD devices.

Page/Spooling Minidisks

Page/spooling minidisks are not good candidates for caching. In fact, overall VM system performance may be degraded if these minidisks are cached.

VM NCL Management

The VM space management in IXFP Release 1 was a facility to release space for deleted minidisks. The value of this utility was minimal, because many VM maintenance packages already clear space on deleted minidisks by rewriting them with zeros. This reduced SVA backend storage so that the release minidisk command had very little space that it could recover. What was really needed was a facility to reduce storage space when files within a CMS minidisk were deleted.

TDISK usage proves to be very efficient on SVA. The TDISK CLEAR option will cause all TDISK space to be rewritten when TDISKS are released by users. This results in good space management by SVA. TDISK formats also take less time on SVA than on traditional DASD.

The TDISK clear option will have some overhead since it will cause the rewriting of this space. For shops that have large amounts of TDISK, this may be a consideration before enabling this option. A TDISK Clear Utility will enable users who do not wish for this to happen automatically but would rather submit a utility at an off-peak time to do so.

The TDISK Clear option will cause all TDISK space to be rewritten when TDISKs are released by users. The following screen is how you set the TDISK clear option in effect. The option is specified in the SYSTEM CONFIG file and invoked at initial program load (IPL). TDISK must reside on a CP-owned volume and the TDISK space is defined in the directory on MAINT and allocated to the CP-owned disk when it is formatted with CPFMTXA.

This page intentionally left blank

Chapter 5 SVA in a Co-Located Environment

Connectivity Terms

There are three terms that are commonly thought to mean the same thing - they don't.

- *Host Path* - This is a StorageTek term coined because an SVA always presents group of four Control Unit (CU) images to a host. A host path is the communication path between a host and all of the 9500 SVA's logical devices (up to 256 logical devices) in each control unit image (for a maximum total of 1024 logical devices). Each physical connection to an SVA - the ESCON cable - may contain up to eight host paths in an eight card configuration[1]. There are two ESCON ports on each ICE card, so with all eight ICE cards installed, there could be 128 host connections to an SVA (see Figure 5-1 on page 74 and Figure 5-2 below). There are four sets of 256 devices in an SVA, each host path consists of four logical paths.
- *Logical Path* - A logical path is a connecton between a channel image and a control unit image. Every logical path has an associated physical path, but a physical path may contain multiple logical paths (up to 32; 512 logical paths to the SVA, divided by 16 links per ICE card equals 32 logical paths per physical link). A host connected to an SVA by one physical link may have up to four logical paths over that link - one to each of the four 3990 control unit images within the 9500. An SVA, with all eight ICE cards installed, could then have 512 logical paths connected to it.
- *Physical Path* - A physical path between a channel and a control unit is the set of one or more links (ESCON cables) which make up the connection between the channel and the control unit. In the case of a channel connected directly to a control unit, there is one link, and the terms "link" and "Physical path" are synonymous. Each end of the physical path has an associated, unique, address, called a link address.

1. With a mainframe attach, ICE cards are sold in configurations of four or eight cards. With SCSI attach alone, there is a minimum configuration of two ICE cards allowed.

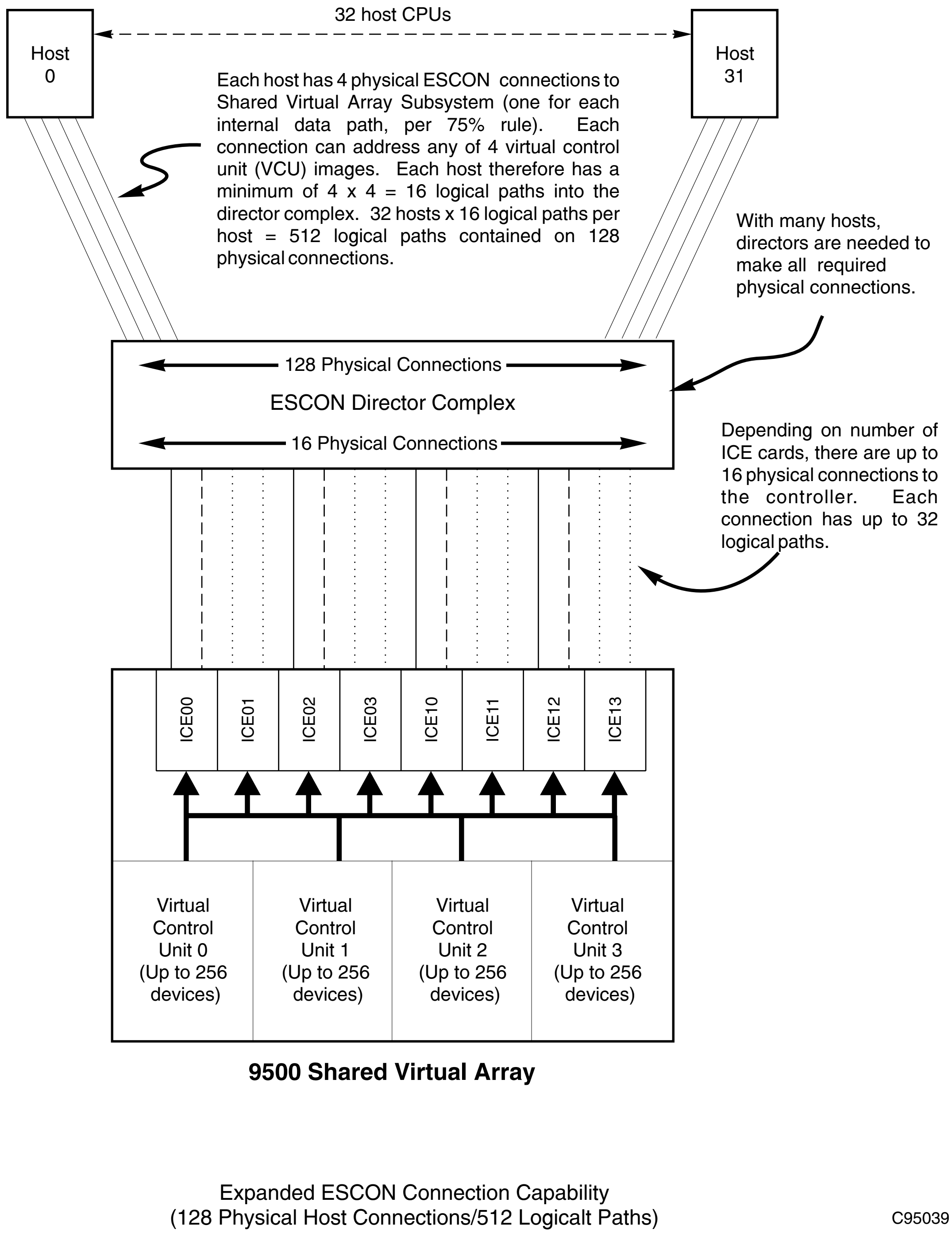


Figure 5-1 Path Connections

Host Path Connections

The first consideration when operating the SVA in the co-located environment is the reduction in the number of host paths available to a mainframe host. With the allocation of ICE cards to the open systems host, there is a reduction of the number of host paths availble to the mainframe as shown in below.

ESCON Channels

The 9500 is set to dual link mode - eight logical host paths per ICE card port.

Figure 5-2 depicts a fully populated 9500 with ESCON attach to an OS/390. It shows eight logical host paths per port on the ICE card (there are two ports on an ICE card) and all eight ice cards installed (8 ICE cards X 2 ports per card X 8 hosts paths per ICE card port = 128 host paths).

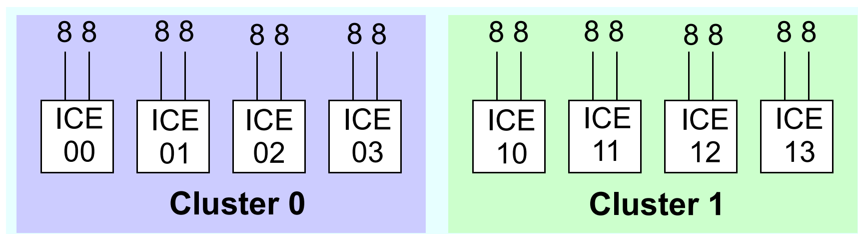


Figure 5-2 A Fully Populated 9500 with ESCON Attach Only

Figure 5-3 depicts a fully populated 9500 with ESCON attach to an OS/390 on six of the ICE cards and a SCSI host attach on two of the ICE cards. The host paths now available to the OS/390 would be the number of ICE cards attached to the OS/390, multiplied by the number of ports used (two), multiplied by then number of host addresses available on each port (eight), or 6 X 2 X 8 = 96.

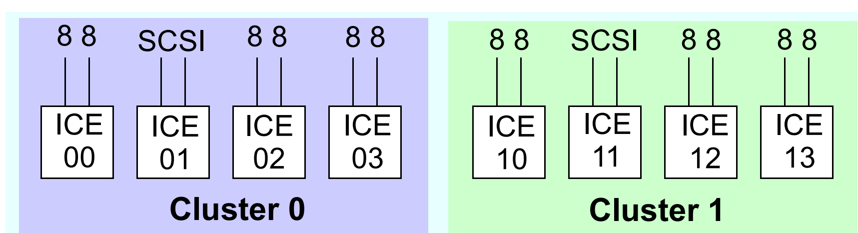


Figure 5-3 A 9500 with Two ICE Cards Used For A SCSI Connection

Table 5-3 on page 77 shows the number of ports available for a given number of ICE and ICF cards installed.

Table Table 5-1 below shows the various SCSI host bus adapters supported for each host type and the software level required.

Table 5-1 SVA SCSI Platform Matrix

Servers	Bus	Operating System(s)	Host Bus Adapter type
IBM - Mainframe			
Native MVS - OS/390	ESCON	MVS 5.2.2 - OS/390 2.7	N/A
MVS - OS/390 Guest on VM	ESCON	MVS 5.2.2 - OS/390 2.7	N/A
P/390	ESCON	OS/390 2.4	N/A
SUN - Solaris			
Sun Workgroup Servers	PCI	2.51, 2.6	integrated
Sun Mid-range Servers (E3000 - E6500)	Sbus	2.6	1065A
HP-UX (Mid-range)			
J class	Integrated	11	A4107A
K class	HSC	10.2, 11.0	A2965A
IBM - AIX			
F50	PCI	4.3	6207/6209
Windows NT 4.0			
Pentium II, 128 MB of memory, and 1 GB Hard drive or better.	PCI	NT 4.0 with service pack 5.0 or greater	Adaptec AHA-2944UW Differential

Fibre Channels

Fibre channel cards are similar to ESCON cards but have 32 host path connections available per port on the card.

Table 5-2 on page 77 shows the host bus adapters supported for fibre connections and information about those host bus adapters.

Note: The 9500 allows for attachmet of up to 256 LUNs (LUN numbers 0 through 255) on each fibre connection. However, the actual number is often much less than that as per restrictions imposed by the host system and the host bus adapter. Some software work-arounds may overcome some of the before mentioned restrictions.

Table 5-2 Fibre Channel HBA Types

Platform Vendor	Bus Type	Server	HBA Vendor	HBA Product Number	HBA Model Number	#FCA Ports per HBA	HBA Port Media Type	HBA Prot Connector	OS Support	Minimum Driver Level
Sun	SBus		JNI	FC64-1063-N	FC64-1063-N	1	MM	Dual SC	2.6, 2.7	2.4
					FC64-1063-L	1	SM	Dual SC	2.6, 2.7	2.4
	PCI		Emulex	LP8000	LP8000-F1	1	MM	Dual SC	2.6, 2.7	4.0
					LP8000-N1	1	MM (GBIC)	Dual SC	2.6, 2.7	4.0
					LP8000-L1	1	SM (GBIC)	Dual SC	2.6, 2.7	4.0
HP	PCI	V or N Class	HP	A3740A	A3740A	1	MM	Dual SC	11.0	
	HSC	K or J Class	HP	A3404A	A3404A	1	MM	Dual SC	10.2, 11.0	
	HSC	D Class	HP	A3591A	A3591A	1	MM	Dual SC	10.2, 11.0	
RS/ 6000	PCI		Emulex	LP8000-x	See Above				4.2, 4.3	
NT	PCI	X86	Emulex	LP8000-x	See Above				4.0	
	PCI	X86	Qlogic	QLA2200	QLA2200F	1	MM	Dual SC	4.0	
					QLA2200G	1	GBIC	GBIC	4.0	

Note:
1. QLA2200 and 2200F are available with /33 or /66 suffix (pci bus speed in MHZ)
2. QLA2200 does not include the GBIC (Multi Mode and Single Mode are available)
3. MM = Multi Mode and SM = Single Mode.

Table 5-3 ICE2 and ICF Card Matrix

Feature Numbers	# of ICE Cards	# of ICF Cards	Cluster 0				Cluster 1			
			Slot 00	Slot 01	Slot 02	Slot 03	Slot 10	Slot 11	Slot 12	Slot 13
#ES08 + #SR00 + #FM00 + #FS00	4	0	E		E		E		E	
#ES12 + #SR00 + #FM00 + #FS00	6	0	E	E	E		E	E	E	
#ES16 + #SR00 + #FM00 + #FS00	8	0	E	E	E	E	E	E	E	E

Table 5-3 ICE2 and ICF Card Matrix

#ES00 + #SR08 + #FM00 + #FS00	4	0	S		S		S		S	
#ES00 + #SR12 + #FM00 + #FS00	6	0	S	S	S		S	S	S	
#ES00 + #SR16 + #FM00 + #FS00	8	0	S	S	S	S	S	S	S	S
#ES00 + #SR00 + #FM08 + #FS00	0	4	FM		FM		FM		FM	
#ES00 + #SR00 + #FM06 + #FS02	0	4	FM		FM		FM		FS	
#ES00 + #SR00 + #FM04 + #FS04	0	4	FM		FS		FM		FS	
#ES00 + #SR00 + #FM02 + #FS06	0	4	FM		FS		FS		FS	
#ES00 + #SR00 + #FM00 + #FS08	0	4	FS		FS		FS		FS	
#ES00 + #SR00 + #FM12 + #FS00	0	6	FM	FM	FM		FM	FM	FM	
#ES00 + #SR00 + #FM10 + #FS02	0	6	FM	FM	FM		FM	FM	FS	
#ES00 + #SR00 + #FM08 + #FS04	0	6	FM	FM	FS		FM	FM	FS	
#ES00 + #SR00 + #FM06 + #FS06	0	6	FM	FM	FS		FM	FS	FS	
#ES00 + #SR00 + #FM04 + #FS08	0	6	FM	FS	FS		FM	FS	FS	
#ES00 + #SR00 + #FM02 + #FS10	0	6	FM	FS	FS		FS	FS	FS	
#ES00 + #SR00 + #FM00 + #FS12	0	6	FS	FS	FS		FS	FS	FS	
#ES00 + #SR00 + #FM16 + #FS00	0	8	FM	FM	FM	FM	FM	FM	FM	FM
#ES00 + #SR00 + #FM14 + #FS02	0	8	FM	FM	FM	FM	FM	FM	FM	FS
#ES00 + #SR00 + #FM12 + #FS04	0	8	FM	FM	FM	FS	FM	FM	FM	FS
#ES00 + #SR00 + #FM10 + #FS06	0	8	FM	FM	FM	FS	FM	FM	FS	FS
#ES00 + #SR00 + #FM08 + #FS08	0	8	FM	FM	FS	FS	FM	FM	FS	FS
#ES00 + #SR00 + #FM06 + #FS10	0	8	FM	FM	FS	FS	FM	FS	FS	FS
#ES00 + #SR00 + #FM04 + #FS12	0	8	FM	FS	FS	FS	FM	FS	FS	FS
#ES00 + #SR00 + #FM02 + #FS14	0	8	FM	FS	FS	FS	FS	FS	FS	FS
#ES00 + #SR00 + #FM00 + #FS16	0	8	FS	FS	FS	FS	FS	FS	FS	FS
#ES08 + #SR00 + #FM04 + #FS00	4	2	E	FM	E		E	FM	E	
#ES08 + #SR00 + #FM02 + #FS02	4	2	E	FM	E		E	FS	E	
#ES08 + #SR00 + #FM00 + #FS04	4	2	E	FS	E		E	FS	E	
#ES08 + #SR00 + #FM08 + #FS00	4	4	E	FM	E	FM	E	FM	E	FM
#ES08 + #SR00 + #FM06 + #FS02	4	4	E	FM	E	FM	E	FM	E	FS
#ES08 + #SR00 + #FM04 + #FS04	4	4	E	FM	E	FS	E	FM	E	FS
#ES08 + #SR00 + #FM02 + #FS06	4	4	E	FM	E	FS	E	FS	E	FS
#ES08 + #SR00 + #FM00 + #FS08	4	4	E	FS	E	FS	E	FS	E	FS
#ES00 + #SR04 + #FM04 + #FS00	2	2	S	FM			S	FM		
#ES00 + #SR04 + #FM02 + #FS02	2	2	S	FM			S	FS		
#ES00 + #SR04 + #FM00 + #FS04	2	2	S	FS			S	FS		
#ES00 + #SR04 + #FM08 + #FS00	2	4	S	FM		FM	S	FM		FM
#ES00 + #SR04 + #FM06 + #FS02	2	4	S	FM		FM	S	FM		FS
#ES00 + #SR04 + #FM04 + #FS04	2	4	S	FM		FS	S	FM		FS
#ES00 + #SR04 + #FM02 + #FS06	2	4	S	FM		FS	S	FS		FS

Table 5-3 ICE2 and ICF Card Matrix

#ES00 + #SR04 + #FM00 + #FS08	2	4	S	FS		FS	S	FS		FS
#ES00 + #SR08 + #FM04 + #FS00	4	2	S	FM	S		S	FM	S	
#ES00 + #SR08 + #FM02 + #FS02	4	2	S	FM	S		S	FS	S	
#ES00 + #SR08 + #FM00 + #FS04	4	2	S	FS	S		S	FS	S	
#ES00 + #SR08 + #FM08 + #FS00	4	4	S	FM	S	FM	S	FM	S	FM
#ES00 + #SR08 + #FM06 + #FS02	4	4	S	FM	S	FM	S	FM	S	FS
#ES00 + #SR08 + #FM04 + #FS04	4	4	S	FM	S	FS	S	FM	S	FS
#ES00 + #SR08 + #FM02 + #FS06	4	4	S	FM	S	FS	S	FS	S	FS
#ES00 + #SR08 + #FM00 + #FS08	4	4	S	FS	S	FS	S	FS	S	FS
#ES08 + #SR04 + #FM00 + #FS00	6	0	E	S	E		E	S	E	
#ES08 + #SR08 + #FM00 + #FS00	8	0	E	S	E	S	E	S	E	S
#ES12 + #SR04 + #FM00 + #FS00	8	0	E	E	E	S	E	E	E	S
#ES04 + #SR04 + #FM04 + #FS00	4	2	E	FM	S		E	FM	S	
#ES04 + #SR04 + #FM02 + #FS02	4	2	E	FM	S		E	FS	S	
#ES04 + #SR04 + #FM00 + #FS04	4	2	E	FS	S		E	FS	S	
#ES04 + #SR04 + #FM08 + #FS00	4	4	E	FM	S	FM	E	FM	S	FM
#ES04 + #SR04 + #FM06 + #FS02	4	4	E	FM	S	FM	E	FM	S	FS
#ES04 + #SR04 + #FM04 + #FS04	4	4	E	FM	S	FS	E	FM	S	FS
#ES04 + #SR04 + #FM02 + #FS06	4	4	E	FM	S	FS	E	FS	S	FS
#ES04 + #SR04 + #FM00 + #FS08	4	4	E	FS	S	FS	E	FS	S	FS

Note:

- E = ESCON (an ICE card), S = SCSI (an ICE card with code loaded for SCSI attach), FM = a fibre card (ICF card) with Multi-mode GBICs installed, FS = a fibre card (ICF card) with Single-mode GBICs installed.
- Both ICE2 and ICF cards have two ports. On ICE2 cards these ports must both be configured as either ESCON or as SCSI. An ICE2 card cannot have a mixed SCSI and ESCON configuration.
- Both clusters must have the same card configuration (balanced). (There cannot be a single card in a cluster.)
- In any even/odd slot pair, the legal combinations are:
    - both empty,
    - slot 0 occupied/slot 1 empty,
    - both occupied with the same card type, or
    - slot 1 has fibre (due to ISP code operation).

If the customer upgrades from fibre to fibre/SCSI or fibre/ ESCON, they have to move the fibre card to slot 1 and install the ICE2 card in slot 0.

Setting I/O Timers

Use the following information to change the SCSI time out values (to 270 seconds) on the supported Open System platforms. These changes will enhance the recovery of processing jobs/data in the event of a check 0 recovery on the SVA/XSA.

You will need a system administrator with root login access. All commands to change the time out values to a minimum value of 270 seconds must be made from the root. If any value already exists which exceeds the 270 second minimum limit do not change these values.

HP-UX

There are two commands that can be used to view and change IO time-out parameters on a per device basis. These commands are "pvchange" and "pvdisplay". See the system manual pages for more information on each of these commands. An example for "pvdisplay" would be:

```
pvdisplay  /dev/dsk/c1t3d0
```

This will show an IO_time value in seconds which may say default. To change this value to 270 seconds use the command:

```
pvchange t 270 /dev/dsk/c1t3d0
```

The device needs to be part of a volume group before "pvdisplay" and "pvchange" will work. Once the devices are changed to the 270 time out value the host system will need to have a REBOOT to download the changes to the drivers.

RS6000 - AIX

The changes can be made via "smit", via a command line entry, or script file using "chdev" to change the time out value and "lsattr" to display the attributes of the devices. The changes can be made upon initial set up to the raw devices or with a system REBOOT to devices already configured to a volume group.

If queuing is not turned on request that queuing be turned on to at lease the minimum levels listed below.

SMIT

Devices

Fixed Disk

Change / Show Characteristics of a Disk

Select the hdisk # to change:

Queue Depth	[16]
Queuing TYPE	[simple]
Use QERR bit	[no]
Device CLEARS its Queue on error	[no]
READ/WRITE time out value	[270]
START UNIT time out value	[270]
REASSIGN time out value	[270]

Note: If the timeout values are already greater than 270 do not change these values. Also if queuing values are set higher than those listed here no queue value change is needed.

If this is a raw disk you do not need to make any further changes. If this disk is already established in a volume group, make the following changes and note that the changes will no be applied until the host system is rebooted.

```
Apply change to DATABASE only        yes
```

Via Command Line or you can build a script to include several command lines:

Raw Disk:

```
Chdev l 'hdisk#' a queue_depth='16' a q_type='simple' a
rw_timeout='270' a start_timeout='270' a reassign_to='270' a
q_err='no'
```

Established volume group (REBOOT required):

```
Chdev l 'hdisk#' a queue_depth='16' a q_type='simple' a
rw_timeout='270' a start_timeout='270' a reassign_to='270' a
q_err='no' P
```

SUN

The changes on the Sun are made by editing the '/etc/system' file. Add or modify the following line within this file. Again the value of 270 is a minimum value. If a larger value already exists use the larger value.

```
set  sd:sd_io_time=270
```

A system REBOOT must occur for this value to take effect. It is suggested that you reboot your system with the following parameters:

```
boot r
```

Also see “Missing-Interrupt Handler (MIH)” on page 15.

Windows NT

To increase the disk I/O time-out value on NT perform the following:

1. Click on Start on the task bar. Select run and enter regedt32 in the popup window. Click OK to invoke registry editor.
2. Select HKEY_LOCAL_MACHINE->SYSTEM->CurrentControlSet->Services->Disk.
3. Under Edit, select Add Value. Enter a Value Name of “TimeOutValue” and select Data Type of REG_DWORD.
4. The DWORD editor window will pop up. Select a Radix of Dec and enter a value of 270. This value is in seconds. The registry entry should be TimeOutValue : REG_DWORD : 0x10e.

This page intentionally left blank

Chapter 6 Monitoring and Controlling an SVA

Two interfaces are available to monitor, control, or alter Shared Virtual Array's operating characteristics:

- The local operator panel at the Shared Virtual Array Controller
- Shared Virtual Array Administrator (SVAA) at a host-attached terminal.

The local operator panel is the hardware module on the Shared Virtual Array Controller that allows the local operator to configure, monitor, and control a subsystem. The local operator panel is documented in the *Shared Virtual Array Operation and Recovery.*

To monitor and control access density, net capacity load, and overall cache effectiveness, SVAA is the most efficient and flexible interface to use. Also, other measurement facilities can be used to measure these statistics (as described in "Measurement Tools" on page 36), but they cannot report Shared Virtual Array-specific critical performance factors, such as NCL.

Shared Virtual Array Administrator (SVAA)

SVAA is the optional host software program that can continuously monitor and report on subsystem performance and capacity load statistics. It can provide detailed monitoring data for a subsystem. The data can then be used to understand the operation of the subsystem and to optimize Shared Virtual Array's performance and capacity utilization.

The SVAA Reporter facility component reports on functional device and cache performance and on space utilization. Several predefined reports and graphs of this data are available. The Reporter facility also provides the user with subsystem data in a flat file or in SMF, which can be manipulated by the user's own report writer and graphics display tools.

SVAA subsystem configuration facilities enable the user to control Shared Virtual Array's functional device configuration, as it appears to the host. In addition, the user can control subsystem storage director and channel path parameters, or perform physical device operations such as forming arrays or draining devices. SVAA also reports on physical and functional configuration of a Shared Virtual Array subsystem.

Besides its system reporting feature, SVAA provides extended operator control facilities and schedules the execution of DDSR (refer to "Deleted Data Space Release" on page 25). SVAA allows the user to control the time and frequency of DDSR's execution to minimize possible interference with normal operations.

The SVAA console operator commands provide four basic functions for controlling a Shared Virtual Array:

1. DISPLAY CHANNEL - displays the status of the Shared Virtual Array channel interface.
2. VARY CHANNEL - enables or disables a Shared Virtual Array channel interface.
3. DISPLAY DEVICE - displays the status of a functional device within an Shared Virtual Array subsystem.
4. VARY DEVICE - enables or disables a functional device and adds or removes write protection for a functional device.

Note: These device-oriented commands operate at the functional device level, not at the physical device level.

Shared Virtual Array Reporting

The SVAA Reporter facility component reports on functional device and cache performance and on space utilization. Several predefined reports and graphs of this data are available. Reporter (the Reporter facility) also provides the user with subsystem data in a flat file or in SMF, which can be manipulated by the user’s own report writer and graphics display tools.

Free Space Collection Load

The Reporter facility reports a value called free space collection load (FSCL). This number is the ratio of the number of bytes that were collected by FSC to the number of bytes actually freed by free space collection. It reflects the amount of work being done by Shared Virtual Array to maintain free space and reclaim array cylinders. It is expressed as a number between 0 and 9 and it is the ratio of data moved to free space collected.

For example, a filled array cylinder holds approximately 10MB of data. If 500MB of data were staged off 150 collection array cylinders, the result would be 1500MB of space being freed. This is a free space collection load of 3. The result of having freed 1500MB in a two-array 100GB subsystem would affect collected free space by approximately 1.5 percent.

As array cylinders are collected, the free space collection load will vary according to the ratio of free space to user data. If the subsystem is very busy, then FSC falls behind and, as a result, will collect array cylinders with a higher percentage of free space. This will allow the collection process to operate efficiently because it does not have to off load much data to collect an array cylinder. When the subsystem is less busy, FSC will catch up and collect array cylinders that do not have as much free space. This will be reflected by a higher free space collection load. When all the array cylinders have been collected, free space collection will stop and the FSCL will go to zero. It is normal to see free space collection load be low (i.e., 2 or 3) during heavy operation, and then get higher (i.e., 7 or

9) as activity subsides until it finally goes to zero when FSC stops or sleeps.

SnapShot

SnapShot is a combination of host software and Shared Virtual Array architecture blended to form a unique and powerful DASD copying process. This section describes how SnapShot operates in the host and in the Shared Virtual Array subsystem.

How SnapShot Works

SnapShot operations in the Shared Virtual Array subsystem are invoked from the host and executed in the subsystem. SnapShot can "snap" entire DASD volumes or individual data sets.

Operation in the Host

SnapShot host software:

- Validates SnapShot requests
- Identifies the DASD volumes involved
- Sends commands to the subsystem to control the snapping
- Handles serialization of the resources involved
- Handles security measures for the resources involved.

The related host data (VTOC, VTOC index, and catalog structures) must also be updated in order for the host to recognize copied data (either a volume or a data set). In some cases, SVAA handles these updates automatically.

Operation in the Shared Virtual Array Subsystem

The Shared Virtual Array subsystem:

- Receives and validates the SnapShot commands that the host software sends
- Initiates, from the system’s point of view, a point-in-time copy of the identified source extent to the target extent
- Updates the mapping tables to point to the copy
- Stores the updated mapping tables, such that the SnapShot is recoverable
- Responds with completed information to the host SnapShot software.

The host can now access the source data, either through the functional address of the original source or through the functional address of the new snap (the target). The host can view these copies as separate extents.

SnapShot Scenarios

Two basic scenarios, Hot Snap and Quick Snap, provide maximum 24x7 coverage.

Hot Snap

Hot Snap is used to create recovery copies without impacting transaction response time. The likelihood of needing to recover the application is small, but recovery may take longer than with Quick Snap.

Use Hot Snap when recovery is necessary but you cannot tolerate down time to take a copy.

Quick Snap

Quick Snap is used when the applications may need to be recovered. Transaction response is somewhat impacted while the snap is being made, but recovery is faster than with Hot Snap.

Use Quick Snap for disaster backup. Quick Snap requires less log-forward processing, but makes use of Shared Virtual Array's subsystem-level copy to dramatically reduce the transaction down time.

The snap process is further enhanced by making the migration to tape at any time after the snap is made, while allowing this copy to remain in a position to be used for a software failure.

If enough functional volumes are available, you can maintain at least one Hot Snap and one Quick Snap constantly. But, if additional functional volumes are not available once a Quick Snap has been migrated to tape for disaster recovery, the same volume can be reused for an incremental Hot Snap.

Refer to *Shared Virtual Array Using SnapShot*, for more detailed information about SnapShot's capabilities.

For more detailed information about SVAA's capabilities, refer to *SVAA Configuration and Administration*.

Chapter 7 Exception Conditions

As with all data storage devices, Shared Virtual Array experiences exception conditions that must be addressed. This chapter outlines responses to the more common exceptions.

Fencing

Shared Virtual Array has removed the responsibility for fence management from the operator by automatically isolating failed resources and clearing an imposed fence when the resource has been replaced. The StorageTek CSE performs the only human intervention required in the fence processing sequence. Therefore, the Shared Virtual Array Controller rejects the ICKDSF CONTROL CLEARFENCE command.

A Shared Virtual Array does not fence functional devices, which are the equivalent of conventional devices behind a 3990. Since data is stored to, and accessed from, multiple physical drive modules within an array, there are multiple paths to a functional volume from the Controller. Should an error occur that restricts access to a physical device within the array, the drive reconstruction recreates the data stored on that device and places it on a spare drive.

Data Assurance Check Mode

In certain conditions, a Shared Virtual Array subsystem may enter a data assurance check mode. In this state, data cannot be assured to be correct and commands that access data are not accepted.

The conditions that cause the subsystem to enter a data assurance check mode are:

- The battery backup units that support nonvolatile storage (NVS) have discharged after a power failure with modified records in NVS
- The mapping tables cannot be completely recovered during a Controller initialization.

In the data assurance check mode, sense bytes 22 and 23, which contain the fault symptom code, are 9203 or E203. The console log or PM2 reports for this time period may provide more information about the condition.

You can reset this condition and exit from this state by resetting the data assurance check mode at the local operator panel (refer to the *Shared Virtual Array Operation and Recovery*) After resetting the subsystem, normal operation can proceed; however, you must take steps to determine what, if any, data has been compromised.

Pinned Data

Traditionally, pinned data is data in cache that cannot be written to a device, and thus must remain in cache until it is manually erased. Pinned data is usually the result of an un-correctable media failure where the record is stored.

A Shared Virtual Array subsystem does not experience pinned data because of dynamic mapping. All write operations result in a functional track being copied from cache to a disk array. The functional track is always written to a new location, never to the position from which it was read.

Exception Reporting

Shared Virtual Array exception conditions are reported in several ways and can be collected and formatted by using the utilities described below.

EREP Exception Reporting

The use of Environmental Record Editing and Printing (EREP) for exception reporting is of minimal value with Shared Virtual Array. Most problems are resolved within the subsystem and are not visible to the host. Correctable errors are resolved using the redundancy data to reconstruct damaged functional tracks.

PM2 Exception Reporting

Performance Monitoring/Predictive Maintenance (PM2) is a maintenance package that provides hardware error reporting to aid support personnel in problem diagnosis. Reports are provided for history maintenance, daily reporting, monthly statistics, and miscellaneous information. PM2 correlates service information messages (SIMs) and StorageTek's ServiceTek (StorageTek's machine-initiated maintenance facility) from LOGREC, for customer-initiated maintenance.

POST Reporting

Program for Online System Testing (POST) provides a valuable diagnostic tool for the CSE and support personnel in both online and stand-alone environments. It can be used to verify subsystem operation and to report all of the errors contained in the sense data and SIM information.

Using SIMs

Service information messages (SIM) contain sense data that describe certain error conditions encountered by the subsystem. All SIMs are recorded in the error recording data set (ERDS). Based on the SIM sense data, a SIM alert is sent to the operator console indicating that an error condition has been encountered and recorded in the ERDS.

A SIM alert includes:

- Machine type and model
- Plant of manufacture and machine serial number
- Failing part of the subsystem
- Severity of the failure
- Effect of the repair
- REFCODE.

When requesting service, the customer reports the machine type and machine serial number.

The SIM sense data recorded in the ERDS contains more specific information about the error than does the SIM alert message. The Environmental Record Editing and Printing (EREP) program, PM2, or a similar program can be used to produce a report describing the details of the error.

SIM alerts are issued to the operator console based on the severity of the error. A Shared Virtual Array subsystem generates SIM alerts for the following types of error conditions:

- Controller failures
- Cache failures
- Device failures.

Shared Virtual Array does not generate SIM alerts for media failures, because the disk arrays allow the dynamic recovery of any data stored on failed media. For more information about track recovery and drive reconstruction, refer to the *Shared Virtual Array Introduction*, or the *Shared Virtual Array Reference*.

SIM sense data is also generated for conditions that are unique to Shared Virtual Array. The unique conditions detected by Shared Virtual Array are:

- Low number of spare drives
- Low NVS battery voltage
- Physical device failure: reconstruction process initiated
- Physical device failure: reconstruction process completed
- Net capacity load (NCL) threshold exceeded
- Drain operation complete
- Battery back to full charge
- Out of back-end space.

A SIM alert for these unique conditions may be generated based on the severity of the condition. For information about Shared Virtual Array-specific SIMs and SIM alerts, refer to the *Shared Virtual Array Reference* or *IXFP Messages and Codes*.

Establishing SIM Handling Procedures

In an MVS environment, a SIM alert remains on the operator console until the operator or storage administrator responds to the message. When a SIM alert is detected, the “A3” records generated by the SIM should be collected from all attached systems’ ERDS. To do this, run EREP with the input ERDS merged or perform a similar process with PM2.

For MVS, the process to collect the “A3” records can be automated in several ways. The appropriate JCL and control statements necessary to generate an EREP report or PM2 report can be stored in a procedure library and invoked from a CLIST or REXX EXEC. Most console message automation utilities can be used to initiate an EREP report or PM2 report following a SIM alert.

Because SIM alerts in the VM environment do not require a response, the operator may miss the message. There are several methods by which the operator can be made aware of SIM occurrences.

1. In VM/ESA environments, a programmable operator facility exists that allows interception and handling of messages. The programmable operator facility can be used to intercept a SIM alert, initiate the execution of an EREP or PM2 report, and route the SIM alert directly to the operator or storage administrator.
2. As in the MVS environment, the A3 records from the ERDS must be collected from all attached subsystems.

For further information on EREP, VM/ESA, or IBM 3990 disk controllers, refer to the *IBM Publications KWIC Index* for the appropriate manual. For information about PM2, refer to the *PM2 Installation/User’s Manual.*

ServiceTek

If a subsystem is supported by ServiceTek (StorageTek’s machine-initiated maintenance facility) the subsystem produces two types of ServiceTek alerts that initiate automatic connection to the Customer Service Remote Center (CSRC).

Unit Failure

This type of alert occurs when there has been a Shared Virtual Array subsystem failure, when a Shared Virtual Array subsystem is about to fail, or when the Shared Virtual Array subsystem is running with degraded performance.

Trace/Event Log Down-Load

This type of alert occurs periodically and involves the down-loading of diagnostic data. Additionally, when Shared Virtual Array subsystem performance and statistical thresholds have been reached, a call is initiated to CSRC, and subsystem data is down-loaded.

The ServiceTek Facility can be enabled or disabled through a local or extended Shared Virtual Array subsystem local operator panel. If ServiceTek is disabled, a record is entered in the Shared Virtual Array event log indicating that ServiceTek is disabled. In this event, exception conditions are still logged to the Shared Virtual Array event log and are accessible through the local or extended local operator panel.

Disk Array Recovery

The Shared Virtual Array Control Unit can automatically recover from a simultaneous failure of two drive modules in the same array. Redundancy data is read from the remaining drives in the array and the data is reconstructed on a spare drive. However, three simultaneous device failures within a single array is considered an array failure because insufficient redundancy data remains to reconstruct the failing devices.

While an array failure is extremely unlikely, data loss in such a case can be significant. A detailed description follows of what should be done in case of an array failure.

In the event of an array failure, data residing on the other disk arrays within the subsystem are still accessible. However, to reduce further data loss, all activity to the subsystem with the failed array should be halted and recovery procedures should be initiated for all functional volumes in the subsystem. As with all DASD storage subsystems, you should have adequate backup and recovery procedures defined and implemented for the data center.

Utilities such as DFSMSdss can save the remaining readable data on the failed array, so that damaged data sets can be identified. In doing so, all channels to the subsystem should be driven at or near their maximum transfer speed. This process is expedited when:

- The dump utility options specify processing by cylinder
- The dump utility options are set to tolerate I/O errors
- Host compression is not specified for the utility
- The target devices are tape devices
- Multiple functional volumes are dumped simultaneously.

Recovery Procedure

The following procedure provides a guideline for recovering data from a Shared Virtual Array subsystem in the event of an array failure. Tailor this procedure to your environment.

Upon discovering an array failure:

1. Immediately halt all update activity to the subsystem.
2. Initiate DDSR processing for all functional volumes in the subsystem. This will delete un-allocated data space within the subsystem.

3. Begin a functional volume dump operation for all of the volumes on the subsystem. This step will capture the remaining data specifying DATA ONLY for the dump operation. Use the CYLINDER processing option for the target devices to maximize performance.
4. Drain the damaged array (refer to the *Shared Virtual Array Operation and Recovery*).
5. Using the reports generated in the dump process, identify the data sets that could not be dumped because of I/O errors.
6. Un-catalog and delete the unreadable data sets.
7. When the array is drained, initiate DDSR processing for all of the functional volumes. This deletes the un-allocated space just created by the un-catalog/delete processing.
8. Replace the failed devices with new ones.
9. Form a new production array from the Spares partition.
10. Recover and/or restore the data sets identified as unreadable from the most current backup available.
11. Perform forward recovery for restored data sets, as needed.
12. Resume normal processing.

The time required to perform the entire procedure will depend upon the net capacity load and the size of the subsystem. However, use the following formulas to estimate the “expected” time required to perform the dump operation. (The two examples provided here are extreme cases, but they may be used to estimate the time for a specific site.)

Recovery Time Estimate

The amount of data contained on a Shared Virtual Array subsystem (its functional load) can be determined by multiplying the physical capacity by the net capacity load and dividing that value by the compression-compaction index:

$$\text{functional load} = \frac{\text{(physical capacity) X (\% net capacity load)}}{\text{(compression-compaction index)}}$$

Using the functional load, the amount of time required to dump that data can be estimated by multiplying the channel speed by the number of channels and dividing that value into the functional load. However, this formula assumes that tape devices are not a constraining factor.

$$\text{dump time (sec)} = \frac{\text{(functional load)}}{\text{(channel speed X number of channels)}}$$

As indicated, the result is in number of seconds required. The number of minutes required may be obtained by dividing that value by 60.

Appendix A ESCON IOCP EXAMPLES, 1024 Devices

Use the following examples to develop an ESCON IOCP for your specific environments. Each example shows a configuration diagram followed by the associated IOCP statements. Listed below are examples, along with a brief description.

Example 1. SVA with ESCON attachment from multi hosts through a single ESCD.

Example 2. SVA in a multi-host environment with ESCON attachment through two ESCDs.

Example 3. **Eight-Path Example** of SVA in a multi-LPAR environment using EMIF attached through two ESCDs. Includes VSE definition.

Example 4. SVA in a two-host environment, both with ESCON channels. The first host is MVS with 511 3390 devices and one 3380 device; the second host is VM with 256 3390 devices and 256 3380 devices.

Example 5. SVA in a single-host MVS environment, with 512 3390 devices.

Example 6. SVA in a single-host MVS environment. ESCON channels are direct-connected, with 512 3390 devices.

Example 1, 1024 Maximum Devices

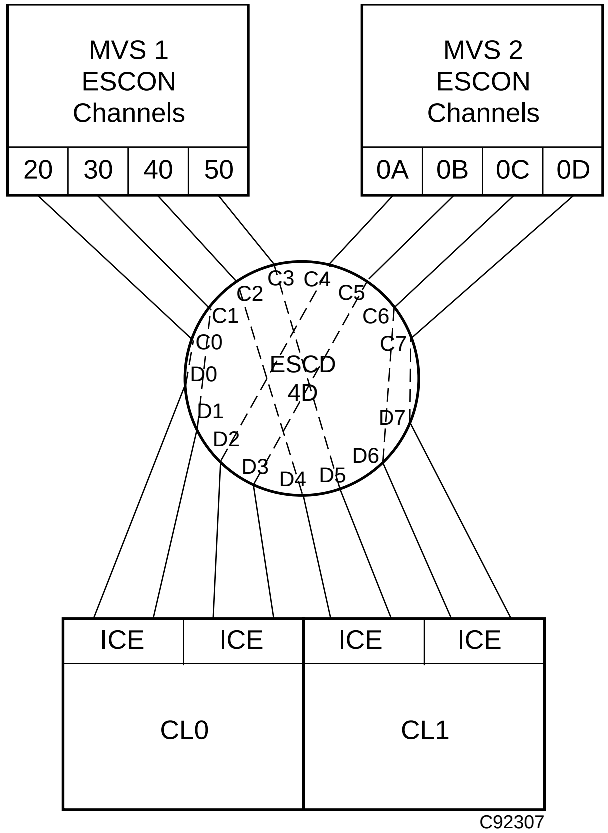


Figure A-1 Example 1. Multi-host ESCON Channel Attachment through an ESCON Director

Table A-1 Multi-Host ESCON Channel Attachment through an ESCON Director

Base	00
Range	1024
BFID	00

IOCP for MVS 1 - ESCON Channels

```
CHPID    PATH=((20),(30),(40),(50)),TYPE=CNC,SWITCH=4D
CNTLUNIT CUNUMBR=001,          CNTLUNIT CUNUMBR=002,
      UNIT=3990,                 UNIT=3990,
      UNITADD=((00,256)),        UNITADD=((00,256)),
      PATH=(20,30),              PATH=(40,50),
      LINK=(D0,D1),              LINK=(D4,D5)
      CUADD=0                    CUADD=0
CNTLUNIT CUNUMBR=003,          CNTLUNIT CUNUMBR=004,
      UNIT=3990,                 UNIT=3990,
      UNITADD=((00,256)),        UNITADD=((00,256)),
      PATH=(20,30),              PATH=(40,50),
      LINK=(D0,D1),              LINK=(D4,D5),
      CUADD=1                    CUADD=1
CNTLUNIT CUNUMBR=005,          CNTLUNIT CUNUMBR=006,
      UNIT=3990,                 UNIT=3990,
      UNITADD=((00,256)),        UNITADD=((00,256)),
      PATH=(20,30),              PATH=(40,50),
      LINK=(D0,D1),              LINK=(D4,D5),
      CUADD=2                    CUADD=2
CNTLUNIT CUNUMBR=007,          CNTLUNIT CUNUMBR=008,
      UNIT=3990,                 UNIT=3990,
      UNITADD=((00,256)),        UNITADD=((00,256)),
      PATH=(20,30),              PATH=(40,50),
      LINK=(D0,D1),              LINK=(D4,D5),
      CUADD=3                    CUADD=3
IODEVICE UNIT=3390,ADDRESS=(100,256),CUNUMBR=(001,002),
      UNITADD=00,STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3390,ADDRESS=(200,256),CUNUMBR=(003,004),
      UNITADD=00,STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3390,ADDRESS=(300,256),CUNUMBR=(005,006),
      UNITADD=00,STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3390,ADDRESS=(400,224),CUNUMBR=(007,008),
      UNITADD=00,STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3380,ADDRESS=(480,32),CUNUMBR=(007,008),
      UNITADD=80,STADET=Y,FEATURE=SHARED
```

Note: UNITADD=80 is coded in the last IODEVICE macro to define a group of 3380 devices. The UNITADD parameter must be two digits within the hexadecimal range of 00-FF for each set of 256 addresses.

IOCP for MVS 2 - ESCON Channels

```
CHPID    PATH=((0A),(0B),(0C),(0D)),TYPE=CNC,SWITCH=4D
CNTLUNIT CUNUMBR=001,          CNTLUNIT CUNUMBR=002,
      UNIT=3990,                     UNIT=3990,
      UNITADD=((00,256)),            UNITADD=((00,256)),
      PATH=(0A,0B),                  PATH=(0C,0D),
      LINK=(D2,D3),                  LINK=(D6,D7),
      CUADD=0                        CUADD=0
CNTLUNIT CUNUMBR=003,          CNTLUNIT CUNUMBR=004,
      UNIT=3990,                     UNIT=3990,
      UNITADD=((00,256)),            UNITADD=((00,256)),
      PATH=(0A,0B),                  PATH=(0C,0D),
      LINK=(D2,D3),                  LINK=(D6,D7)
      CUADD=1                        CUADD=1
CNTLUNIT CUNUMBR=005,          CNTLUNIT CUNUMBR=006,
      UNIT=3990,                     UNIT=3990,
      UNITADD=((00,256)),            UNITADD=((00,256)),
      PATH=(0A,0B),                  PATH=(0C,0D),
      LINK=(D2,D3),                  LINK=(D6,D7),
      CUADD=2                        CUADD=2
CNTLUNIT CUNUMBR=007,          CNTLUNIT CUNUMBR=008,
      UNIT=3990,                     UNIT=3990,
      UNITADD=((00,256)),            UNITADD=((00,256)),
      PATH=(0A,0B),                  PATH=(0C,0D),
      LINK=(D2,D3),                  LINK=(D6,D7),
      CUADD=3                        CUADD=3
IODEVICE UNIT=3390,ADDRESS=(100,256),CUNUMBR=(001,002),
      UNITADD=00, STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3390,ADDRESS=(200,256),CUNUMBR=(003,004),
      UNITADD=00, STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3390,ADDRESS=(300,256),CUNUMBR=(005,006),
      UNITADD=00, STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3390,ADDRESS=(400,224),CUNUMBR=(007,008),
      UNITADD=00, STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3390,ADDRESS=(480,32),CUNUMBR=(007,008),
      UNITADD=80, STADET=Y,FEATURE=SHARED
```

Example 2, 1024 Maximum Devices

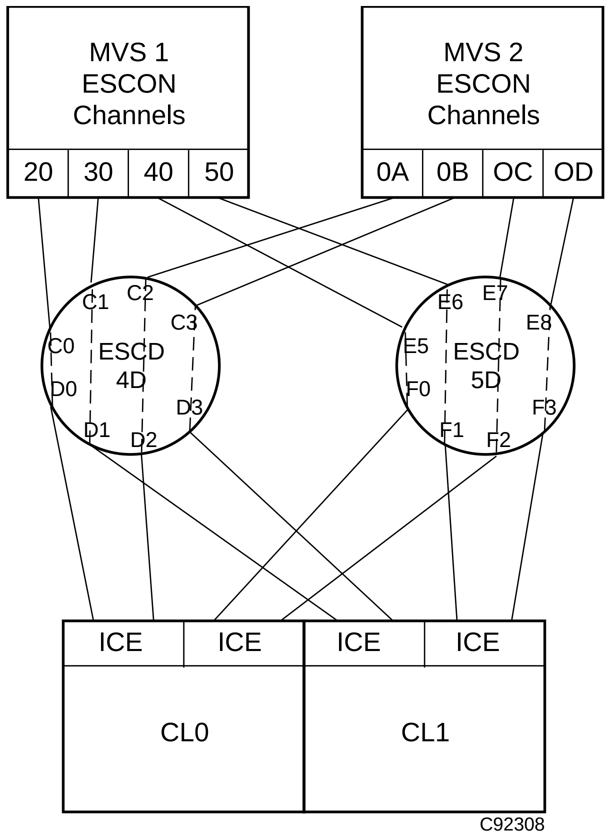


Figure A-2 Example 2. Multi-host ESCON Channel Attachment through Two ESCON Directors

Table A-2 Multi-Host ESCON Channel Attachment through Two ESCON Directors

Base	00
Range	1024
BFID	00

IOCP for MVS 1 - Using Two ESCON Directors

```
CHPID    PATH=((20),(30)),TYPE=CNC,SWITCH=4D
CHPID    PATH=((40),(50)),TYPE=CNC,SWITCH=5D
CNTLUNIT CUNUMBR=001,   CNTLUNIT CUNUMBR=002,
     UNIT=3990,              UNIT=3990,
     UNITADD=((00,256)),     UNITADD=((00,256)),
     PATH=(20,40),           PATH=(30,50),
     LINK=(D0,F0),           LINK=(D1,F1),
     CUADD=0                 CUADD=0
CNTLUNIT CUNUMBR=003,   CNTLUNIT CUNUMBR=004,
     UNIT=3990,              UNIT=3990,
     UNITADD=((00,256)),     UNITADD=((00,256)),
     PATH=(20,40),           PATH=(30,50),
     LINK=(D0,F0),           LINK=(D1,F1),
     CUADD=1                 CUADD=1
CNTLUNIT CUNUMBR=005,   CNTLUNIT CUNUMBR=006,
     UNIT=3990,              UNIT=3990,
     UNITADD=((00,256)),     UNITADD=((00,256)),
     PATH=(20,40),           PATH=(30,50),
     LINK=(D0,F0),           LINK=(D1,F1),
     CUADD=2                 CUADD=2
CNTLUNIT CUNUMBR=007,   CNTLUNIT CUNUMBR=008,
     UNIT=3990,              UNIT=3990,
     UNITADD=((00,256)),     UNITADD=((00,256)),
     PATH=(20,40),           PATH=(30,50),
     LINK=(D0,F0)            LINK=(D1,F1)
     CUADD=3                 CUADD=3
IODEVICE UNIT=3390,ADDRESS=(100,256),CUNUMBR=(001,002),
     UNITADD=00,STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3390,ADDRESS=(200,256),CUNUMBR=(003,004),
     UNITADD=00,STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3390,ADDRESS=(300,256),CUNUMBR=(005,006),
     UNITADD=00,STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3390,ADDRESS=(400,256),CUNUMBR=(007,008),
     UNITADD=00,STADET=Y,FEATURE=SHARED
```

IOCP for MVS 2 - Using Two ESCON Directors

```
CHPID    PATH=((0A),(0B)),TYPE=CNC,SWITCH=4D
CHPID    PATH=((0C),(0D)),TYPE=CNC,SWITCH=5D
CNTLUNIT CUNUMBR=001,     CNTLUNIT CUNUMBR=002,
      UNIT=3990,                 UNIT=3990,
      UNITADD=((00,256)),        UNITADD=((00,256)),
      PATH=(0A,0C),              PATH=(0B,0D),
      LINK=(D2,F2),              LINK=(D3,F3),
      CUADD=0                    CUADD=0
CNTLUNIT CUNUMBR=003,   CNTLUNIT CUNUMBR=004,
      UNIT=3990,                 UNIT=3990,
      UNITADD=((00,256)),        UNITADD=((00,256)),
      PATH=(0A,0C),              PATH=(0B,0D),
      LINK=(D2,F2),              LINK=(D3,F3),
      CUADD=1                    CUADD=1
CNTLUNIT CUNUMBR=005,    CNTLUNIT CUNUMBR=006,
      UNIT=3990,                 UNIT=3990,
      UNITADD=((0A,0C)),         UNITADD=((00,256)),
      PATH=(2A,3A),              PATH=(0B,0D),
      LINK=(D2,F2),              LINK=(D3,F3),
      CUADD=2                    CUADD=2
CNTLUNIT CUNUMBR=007,   CNTLUNIT CUNUMBR=008,
      UNIT=3990,                 UNIT=3990,
      UNITADD=((00,256)),        UNITADD=((00,256)),
      PATH=(0A,0C),              PATH=(0B,0D),
      LINK=(D2,F2),              LINK=(D3,F3),
      CUADD=3                    CUADD=3
IODEVICE UNIT=3390,ADDRESS=(100,256),CUNUMBR=(001,002),
      UNITADD=00, STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3390,ADDRESS=(200,256),CUNUMBR=(003,004),
      UNITADD=00, STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3390,ADDRESS=(300,256),CUNUMBR=(005,006),
      UNITADD=00, STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3390,ADDRESS=(400,256),CUNUMBR=(007,008),
      UNITADD=00, STADET=Y,FEATURE=SHARED
```

Example 3, 1024 Maximum Devices

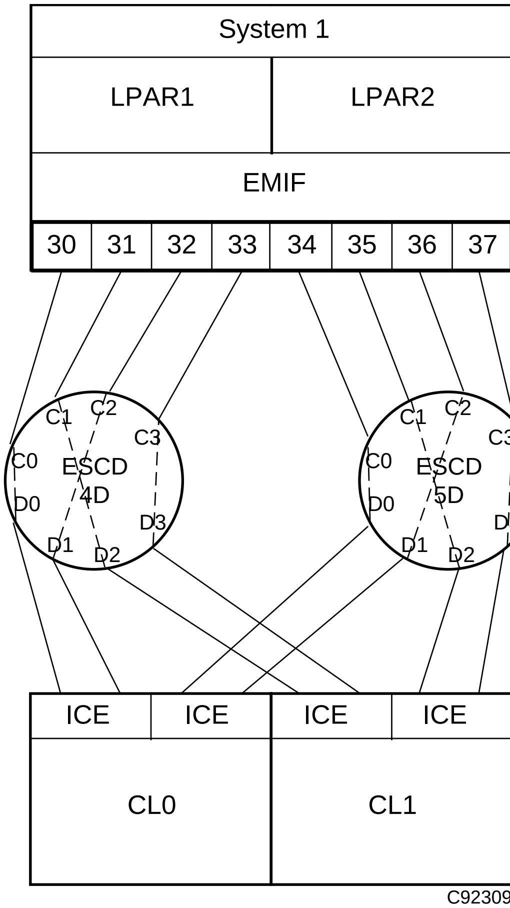


Figure A-3 Example 3. Multi-LPAR ESCON Channel Attachment through Two ESCON Directors

Table A-3 Multi-LPAR ESCON Channel Attachment through Two ESCON Directors

Base	00
Range	1024
BFID	00

VSE/ESA Gen for Example 3

The IOCP for Example 3 in a VSE environment is similar to that of the MVS IOCP. The following GEN statements reflect those necessary for a VSE/ESA system.

ADD 100:1FF,ECKD

IOCP for LPAR1 and LPAR2 with EMIF

```
RESOURCE PART=((LPAR1,1),(LPAR2,2))
CHPID    PATH=((30,31,32,33)),TYPE=CNC,PART=((LPAR1,LPAR2)),
      SHARED,SWITCH=4D
CHPID    PATH=((34,35,36,37)),TYPE=CNC,PART=((LPAR1,LPAR2)),
      SHARED,SWITCH=5D
CNTLUNIT CUNUMBR=001,           CNTLUNIT CUNUMBR=002,
      UNIT=3990,                      UNIT=3990,
      UNITADD=((00,256)),             UNITADD=((00,256)),
      PATH=(30,32,34,36),             PATH=(31,33,35,37),
      LINK=(D0,D1,D0,D1),             LINK=(D2,D3,D2,D3),
      CUADD=0                         CUADD=0
CNTLUNIT CUNUMBR=003,           CNTLUNIT CUNUMBR=004,
      UNIT=3990,                      UNIT=3990,
      UNITADD=((00,256)),             UNITADD=((00,256)),
      PATH=(30,32,34,36),             PATH=(31,33,35,37),
      LINK=(D0,D1,D0,D1),             LINK=(D2,D3,D2,D3),
      CUADD=1                         CUADD=1
CNTLUNIT CUNUMBR=005,           CNTLUNIT CUNUMBR=006,
      UNIT=3990,                      UNIT=3990,
      UNITADD=((00,256)),             UNITADD=((00,256)),
      PATH=(30,32,34,36),             PATH=(31,33,35,37),
      LINK=(D0,D1,D0,D1),             LINK=(D2,D3,D2,D3),
      CUADD=2                         CUADD=2
CNTLUNIT CUNUMBR=007,           CNTLUNIT CUNUMBR=008,
      UNIT=3990,                      UNIT=3990,
      UNITADD=((00,256)),             UNITADD=((00,256)),
      PATH=(30,32,34,36),             PATH=(31,33,35,37),
      LINK=(D0,D1,D0,D1)              LINK=(D2,D3,D2,D3),
      CUADD=3                         CUADD=3
IODEVICE UNIT=3390,ADDRESS=(100,256),CUNUMBR=(001,002),
      UNITADD=00,STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3390,ADDRESS=(200,256),CUNUMBR=(003,004),
      UNITADD=00,STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3390,ADDRESS=(300,256),CUNUMBR=(005,006),
      UNITADD=00,STADET=Y,FEATURE=SHARED
IODEVICE UNIT=3390,ADDRESS=(400,256),CUNUMBR=(007,008),
      UNITADD=00,STADET=Y,FEATURE=SHARED
```

Example 4, 1024 Maximum Devices

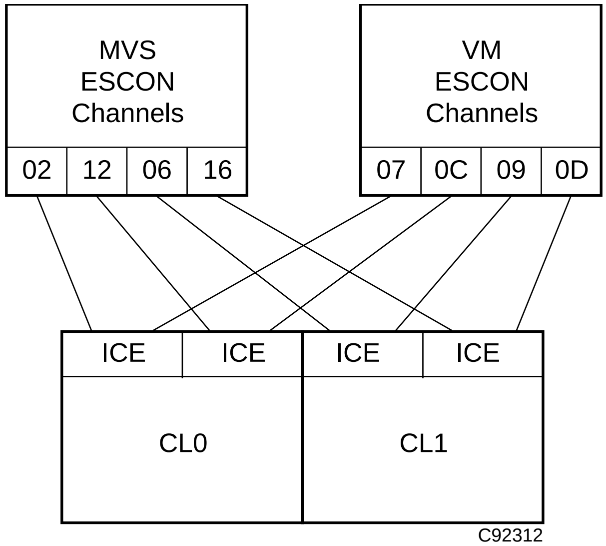


Figure A-4 Example 4. SVA in a Two-host (MVS/VM) Environment, both with ESCON Channels

Table A-4 MVS/VM Two-Host Environment, both with ESCON Channels

Base	00
Range	1024
BFID	00

IOCP in MVS, Two-host Environment, ESCON Channels, Non-shared Devices

The first host in this environment is MVS, with 511 3390 devices and one 3380 device. The second host in this environment is VM, with 256 3390 devices and 256 3380 devices. Devices are not shared.

```
CHPID PATH=(02,06,12,16),TYPE=CNC
CNTLUNIT CUNUMBR=440,PATH=(02,12),CUADD=0,UNIT=3990,
        UNITADD=((00,256))
CNTLUNIT CUNUMBR=450,PATH=(06,16),CUADD=0,UNIT=3990,
        UNITADD=((00,256))
IODEVICE ADDRESS=((B00,255),CUNUMBR=(440,450),UNIT=3390,
        UNITADD=00,STADET=Y
IODEVICE ADDRESS=(BFF,1),CUNUMBER=(440,450),UNIT=3380,
        UNITADD=FF,STADET=Y
 CNTLUNIT CUNUMBR=441,PATH=(02,12),CUADD=1,UNIT=3990,
        UNITADD=((00,256))
CNTLUNIT CUNUMBR=451,PATH=(06,16),CUADD=1,UNIT=3990,
        UNITADD=((00,256))
IODEVICE ADDRESS=(C00,256),CUNUMBR=(441,451),UNIT=3390,
        UNITADD=00,STADET=Y
```

IOCP in VM

```
CHPID PATH=(07,09,0C,0D),TYPE=CNC
CNTLUNIT CUNUMBR=480,PATH=(07,0C),CUADD=2,UNIT=3990,
        UNITADD=((00,256))
CNTLUNIT CUNUMBR=490,PATH=(09,0D),CUADD=2,UNIT=3990,
        UNITADD=(00,256))
IODEVICE ADDRESS=((D00,256),CUNUMBR=(480,490),UNIT=3390,
        UNITADD=00,STADET=Y
CNTLUNIT CUNUMBR=481,PATH=(07,0C),CUADD=3,UNIT=3990,
        UNITADD=((00,256)),SHARED=Y
CNTLUNIT CUNUMBR=491,PATH=(09,0D),CUADD=3,UNIT=3990,
        UNITADD=((00,256)),SHARED=Y
IODEVICE ADDRESS=(E00,256),CUNUMBR=(481,491),UNIT=3380,
        UNITADD=00,STADET=Y
```

Note: VM does not accept the FEATURE= option unless 'IGNORE' is specified when the IOCP program is run. (The VM default is 'NOIGNORE').

Note: If the devices are shared between hosts, then FEAATURE=SHARED must be coded on the IODEVICE statement.

Example 5, 1024 Maximum Devices

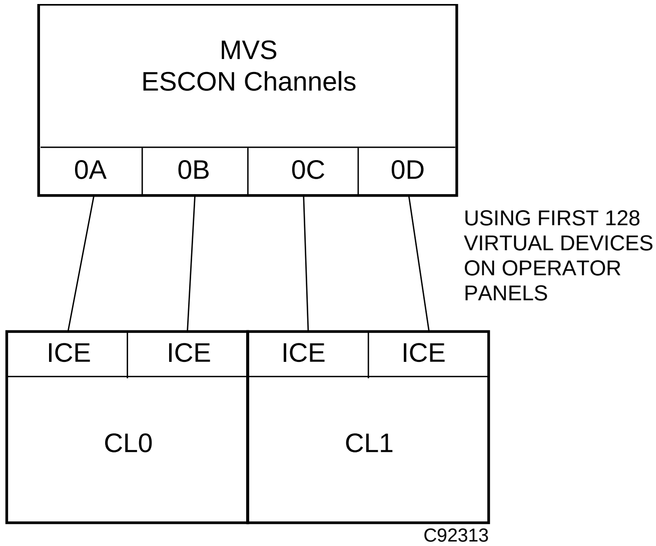


Figure A-5 Example 5. SVA in a Single-host MVS Environment, 512 3390 Devices

Table A-5 SVA in a Single-Host MVS Environment, 512 3390 Devices

Base	00
Range	1024
BFID	00

IOCP in a Single Host MVS Environment (512 3390 DEVICES)

Device address range of 300 through 4FF. ESCON channels are direct-connected.

```
CHPID     PATH=((0A),(0B),(0C),(0D)),TYPE=CNC
CNTLUNIT CUNUMBR=001,   CNTLUNIT CUNUMBR=002,
      UNIT=3990,              UNIT=3990,
      UNITADD=((00,256)),     UNITADD=((00,256)),
      PATH=(0A,0B),           PATH=(0D,0D),
      CUADD=0                 CUADD=0
CNTLUNIT CUNUMBR=003,   CNTLUNIT CUNUMBR=004,
      UNIT=3990,              UNIT=3990,
      UNITADD=((00,256)),     UNITADD=((00,256)),
      PATH=(0A,0B),           PATH=(0D,0D),
      CUADD=1                 CUADD=1
IODEVICE UNIT=3390
      ADDRESS=(300,256),
      UNITADD=00,
      STADET=Y,
      CUNUMBR=(001,002)
IODEVICE UNIT=3390
      ADDRESS=(400,256),
      UNITADD=00,
      STADET=Y,
      CUNUMBR=(003,004)
```

Note: This utilizes the first 512 virtual devices on the local operator panels. Consecutive address ranges of 256 (00-FF, the last two hexadecimal digits) is highly recommended.

Example 6, 1024 Maximum Devices

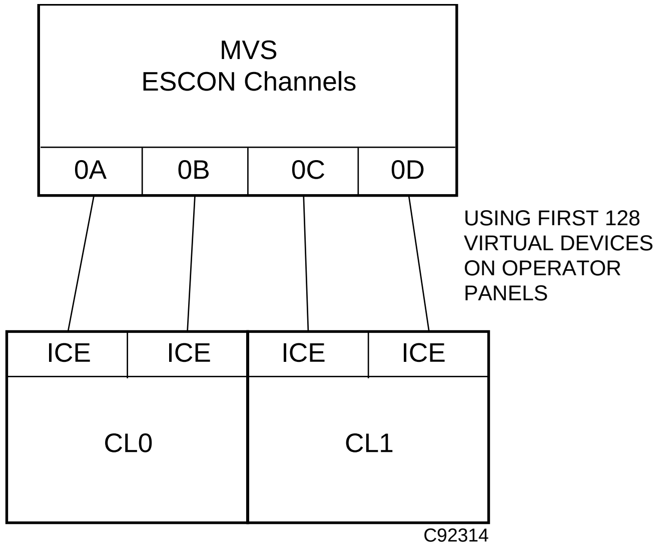


Figure A-6 Example 6. SVA in a Single-host MVS Environment, 512 3390 Devices

Table A-6 SVA in a Single Host MVS Environment

Base	00
Range	1024
BFID	00

SVA in a Single Host MVS Environment

Device address ranges of 100-1FF and 300-3FF are assigned to this subsystem. ESCON channels are direct-connected.

```
CHPID    PATH=(0A),(0B),(0C,)(0D),TYPE=CNC
CNTLUNIT CUNUMBR=001,          CNTLUNIT CUNUMBR=002,
      UNIT=3990,                    UNIT=3990,
      UNITADD=((00,256)),           UNITADD=((00,256)),
      PATH=(0A,0B),                 PATH=(0C,0D),
      CUADD=0                       CUADD=0
CNTLUNIT CUNUMBR=003,          CNTLUNIT CUNUMBR=004,
      UNIT=3990,                    UNIT=3990,
      UNITADD=((00,256)),           UNITADD=((00,256)),
      PATH=(0A,0B),                 PATH=(0C,0D),
      CUADD=1                       CUADD=1
IODEVICE UNIT=3390,ADDRESS=(100,256),UNITADD=00,STADET=Y,
      CUNUMBR=(001,002)
IODEVICE UNIT=3390,ADDRESS=(300,256),UNITADD=00,STADET=Y,
      CUNUMBR=(003,004)
```

Note: This utilizes the first 512 virtual devices on the local operator panels. Consecutive address ranges of 256 (00-FF, the last two hexadecimal digits) is highly recommended.

This page intentionally left blank

Appendix B ESCON HCD EXAMPLES

The following example shows the channel path definition for CHPIDs 30-33:

```
  Add  Modify  View  Filter  Backup  Query  Help
_                          Add Channel Path
S      Specify or revise the following values.
P      Processor ID . . . : SYSTEM1      Production System 1
C      Configuration mode : LPAR
        Channel path ID. . . . . . . . . 30 +
/       Number of CHPIDs . . . . . . . 4
_      Channel path type. . . . . . . CNC +
_      Operation mode . . . . . . . . SHR +
_     Description. . . . . . . . . . . .Shared Virtual Array with EMIF
_
_      Specify the following values only if connected to a switch:
_
_      Dynamic switch ID. . . . .  4D + (00 - FF)
_      Entry switch ID  . . . . . . . 4D +
_      Entry port . . . . . . . . . .  C0 +
```

The following example shows the channel path definition for CHPIDs 34-37:

```
  Add  Modify  View  Filter  Backup  Query  Help
_                          Add Channel Path
S      Specify or revise the following values.
P      Processor ID . . . : SYSTEM1      Production System 1
C      Configuration mode : LPAR
        Channel path ID. . . . . . . . . 34 +
/       Number of CHPIDs . . . . . . . 4
_      Channel path type. . . . . . . CNC +
_      Operation mode . . . . . . . . SHR +
_     Description. . . . . . . . . . . .Shared Virtual Array with EMIF
_
_      Specify the following values only if connected to a switch:
_
_      Dynamic switch ID. . . . . . . 5D + (00 - FF)
_      Entry switch ID  . . . . . . . 5D +
_      Entry port . . . . . . . . . . C0 +
```

The following example defines CHPIDS and ESCON Director Entry Ports:

```
   Add  Modify  View  Filter  Backup  Query  Help
 _                    Update Switch Connections
 S      Specify or revise the following values.
 P      Processor ID . . . : SYSTEM1
 _
 _          Dyn + --Entry +--
 _      CHPID Switch Switch Port
 _      30    4D     4D     C0
 _      31    4D     4D     C1
 _      32    4D     4D     C2
 _      33    4D     4D     C3
 _      34    5D     5D     C0
 _      35    5D     5D     C1
 _      36    5D     5D     C2
 _      37    5D     5D     C3
```

The following examples define the LPAR Access List:.

```
   Add  Modify  View  Filter  Backup  Query  Help
 _                    Define Candidate List
 S      Select one or more partitions for inclusion in the candidate
 list.
 P      Channel path ID. . : 30       Channel path type . . : CNC
 C      Operation mode . . : SHR      Number of CHPIDs. . . : 8
        / Partition Name  Number Usage Description
 /      / LPAR1           1      OS    Production LPAR 1
 /      / LPAR2           2      OS    Production LPAR 2
```

The following example defines the logical control units:

```
   Add  Modify  View  Filter  Backup  Query  Help
                            Select Processor / Control Unit
        Modify  View  Query   Help
 _                                 Add Control Unit
 S      Specify or revise the following values.
 C      Control unit number. . : 001         Type . . . . . . . : 3990
        Processor ID . . . . . : SYSTEM1     Production System 1
 /
        Channel path IDs . . . . 30   32   34   36   __   __   __   __   +
 c      Link address . . . . . . . D0   D1   D0   D1   __   __   __   __   +
 _
 *      Unit address . . . . . . . .00   __   __   __   __   __   __   __
 +
        Number of units. . . . .  256_  __   __   __   __   __   __   __
        Logical address. . . . .    0 + (same as CUADD)
 C      Protocol . . . . . . . . . .   + (D, S, or S4)
        I/O concurrency level. . + (1, 2, or 3)
 F
```

```
  Add  Modify  View  Filter  Backup  Query  Help
                          Select Processor / Control Unit
       Modify  View  Query   Help
 _                               Add Control Unit
 S     Specify or revise the following values.
 C     Control unit number. . : 002         Type . . . . . . . . : 3990
       Processor ID . . . . . : SYSTEM1     Production System 1
 /
       Channel path IDs . . . . 31   33   35   37   __   __   __   __   +
 c     Link address . . . . . . D2   D3   D2   D3   __   __   __   __   +
 _
 *     Unit address . . . . . . 00   __   __   __   __   __   __   __   +
       Number of units. . . . . 256_  __   __   __   __   __   __   __
       Logical address. . . . . 0 + (same as CUADD)
 C     Protocol . . . . . . . .   + (D, S, or S4)
       I/O concurrency level. .   + (1, 2, or 3)
 F
```

```
  Add  Modify  View  Filter  Backup  Query  Help
                          Select Processor / Control Unit
       Modify  View  Query   Help
 _                               Add Control Unit
 S     Specify or revise the following values.
 C     Control unit number. . : 003         Type . . . . . . . . : 3990
       Processor ID . . . . . : SYSTEM1     Production System 1
 /
       Channel path IDs . . . . 30   32   34   36   __   __   __   __   +
 c     Link address . . . . . . D0   D1   D0   D1   __   __   __   __   +
 _
 *     Unit address . . . . . . 00   __   __   __   __   __   __   __   +
       Number of units. . . . . 256_  __   __   __   __   __   __   __
       Logical address. . . . . 1 + (same as CUADD)
 C     Protocol . . . . . . . .   + (D, S, or S4)
       I/O concurrency level. .   + (1, 2, or 3)
 F
```

```
  Add  Modify  View  Filter  Backup  Query  Help
                          Select Processor / Control Unit
       Modify  View  Query   Help
 _                               Add Control Unit
 S     Specify or revise the following values.
 C     Control unit number. . : 004         Type . . . . . . . . : 3990
       Processor ID . . . . . : SYSTEM1     Production System 1
 /
       Channel path IDs . . . . 31   33   35   37   __   __   __   __   +
 c     Link address . . . . . . D2   D3   D2   D3   __   __   __   __   +
 _
 *     Unit address . . . . . . 00   __   __   __   __   __   __   __   +
       Number of units. . . . . 256_  __   __   __   __   __   __   __
       Logical address. . . . . 1 + (same as CUADD)
 C     Protocol . . . . . . . .   + (D, S, or S4)
       I/O concurrency level. .   + (1, 2, or 3)
 F
```

```
  Add  Modify  View  Filter  Backup  Query  Help
                           Select Processor / Control Unit
       Modify  View  Query   Help
_                                 Add Control Unit
S      Specify or revise the following values.
C      Control unit number. . : 005         Type . . . . . . . . : 3990
       Processor ID . . . . . : SYSTEM1     Production System 1
/
       Channel path IDs . . . . 30   32   34   36   __   __   __   __   +
c      Link address . . . . . . . D0   D1   D0   D1   __   __   __   __   +
_
*      Unit address . . . . . . . .00   __   __   __   __   __   __   __
+
       Number of units. . . . .  256_  __   __   __   __   __   __   __
       Logical address. . . . .    2 + (same as CUADD)
C      Protocol . . . . . . . . . . . + (D, S, or S4)
       I/O concurrency level. . + (1, 2, or 3)
F
```

```
  Add  Modify  View  Filter  Backup  Query  Help
                           Select Processor / Control Unit
       Modify  View  Query   Help
_                                 Add Control Unit
S      Specify or revise the following values.
C      Control unit number. . : 006         Type . . . . . . . . : 3990
       Processor ID . . . . . : SYSTEM1     Production System 1
/
       Channel path IDs . . . . 31   33   35   37   __   __   __   __   +
c      Link address . . . . . . D2   D3   D2   D3   __   __   __   __   +
_
*      Unit address . . . . . . 00   __   __   __   __   __   __   __   +
       Number of units. . . . . 256_  __   __   __   __   __   __   __
       Logical address. . . . . 2 + (same as CUADD)
C      Protocol . . . . . . . .   + (D, S, or S4)
       I/O concurrency level. .   + (1, 2, or 3)
F
```

```
  Add  Modify  View  Filter  Backup  Query  Help
                           Select Processor / Control Unit
       Modify  View  Query   Help
_                                 Add Control Unit
S      Specify or revise the following values.
C      Control unit number. . : 007         Type . . . . . . . . : 3990
       Processor ID . . . . . : SYSTEM1     Production System 1
/
       Channel path IDs . . . . 30   32   34   36   __   __   __   __   +
c      Link address . . . . . . D0   D1   D0   D1   __   __   __   __   +
_
*      Unit address . . . . . . 00   __   __   __   __   __   __   __   +
       Number of units. . . . . 256_  __   __   __   __   __   __   __
       Logical address. . . . . 3 + (same as CUADD)
C      Protocol . . . . . . . .   + (D, S, or S4)
       I/O concurrency level. .   + (1, 2, or 3)
F
```

```
  Add  Modify  View  Filter  Backup  Query  Help
                        Select Processor / Control Unit
       Modify  View  Query   Help
_                              Add Control Unit
S      Specify or revise the following values.
C      Control unit number. . : 008        Type . . . . . . . : 3990
       Processor ID . . . . . : SYSTEM1     Production System 1
/
       Channel path IDs . . . . 31   33   35   37   __   __   __   __   +
c      Link address . . . . . . D2   D3   D2   D3   __   __   __   __   +
_
*      Unit address . . . . . . 00   __   __   __   __   __   __   __   +
       Number of units. . . . . 256_  __   __   __   __   __   __   __
       Logical address. . . . . 3 + (same as CUADD)
C      Protocol . . . . . . . .   + (D, S, or S4)
       I/O concurrency level. .   + (1, 2, or 3)
F
```

The following examples define devices 100-1FF:

```
  Add  Modify  View  Filter  Backup  Query  Help
                        Add Device
       Specify or revise the following values.
       Device number . . . . . . . . . 100_   (0000 - FFFF)
       Number of devices . . . . . . . 256__
       Device type . . . . . . . . . . 3390___________  +
       Serial number . . . . . . . . . _______________
       Description . . . . . . . . . . _______________
       Connected to CUs. . . 001    002  ___  ___  ___  ___  ___  ___  +
```

```
        Device / Processor Definition
 Modify     View     Query     Help
 __________________________________________________________
                                                           Row 1 of 1
 Select one or more processors, then select an action.
 Device number . .  : 0100          Number of devices . . : 256
 Device type . . .  : 3390
                                                                *****
/   Processor ID   UA + Time-Out  STADET  Preferred Explicit Device
                                          CHPID + Candidate List
_  SYSTEM1         00   No        Yes     __      No
```

```
  Add  Modify  View  Filter  Backup  Query  Help
                        Add Device
       Specify or revise the following values.
       Device number . . . . . . . . . 200_   (0000 - FFFF)
       Number of devices . . . . . . . 256__
       Device type . . . . . . . . . . 3390___________  +
       Serial number . . . . . . . . . _______________
       Description . . . . . . . . . . _______________
       Connected to CUs. . . 003    004  ___  ___  ___  ___  ___  ___  +
```

```
       Device / Processor Definition
Modify     View     Query     Help
__________________________________________________________
                                                  Row 1 of 1
Select one or more processors, then select an action.
Device number . .  : 200          Number of devices . . : 64
Device type . . .  : 3390
                                                       *****
/   Processor ID   UA + Time-Out  STADET  Preferred Explicit Device
                                          CHPID + Candidate List
_  SYSTEM1         00   No        Yes     __      No
```

```
 Add  Modify  View  Filter  Backup  Query  Help
                         Add Device
      Specify or revise the following values.
      Device number . . . . . . . . . 300_   (0000 - FFFF)
      Number of devices . . . . . . . 256__
      Device type . . . . . . . . . . 3390___________  +
      Serial number . . . . . . . . . _______________
      Description . . . . . . . . . . _______________
      Connected to CUs. . . 005    006  ___  ___  ___  ___  ___  ___  +
```

```
       Device / Processor Definition
Modify     View     Query     Help
__________________________________________________________
                                                  Row 1 of 1
Select one or more processors, then select an action.
Device number . .  : 0180          Number of devices . . : 300
Device type . . .  : 3390
                                                       *****
/   Processor ID   UA + Time-Out  STADET  Preferred Explicit Device
                                          CHPID + Candidate List
_  SYSTEM1         00   No        Yes     __      No
```

```
Add  Modify  View  Filter  Backup  Query  Help
                         Add Device
      Specify or revise the following values.
      Device number . . . . . . . . . 400_   (0000 - FFFF)
      Number of devices . . . . . . . 256__
      Device type . . . . . . . . . . 3390___________  +
      Serial number . . . . . . . . . _______________
      Description . . . . . . . . . . _______________
      Connected to CUs. . . 007    008  ___  ___  ___  ___  ___  ___  +
```

```
       Device / Processor Definition
Modify     View     Query     Help
_________________________________________________________
                                                  Row 1 of 1
Select one or more processors, then select an action.
Device number . .  : 01C0          Number of devices . . : 400
Device type . . .  : 3390
                                                       *****
/   Processor ID   UA + Time-Out  STADET  Preferred Explicit Device
                                          CHPID + Candidate List
_  SYSTEM1         00   No        Yes     __      No
```

Appendix C Troubleshooting

Alternative Restoration

DF/DSS cannot do a standalone restore to Shared Virtual Array 3380K (6EA) devices. Because DF/DSS issues a Sense ID command, to which Shared Virtual Array responds as a K device, DF/DSS never issues a Read Device Characteristics command to get the number of cylinders. Therefore, the host Define Extent parameters have an ending extent for a K device, which Shared Virtual Array rejects.

If the customer uses DF/DSS and does not have an alternative system from which to do restores, the customer or CSE must work around this incompatability to restore the data to a real K device. When a customer or CSE is defining (configuring) initially, he/she must do one of the following:

1. Use K devices and not use all of the cylinders
2. Use KE devices. (KE devices are K devices plus the number of cylinders defined.) If the customer must restore to 3380K (6EA) devices, then the customer or CSE must restore to a K device and then zap the DIRF bit and the VTOC, which forces a free space recalculation.

This page intentionally left blank

Appendix D SCSI Open System Attachment

Note: In spite of using a fibre optic cable, ICF cards cannot be use to connect to SCSI devices via an XSA.

Introduction

The StorageTek Shared Virtual Array Extended SCSI Attach Feature (SVA Attach Feature (or XSA, model number EXSA-001)) enhances the SVA's capability by providing open systems SCSI based host attach. Figure D-1 shows how the SVA Attach Feature connects open systems host and a mainframe to the SVA.

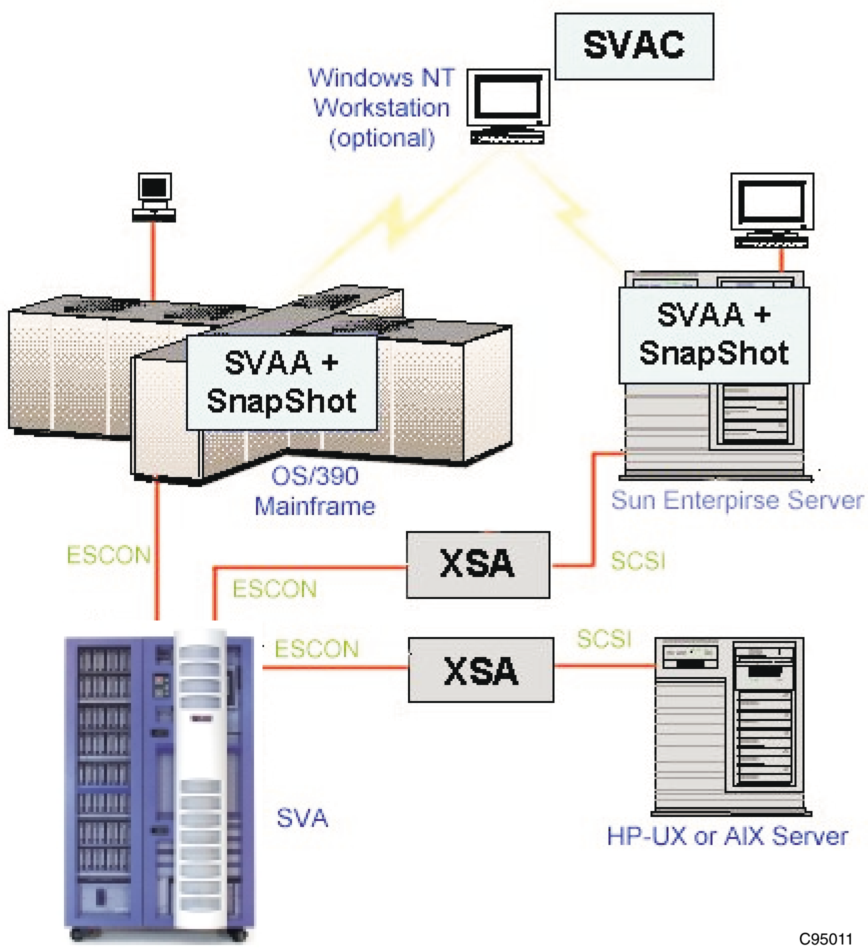


Figure D-1 Architecture Overview

This feature also provides a highly reliable mechanism for extending the distance of a pair of 'ultra and wide' SCSI buses up to three kilometers. This allows a host processor to connect to a Shared Virtual Array disk storage system that is across the room, on a different floor of a building, or in another building, instead of the normal limitation of under 25 meters. Figure D-2 on page 118 shows the SVA Attach Feature Unit.

C51177

Figure D-2 SVA Attach Feature External Structure

SVA Attach Feature units have two differential SCSI interfaces, two LED based fibre optic interfaces, and are packaged in a Pentium-based low profile desktop enclosure.

The SVA Attach Feature maintains the SCSI system timing requirements while emulating the Shared Virtual Array to the host. Through a unique and proprietary fibre optic connection, the actual Shared Virtual Array may be up to three kilometers away. The term 'fibre' is used throughout the rest of this document to describe this connection.

The SCSI cable requirement (25 meters or less) is maintained because the SVA Attach Feature acts as the Shared Virtual Array.

The physical connection is in total conformance with dimensional and optical ESCON fibre specifications as outlined in *ESCON I/O Interface - Physical Layer*, SA23-0394-00, from IBM.

Additionally, the SVA Attach Feature provides a number of features designed to protect the data as it is transferred, including:

1. The SVA Attach Feature uses two bytes of Cyclic Redundancy Check (CRC) per data frame, the same as the fibre protocol.
2. All data busses are protected by parity, as well as other circuitry which detects conditions such as under/over-run.
3. The fibre optic interface uses 8B/I0B encoding[1], and any single invalid code detected by the fibre optic receiver will cause an operation to be flagged and reported as being in error.

At the end of each operation, the SVA Attach Feature unit attached to the host checks that no errors were reported either by the Shared Virtual Array or itself. The SVA Attach Feature unit reports any errors detected using standard SCSI check status/sense data reporting techniques, and these

1. For a complete explanation of the 8B/10B encoding see “IBM Enterpise Systems Architecture/390 fibre Interface, SA22-7202”

errors (if there were any) can be diagnosed as required. A complete listing of SVA Attach Feature-generated sense data is included in Chapter 5.

XSA Hardware Features

The XSA provides:

- SVA connectivity to Sun enterpirse servers, HP 9000 servers, and IBM RS/6000 Enterprise servers,
- dual Ultra-SCSI and dual, multiplexed, ESCON-encapusulated-SCSI connections,
- attachment to open systems platforms quickly and easily over extended distances of up to three kilometers away,
- and attachment to open system platforms up to 26 kilometers away with ESCON directors in static mode.

SCSI Addressing Concepts

Host Bus Adapters/Initiators

The terms “Host Bus Adapter” and “Initiator” mean essentially the same thing. Typically this is a card within the host processor that issues commands on the SCSI bus. In the mainframe world, this is the rough equivalent of a channel.

Domains

The Domain Concept

StorageTek uses a concept called "domain" to allow open systems hosts access to blocks of logical devices (the domains) within an SVA.

A domain is another layer of addressing but this layer is manually set in the SVA Attach Feature and in the SVA. This layer of addressing divides the SVA into “domains of access.” Each open systems host initiator is connected to a domain, giving it access to the devices that have been configured within that domain - an open systems host cannot see the devices in domains other than the on to which it is attached (this restriction is not true of the mainframe attached to the SVA, but will be addressed later). Multiple initiators may be connected to the same domain by the SVA Attach Feature. There can be up to 16 (0-15) domains per SVA, with each domain having up to 120 devices (LUNs). There is a limitation of 1024 total devices available within an SVA[2].

Figure D-3 shows open system platform A connecting to domain one, and open system platform B connecting to domain two of the attached SVA. The mainframe would be using the remaining domains zero and three.

2. Using all allowed domains, targets, and LUNs, there are more than 1024 logical devices but the SVA has a limit of 1024 logical devices.

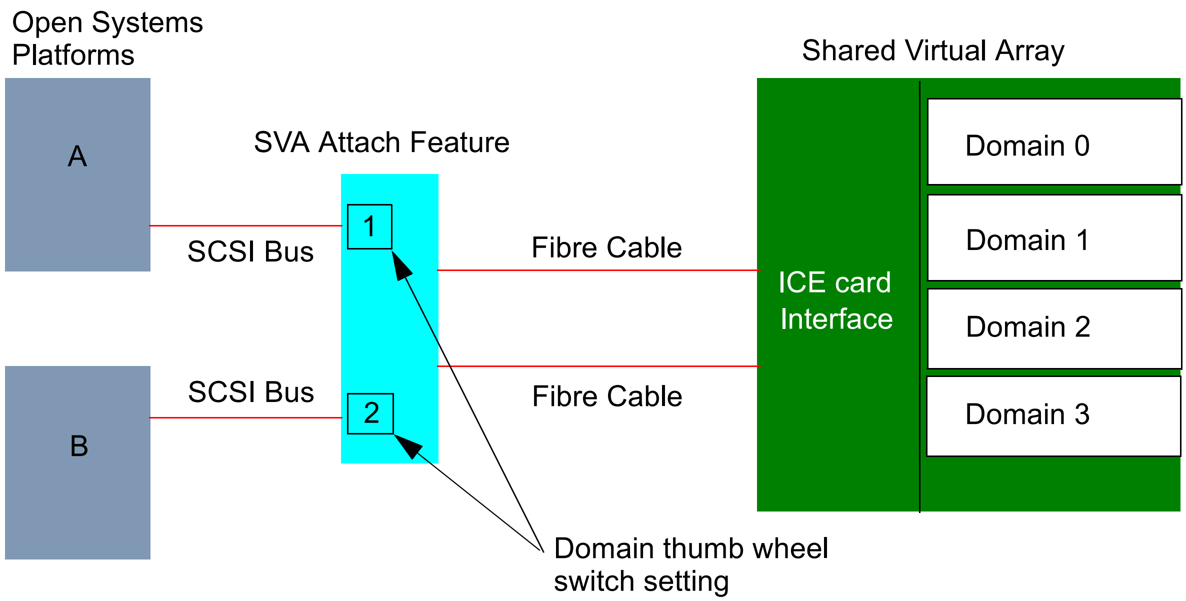


Figure D-3 Basic Domain *Use*

Note: In the above figure, a domain can represent up to 120 logical devices (15 Targets X 8 LUNs = 120 Logical Devices). There is a limit of 1024 logical devices with an SVA.

Domain Numbering

The domain number is configured in two places. One is in the configuration of the SVA, and the other is in the configuration of the XSA. Both of these are transparent to the host platform.

The domain number, configured in the XSA, is attached to the I/O request and sent to the SVA (along with the XSA's serial number) to indicate which domain within the SVA the I/O request is addressing. Remember, there are two SCSI ports on an XSA, and there are two fibre cables connecting the XSA to the SVA. An I/O request can come down either fibre cable and there is no correlation between the fibre cables and the SCSI ports. The XSA serial number is used to make sure the information is returned to the correct XSA since more than one XSA with the same domain numbers can be attached to an SVA (see "Domain Sharing" below).

When the I/O request is returned to the XSA, the domain number is sent along with the I/O request so the XSA will know to which host initiator to return the information.

The domain number is never seen by the open systems host. From the open systems host's point of view, there is just a target and logical unit number involved in an I/O operation. In the example shown in Figure D-4, the open systems host only sees two targets with three logical units attached to each target. Again, domain number never enters into the picture from the open systems host's point of view.

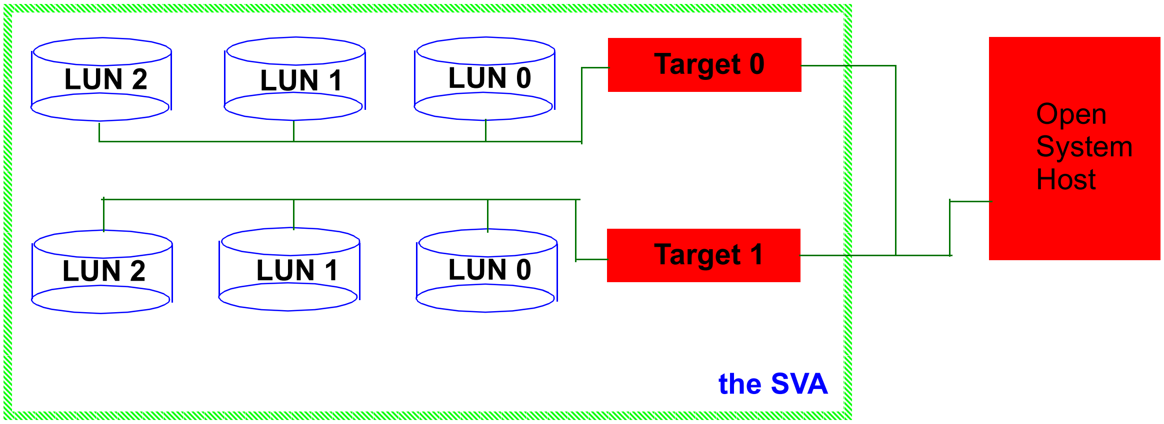


Figure D-4 An SVA As Viewed By An Open Systems Host

Domain Sharing

While it is possible to attach more than one XSA to an SVA and use the same domain numbers in the XSAs (the open systems hosts are sharing the domains and possibly the target numbers in those domains), it is not a good idea to do so since it is very easy for someone to make a mistake and have the two hosts attempt to share a LUN. Two or more open systems host cannot share a LUN - attempting to do so will corrupt the data stored on the LUN - with the exception of host device sharing applications. To further avoid the possibility of data corruption, target numbers within a domain should not be shared.

StorageTek generally recommends that each host bus adaper have a unique domain number, with the exception of failover/path-balancing.

The mainframe attached to the SVA can "see" all of the domains. This is required so the customer can use features like SnapShot. The mainframe effectively shares all of the domains with the open systems hosts. However, those domains set aside for open systems attachment have the read/write bit set to read only for the mainframe. Solaris volumes appear to the mainframe as large and completely full volumes. SnapShot merely copies the pointers from the open system volumes and does not write to them.

Note: To have a mainframe restore a volume (LUN), the read/write bit will have to be set to allow the mainframe to write to that volume. It should be reset to read only after the restore is compete.

Targets

The term "Target" refers to the interface between the SCSI host adapter and the Logical Unit(s). A "wide" attachment allows for 16 targets per HBA. In actual practice this number is limited to fifteen for a "wide" SCSI attachment, or seven for a "narrow" SCSI attachment as the host bus adapter consumes one of these addresses, usually number seven (the address with the highest SCSI priority).

In the mainframe world, a target equates to a control unit address and a LUN equates to the device address.

Note: In the SVA, only 3390-3 and 3390-9 device types are supported for open systems attach.

Logical Unit Numbers (LUNs)

A Logical Unit Number is a data storage area.

The capacity of each LUN is determined by the size and number of logical devices assigned to it. This logical device attachment is a configuration done internally to the SVA and is not visible to the SCSI host processor though the SCSI host processor can obtain the capacity through the SCSI capacity command.

The SVA has a “Large LUN” option. A “Large LUN” is any LUN with more than one Logical Device included in it. By including more than one Logical Device in a LUN, you can greatly increase the capacity of what the SCSI host views as a single logical data storage area. All of the Logical Devices for a Large LUN must reside within the same RAID array, and all of the Logical Devices must be of the same family.

Note: SVA logical units can be either 3390-3 or 3390-9 emulation, but 3380 emulations are not supported for open systems attach.

Typical Read/Write Operations

The SVA Attach Feature not acts a SCSI cable extender between the open system host(s) and the Shared Virtual Array data storage system, but through the use of the SVA Attach Feature’s internal buffer, also acts as a data transfer speed matching device. Refer to Figure D-5 as you read the descriptions of the read and write operations below.

To accomplish the above, the XSA only translates SCSI data to encapsulated ESCON SCSI data and sends it to the SVA. Data from an SVA to the open systems host is translated from ESCON encapsulated SCSI data to SCSI data. All further data translation of SCSI data is done in the SVA itself.

A Typical Write Operation

- Once the host has indicated to the SVA Attach Feature that it wants to initiate a write operation, the Shared Virtual Array tells the SVA Attach Feature how much data it can accommodate at one time.
- The SVA Attach Feature then reconnects to the host that requested the write operation, and indicates how much data to transfer to the SVA Attach Feature (remember, the host thinks it is talking to the disk storage device directly).

- The host then transfers the indicated amount of data to the SVA Attach Feature, and then to the SVA Attach Feature's buffer at the SCSI transfer rate of up to 40 MB/Sec.
- With the completion of this partial data transfer (the entire amount of data the host may wish to send to disk may not be complete at this time), the SVA Attach Feature disconnects from the host. After the disconnect, the SVA Attach Feature is free to receive other SCSI commands.
- The cycle of connecting, transferring data and disconnecting is repeated until all of the requested data is transferred from the host to the SVA.

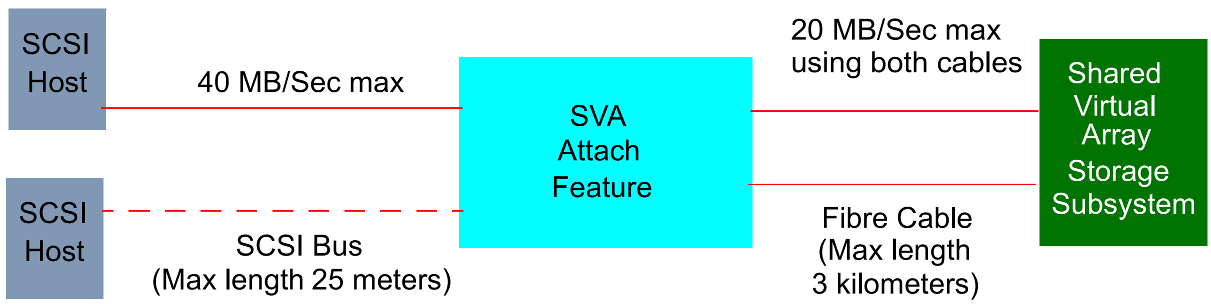


Figure D-5 Data Transfer Block Diagram

Notes: about Figure D-5:

- I/Os are even distributed over the two fibre cables.
- Transfers to and from the Shared Virtual Array are done by 'frame.'
- Data transfers are flow through (the data is not altered on the way through).
- For longer than three kilometers connections, see "Extended Distance Connections" below.

A Typical Read Operation

- A SCSI host connects with the SVA Attach Feature and initiates a read operation.
- The read request is passed to the Shared Virtual Array, and the Shared Virtual Array begins a data transfer to the SVA Attach Feature.
- The cycle of connecting, transferring data and disconnecting is repeated until all of the requested data is transferred from the Shared Virtual Array to the host.

Connections

Distance Restrictions

Standard Connections

The standard connection between a SCSI host and a Shared Virtual Array is shown in Figure D-6. The method of connecting allows for distances of up to three kilometers.

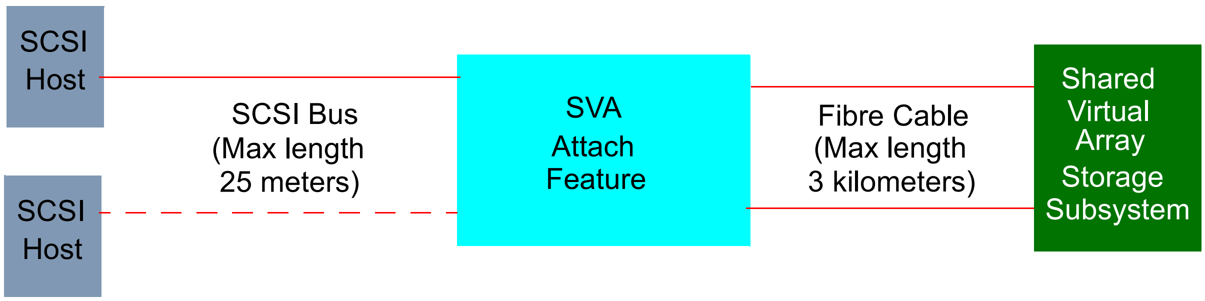


Figure D-6 Connections for Distances Up to 3 KM

Note: For fibre connections beyond 3KM, see the section below.

Extended Distance Connections

An alternative method of connection using ESCON directors allows for connections between a SCSI host and a Shared Virtual Array over distances of up to 26 kilometers (see Figure D-7). Consult a StorageTek marketing representative for the actual details on this type of connection.

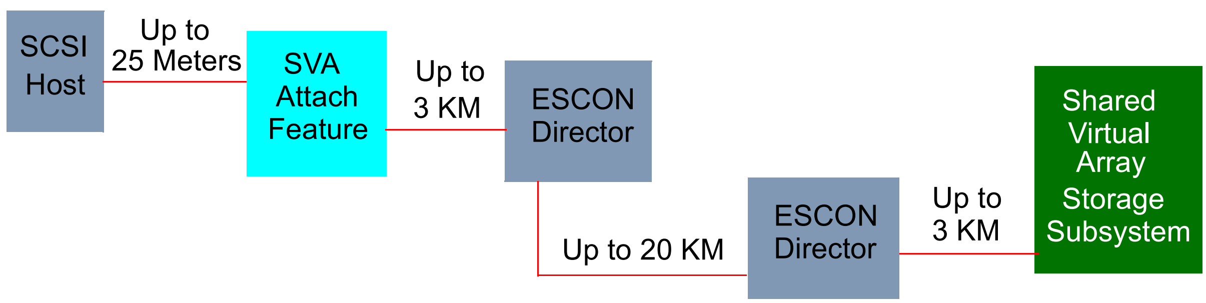


Figure D-7 Connections for Distances Up to 26 KM

Note: The ESCON directors in the above figure must be set to “Static Mode” to prevent them from multiplexing signals. Multiplexing signals will create errors on the link between the Shared Virtual Array and the SVA Attach Feature.

Usage Connections

There are two principal methods of connecting open system hosts to a Shared Virtual Array; connections for availability and connections for connectivity. These are discussed below.

Note: The examples shown below do not reflect all of the ICE Card configuration options available in a Shared Virtual Array; *these are merely examples*. Consult a StorageTek marketing representative for the ICE Card options available in a Shared Virtual Array.

Connections for Availability

The diagram below (Figure D-8) shows a method of connecting two open system hosts to a Shared Virtual Array. If any path fails, there is still a path for the host to access the data.

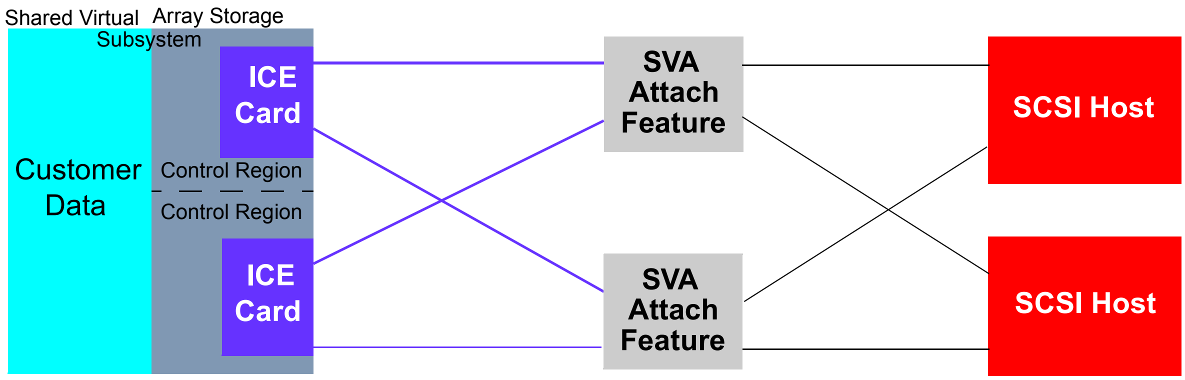


Figure D-8 Connections for Availability

Note: Failover software is required to make this configuration non-disruptive in the event on a component failure. This software will not be available until after GA.

Connections for Connectivity

Figure D-9 on page 125 shows a method of connecting several open system hosts to a Shared Virtual Array. In this case, the emphasis is on connectivity versus performance or availability. Any path failure will result in one or more hosts no longer being able to access the customer data. This method of attachment allows a maximum number of hosts to attach to a Shared Virtual Array with a minimum number of SVA Attach Features.

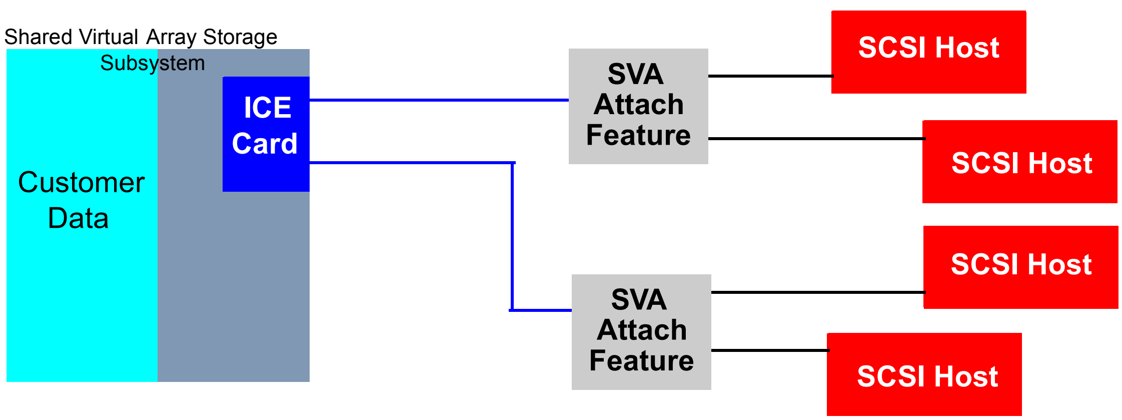
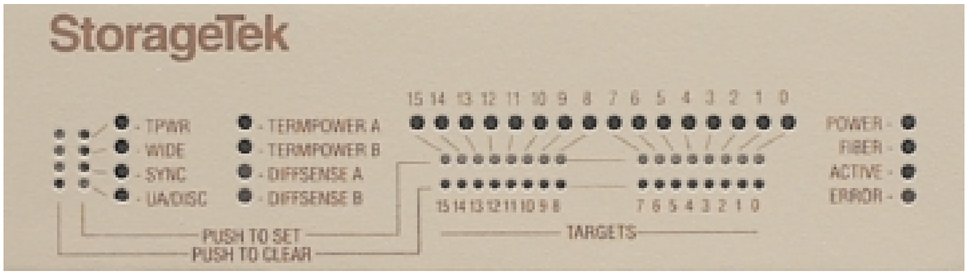


Figure D-9 Connections for Connectivity

Front Panel

Figure D-10 shows the front panel of the SVA Attach Feature. The function/meaning of the switches and indicators are described in the sections below.



C51171

Figure D-10 Front Panel

SVA Attach Feature-initiated Synchronous and Wide Data Transfer Request Negotiations

Enabling the SVA Attach Feature-initiated synchronous data transfer request (SDTR) and wide data transfer request (WDTR) negotiations instructs the SVA Attach Feature to initiate negotiations following SVA Attach Feature power-cycles or SCSI resets. Although enabling this feature is desirable for maintaining data transfer negotiations under all conditions, some (especially older) hosts do not expect targets to initiate negotiation and therefore will not respond correctly. If you are uncertain as to the proper setting of these features in your configuration, please contact your technical support representative for more information.

Notes:

1. The SVA Attach Feature will always respond to host-initiated SDTR or WDTR negotiations up to its maximum settings, except for the special case when the ‘sync’ switch is ‘set’ and ‘wide’ switch is set to ‘clear’, which disables wide data transfer altogether (this is sometimes useful in configurations which employ ‘wide’ devices in a ‘narrow’ cabling scheme).
2. Target IDs 8 - 15 are not valid with narrow cables, and the rules for the use of narrow cabling are beyond the scope of this document.

Setting Front Panel Switches

The SCSI and Features switches on the SVA Attach Feature are rocker type Dual Inline Package (DIP) switches mounted behind holes in the front panel. These switches are changed by inserting the supplied tool (See Figure D-11 on page 130) into the correct hole and pushing until you feel the switch move and see the associated front panel LED change.

Caution: If the switch does not move easily do not force the tool into the hole. Instead you should move the tool around to line it up with the switch.

SCSI Target Switch Settings

The sixteen target switch settings on the SVA Attach Feature front panel will be determined by the customer and must be the same address(es) configured within the attached Shared Virtual Array. When a target switch is “set,” the associated LED lights.

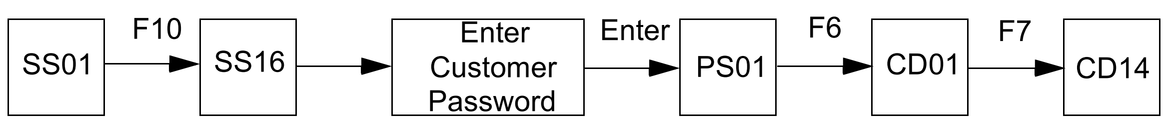
The target switches come factory set to off. They should be turned on as needed.

ICE Card Configuration

Initially, all ICE cards come configured for mainframe ESCON use. One or more ICE cards (for fibre connections to the SVA) must be configured for SCSI use. Once this is done both link ports on the ICE card are available for SCSI and only SCSI use. This should be done during the initial install as it does require an IML of the control region for the ICE card being reconfigured. To configure an ICE card for SCSI use:

Note: The ICE cards need to be configured for 1024 logical devices. Check with your StorageTek CSE to be sure that they are so configured.

1. Go to the Configure ESCON Links screen as per below:



2. Select the card to be reconfigured, using the Cursor Down [F6] and Cursor Up [F7] keys.
3. Enter a "1" to reconfigure the ICE card as SCSI.
4. Press the "Mod ESCN" button [F10].
5. Answer yes (by pressing [F9]) to the question “Reconfigure ICE cards between ESCON/SCSI continue?”
6. Answer yes (by pressing [F9]) to the question “Subsystem will Re-IML. Do you want to continue?”

When the IML completes, the ICE card is ready for SCSI use.

How to Set a Domain

Each SCSI controller in a SCSI host will be connected to one of the two ports on the Extended SCSI Adapter using an Ultra and Wide Differential SCSI cable. The domain (0-15) is set on a "thumb-wheel" switch at the port, on the back of the Extended SCSI Adapter. Devices configured in the SVA with this domain number will be visible to the host through this connection.

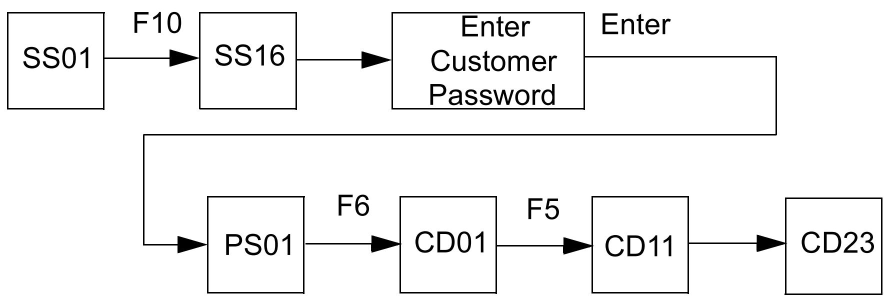
Setting Targets on the Extended SCSI Adapter

On the front panel of the Extended SCSI Adapter are a set of switches for configuring the targets to which the Extended SCSI Adapter will respond.

Each target is controlled by a switch, allowing each Extended SCSI Adapter to answer to only those targets associated with hosts attached to it's ports. The customers should decide which set of targets need to be activated for their configuration. Using the tool provided, press the switch for the "On" position for the targets needed. A green light should appear on the panel for those switched turned "On".

Defining Devices for SCSI Use

1. Configuring a device for SCSI use can be done from the LOP or through host software available from StorageTek. To configure from the LOP use the following menu sequence:



2. Select the Functional Device to be configured using the Cursor and/or Page Up ([F6]) and Down keys ([F7]).
3. Press Enter.
4. In the Device Configuration screen (CD32), enter:
   - Name
   - Type (6 for a 1.2GB device and 7 for a 3.5GB device),
   - Format Type (1 for SCSI format, 512 for the blocking factor, the SCSI bus address (domain), target and LUN, 1 for valid to make this device the only one with that Domain)
   - SCSI Block Size
   - Domain
   - Target ID
   - LUN
   - Valid (set to 1)

Note: The values for the below are specifice to the installation.

   - CKD Enable
   - SCSI Enable
   - CKD Write Protect
   - Cache Enable
   - Code Active
   - DASD Fast Write
   - Priveleged

5. Once all the fields have been correctly filled in press the "Modify Functional Device" (F10) key. If the status line says "Completed" the device has been successfully configured for SCSI use.

Note: Remember to set the SCSI Enable attribute to "1" or it won't be visible from the host!

Usage of the External SCSI Adapter

The External SCSI Adapter acts as an extender from Differential SCSI distances of 25 meters to 3 Km using fibre. It also provides flexibility, security, capacity and throughput depending on the number of SCSI hosts or controllers connected, the number of Extended SCSI Adapters installed, the number of ICE cards configured for SCSI use, and the number of fibres connected to each Extended SCSI Adapter. The total number of devices available to a host is based on the number of controllers in the host, domains, and devices configured within those domains. For each domain there are a possible 120 devices (15 targets x 8 LUNs). Each External SCSI Adapter can support 2 domains or 240 devices. For maximum throughput a single SCSI input and two fibre outputs to different ICE cards is recommended (see Figure D-8 on page 125).

Basic Host Configuration

Each Operating System handles the addition of devices to the configuration differently. In general the process will be something like:

- Connect all the SCSI and fibre cables
- Configure the SVA Attach Feature for domains and targets
- Configure the ICE cards for SCSI use
- Configure the devices desired with the appropriate DTL's in the SVA
- Make the new devices known to the controller and Operating System

Once the Operating System can access the devices they can be partitioned, labeled, and have file systems created on them (if necessary). The exact commands to do this vary by Operating System, consult a StorageTek CSE for assistance.

Setting Front Panel Switches

The SCSI and Features switches on the SVA Attach Feature are rocker type Dual Inline Package (DIP) switches mounted behind holes drilled in the front panel. These switches are changed by inserting the supplied tool into the correct hole and pushing until you feel the switch move and see the associated front panel LED change.

Caution: If the switch does not move easily do not force the tool into the hole. Instead you should move the tool around to line it up with the switch.

SCSI Target Switch Settings

The sixteen target switch settings on the SVA Attach Feature front panel will be determined by the customer and must be the same address(es) configured within the attached Shared Virtual Array. When a target switch is “on,” the associated LED lights.

The target switches come factory set to off. They should be turned on as needed.

To set or reset one of these switches, ***only used the supplied tool*** (see Figure D-11 below). If you do not have the supplied tool, contact your StorageTek CSE.

Configuring the SVA Attach Feature

Note: This configuration process is done after the unit is powered on so you can see the LEDs on the front panel, making it easier to tell which switches have been set or cleared.

1. Set the targets on the front panel as per the pre-installation check list. Use the tool provided with the SVA Attach Feature to set the switches (see Figure D-11).

C51176

Figure D-11 SVA Attach Feature Configuration Tool

2. Verify that the following front panel lights are in the following state:
   - TPWR = ON
   - WIDE = ON
   - SYNC = ON
   - UA/DISC = OFF

Set the domain switches on the rear of the unit (next to the SCSI cable attachment connectors). See “Domains” on page 119 for more information on Domains.

SVA Attach Feature Front Panel Feature Switch Settings

Note: The factory settings are usually correct. The top three are on and the bottom one is off.

Certain SVA Attach Feature operating features are set by the front panel DIP switches. A description of each of these features is shown in Table 7-1, below.

Table 7-1 Feature Switch Settings

Switch	Position	Description
TPWR	SET	SVA Attach Feature-supplied diode-protected Terminator Power ENabled
	CLEAR	SVA Attach Feature-supplied diode-protected Terminator Power DISabled
WIDE	SET	SVA Attach Feature-initiated wide data transfer request ENabled (Wide)
	CLEAR	SVA Attach Feature-initiated wide data transfer request DISabled (Narrow)
SYNC	SET	SVA Attach Feature-initiated synchronous data transfer request ENabled
	CLEAR	SVA Attach Feature-initiated synchronous data transfer request DISabled
UA/DISC	SET	SVA Attach Feature unit attention DISabled SVA Attach Feature disconnects DISabled (*rarely used*)
	CLEAR	SVA Attach Feature unit attention ENAabled SVA Attach Feature Disconnects ENabled

Notes:

Most installations will set:

- TPWR = ON
- SYNC = ON
- WIDE =
  - ON for Wide
  - OFF for Narrow)
- UA/DISC = OFF

OFF is indicated by the adjacent LED being dark.
ON is indicated by the adjacent LED being ‘illuminated’.
For some older hosts that do not support target initiated negotiations, you will have both wide and sync set to OFF.

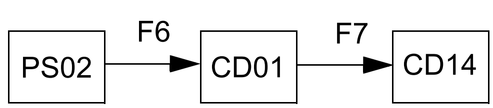
Front Panel Indicators

Table 7-2 Front Panel Indicators

Indicator	Description (when illuminated)
Power	The SVA Attach Feature has power.
Fibre	ON indicates that there is at least one good fibre connection to the Shared Virtual Array. OFF indicates no optical power detected. BLINKING indicates that while the fibre is connected the fibre interface is not in an online state.
Active	An activity light. If the light is on or blinking, it means that the box is servicing host I/Os.
Error	An error condition exists. A customer should contact the StorageTek CSE if this light is ON.
TERMPOWER A/B	The power to the terminators is on. Both these lights should be ON for proper operation. Power can be supplied by either this unit, another target, or an initiator.
DIFFSENSE A/B	Indicates that a single ended device has been connected to a differential SCSI bus, and the SVA Attach Feature drivers are disabled. Proper operation will resume when the offending single ended device is removed and the light goes OFF.

Shared Virtual Array SCSI Configuration

1. Login to the Shared Virtual Array operator panel.
2. From the PS02 screen, use the following menu sequence to get to the screen CD14:



3. Referring the Shared Virtual Array 9393 System Assurance Guide, use [F7] to move the selector to the ICE card you want to change to SCSI.
4. Then use the [#1] key to put the 1 next to the ICE card(s). Continue to repeat steps 3 and 4 for all ICE cards you need to change to SCSI.
5. Press [F10] to modify (save) the changes.
6. At the prompt “Are all channels varied offline at the host?” press [F9] for YES. (It will ask for each card changed.) The system will re-IML at this time.

7. After the IML is complete, verify the results by returning to screen CD14 (see above).
8. Press [F3] to return to the previous menu. This will return you to screen CD01.
9. Press [F5]. This takes you to screen CD11.
10. Press [F7] to go to screen CD23.

Note: If you are using Shared Virtual Array host software you will require a "privileged" ECAM functional device.

11. Use [F7] to move the cursor to the FDID outlined in the planning guide for the ECAM device.
12. Press [Enter]
13. The cursor should be in the name field. Enter the FDID name.
14. Press [F7] to move the cursor down. This will move the cursor to the next field (type).

Note: Partitions should be gone.

15. Use the [#6] key to enter the number 6 as the type. This is a 3390 model 3 which will be set to SCSI-A in the next step.
16. Press [F7] to move the cursor to the next field. This is the Format Type Field.
17. Use the [#1] to enter 1 for a SCSI device.
18. Using [F7] to move the cursor and the [#1] through [#F] keys, fill in the following fields from the worksheet:

    SCSI Block Size

    SCSI Id-Bus_id (Domain)

    Target_id

    LUN

    Index (Can only be zero - reserved for future use)

    Valid

    CKD Enable

    SCSI Enable

    CKD Write Protect

    SCSI Write Protect

    Cache Active

    DASD Fast Write

    Privileged

Note: If you are using Shared Virtual Array host software you will require a "privileged" ECAM functional device.

19. Press [F10] to modify (saves the information you just entered).

20. Press [F3] until you get back out to screen SS01.
21. Press [F3] one more time to exit keyboard master.

Appendix E Initializing SVA SCSI Volumes for OS/390

Introduction

The StorageTek Shared Virtual Array (SVA) has the ability to store data from OS/390 using the standard IBM format, as well as from Sun's Solaris operating system, using a SCSI interface. Usually, SCSI and IBM storage formats are not compatible, but through the following process, StorageTek can make SVA SCSI volumes available to OS/390.

SVA volumes are first formatted as standard OS/390 DASD volumes. When SCSI data is written to an SVA it is simply written 'as is' to the volume, bypassing the VTOC and VTOC Index. As such no, data sets or data are cataloged in the VTOC and the volume appears empty to OS/390. If the volume is varied online to OS/390, the SCSI data will be overwritten by new data from an OS/390 system. We can eliminate this problem by overlaying the volume with several data sets that encapsulate the SCSI data, and a control area. This makes the volume appear full to OS/390, and so can be safely brought online. The included JCL runs the appropriate reformatting jobs.

Depending on the type of devices you have, 3390 Mod 3 or Mod 9, the reformatting will be slightly different. In either case, a data set called the OVDS is allocated for 17 tracks starting at cylinder 0, track 13. Note that this data set starts at this location to leave room for OS/390 to allocate a VVDS should the need arise for VSAM or SMS use. For 3390 Mod 3 volumes, a data set called the OVDTA is allocated starting at cylinder 2 track 0 and filling the remainder of the volume. This is where the SCSI data resides. Because of its large size, for 3390 Mod 9, the OVDTA is divided into three data sets called OVDTA.P00, OVDTA.P01, and OVDTA.P02, that occupy consecutive space starting at cylinder 2, track 0.

Table E-1 shows the data sets written for volumes formatted as 3390 Mod 3 (SCSI A), Figure E-1 on page 136 shows the layout of the volume.

Table E-1 3390 Mod 3 Data Sets

Data Set Name	Position	Explanation
hlq.OVDS.V*volser*	Cyl 0 Trk 13 through Cyl 1 Trk 14	17 tracks in reserved for future use.
hlq.OVDTA.V*volser*.P00	Cyl 2 Trk 0 to End of Volume	Fills the remainder of the volume and contains the SCSI data.

Figure E-1   3390 Mod 3 Layout

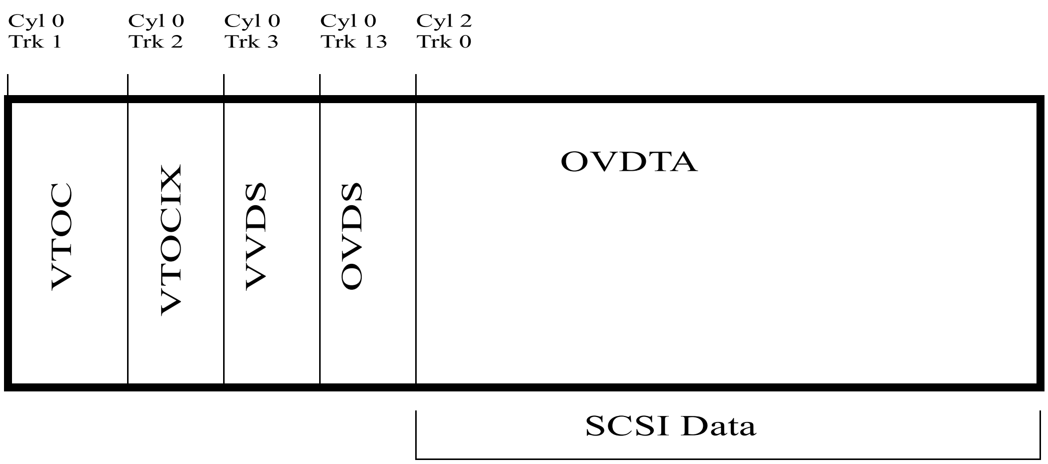
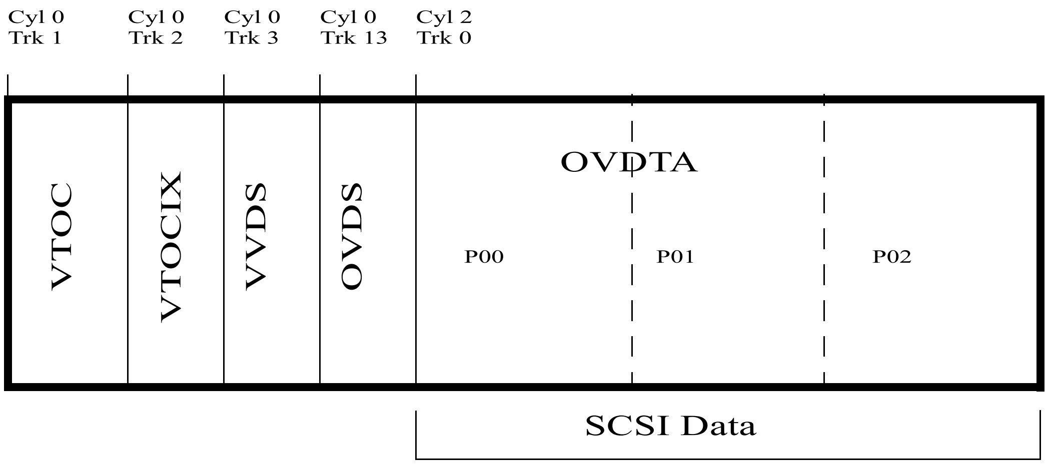


Table E-2 shows the data sets written for volumes formatted as 3390 Mod 9 (SCSI B), Figure E-2 shows the layout of the volume.

Table E-2 3390 Mod 3 Data Sets

Data Set Name	Position	Explanation
hlq.OVDS.V*volser*	Cyl 0 Trk 13 through Cyl 1 Trk 14	17 tracks in reserved for future use.
hlq.OVDTA.V*volser*.P00 *hlq*.OVDTA.V*volser*.P01 *hlq*.OVDTA.V*volser*.P02	Cyl 2 Trk 0 to End of Volume	These three data sets combine to fill the remainder of the volume and contain the SCSI data.

Figure E-2 3390 Mod 9 Layout



Technical Considerations

Please observe the following technical considerations:

- When you create the volume with ICKDSF, you must specify one track each for the VTOC and VTOCIX, and they must be in the first two cylinders of the volume.
- The reformatting JCL may be run either before or after the SCSI data is written to the volume.
- You must have high enough RACF authority to create or change the VTOC, VTOCIX, OVDS, AND OVDTA data sets.
- The data sets created by the reformatting may be renamed.
- You are responsible to ensure that the OVDS and OVDTA data sets are not inadvertently altered, moved, deleted, or migrated, etc.
- Do not use ICKDSF VTOC(OPENDISK) to initialize your SVA volumes.
- SCSI volumes in an SMS environment must have guaranteed space or SMS may not allocate data sets on the specified volumes.

Utilities

The reformatting JCL uses standard IBM OS/390 utilities.

To run the reformatting JCL, you must have access to, and authority to use the following utilities.

- ICKDSF
- AMASPZAP
- IEFBR14

Initialization Procedure

To initialize your SVA SCSI volumes to be compatible with OS/390, perform the following steps. This procedure should be done after the volumes are defined to the SVA, but before they are made accessible to OS/390.

SMS Controlled Volumes

If your volumes are SMS controlled, use the following procedure.

1. Initialize your SVA volume for OS/390. Do this by running the JCL job, scsisa.jcl.
2. Normally, the job will run to completion. If you get the following message, a partially initialized volume exists from a previous unsuccessful run.

   ```
   ICK3120I  VERIFICATION FAILED: VOLUME-SERIAL EXISTS
   ```

   To work around this, modify your JCL by changing `VERIFY(*NONE*)` to `NOVERIFY` in each INIT statement, and run the job again.

   When you re-run the JCL, you will get the following message:

```
ICK003D  REPLY U TO ALTER VOLUME device CONTENTS, ELSE T
```

   Reply 'U' to continue, and you will receive the following message to indicate a successful completion.

```
ICK061I  device VTOC INDEX CREATION SUCCESSFUL:  VOLUME IS IN INDEX FORMAT
```

3. On successful completion of the scsisa.jcl job, vary your volume online to OS/390.
4. Initialize your volumes for OS/390-SCSI compatibility by running one of the following JCL jobs.
   - If your volume is a 3330-3 (SCSIA) SMS, run scsis3b.jcl.
   - If your volume is a 3330-9 (SCSIB) SMS, run scsis9b.jcl.
5. You will get the following message:

```
AMA117D REPLY Y OR N TO UPDATE VTOC volser UCB jobname
```

   Reply 'Y' to continue. On successful completion, you will get the following messages.

```
AMA121I CCHHR UPDATE BY jobname ON volser, chhr,FORMAT4.DSCB
```

AMA121I CCHHR UPDATE BY jobname ON *volser,cchhr,*FORMAT4.DSCB

ICK502I BUILDIX FUNCTION STARTED

ICK503I *device* REQUEST RECEIVED TO CONVERT VTOC TO OSFORMAT

ICK504I *device* VTOC FORMAT IS CURRENTLY IXFORMAT, REQUEST ACCEPTED

IEC604I VTOC CONVERT ROUTINE ENTERED ON device, *volser*,DOS

ICK513 *device* BUILDIX PROCESSING COMPLETED: VTOC IS NOW IN OSFORMAT

ICK502I BUILDIX FUNCTION STARTED

ICK503I *device* REQUEST RECEIVED TO CONVERT VTOC TO IXFORMAT

ICK504I *device* VTOC FORMAT IS CURRENTLY OSFORMAT, REQUEST ACCEPTED

ICK513 *device* BUILDIX PROCESSING COMPLETED: VTOC IS NOW IN IXFORMAT

Non-SMS Controlled Volumes

If your volumes are not SMS controlled, use the following procedure.

1. Initialize your SVA volume for OS/390. Do this by running the JCL job, scsina.jcl.
2. Normally, the job will run to completion. If you get the following message, a partially initialized volume exists from a previous unsuccessful run.

ICK3120I VERIFICATION FAILED: VOLUME-SERIAL EXISTS

To work around this, modify your JCL by changing `VERIFY(*NONE*)` to `NOVERIFY` in each INIT statement, and run the job again.

When you re-run the JCL, you will get the following message:

ICK003D REPLY U TO ALTER VOLUME *device* CONTENTS, ELSE T

Reply 'U' to continue, and you will receive the following message to indicate a successful completion.

ICK061I *device* VTOC INDEX CREATION SUCCESSFUL: VOLUME IS IN INDEX FORMAT

3. On successful completion of the scsina.jcl job, vary your volume online to OS/390.
4. Initialize your volumes for OS/390-SCSI compatibility by running one of the following JCL jobs.
   - If your volume is a 3330-3 (SCSIA) non-SMS, run scsin3b.jcl.
   - If your volume is a 3330-9 (SCSIB) non-SMS, run scsin9b.jcl.
5. You will get the following message:

AMA117D REPLY Y OR N TO UPDATE VTOC *volser* UCB *jobname*

Reply 'Y' to continue. On successful completion, you will get the following messages.

```
AMA121I CCHHR UPDATE BY jobname ON volser, cchhr,FORMAT4.DSCB
AMA121I CCHHR UPDATE BY jobname ON volser, cchhr,FORMAT4.DSCB
```

Initialization JCL

The following section describes the JCL jobs that you will run to initialize your SVA volumes for SCSI/OS/390 compatibility.

scsina.jcl

The following JCL must be run for non-SMS 3390 Mod 3 (SCSIA) or 3390 Mod 9 (SCSIB) volumes. This job writes the VTOC and VTOCIX, and initializes the volume for OS/390 use. See the comments contained in the JCL file for detailed instructions.

```
//jobname JOB 'accounting info'
//*
//* FORMATS MVS NON-SMS SCSIA (3390-3) OR SCSIB (3390-9)
//* VOLUME WITH VTOC AND VTOC INDEX
//*
//* ==> CHANGE UNITADDRESS AND VOLID TO APPROPRIATE VALUES
//*
//STEP01   EXEC PGM=ICKDSF,REGION=4M,PARM='NOREPLYU'
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 INIT UNITADDRESS(unitaddress) -
      VOLID(volid) VERIFY(*NONE*) -
      VTOC(0,1,1) -
      INDEX(0,2,1) -
      NOMAP
/*
```

scsin3b.jcl

The following JCL must be run for non-SMS 3390 Mod 3 (SCSIA) volumes. This job allocates the three data sets that fill the volume for OS/390 use; the VVDS, OVDS control areas and OVDTA data set that encapsulates the SCSI data. See the comments contained in the JCL file for detailed instructions.

```
//jobname  JOB 'accounting info'
//*
//* ALLOCATES DATA SETS, AND ZAPS VTOC
//* ON NON-SMS SCSIA (3390-3) VOLUMES
//*
//BLDMOD3N PROC
//*
//* CREATE VVDS
//*
//ALOC01   EXEC PGM=IEBR14
//TEMPVSAM DD DSN=&TEMPVSAM,DISP=(,DELETE,DELETE),
//            UNIT=3390,VOL=SER=&VOLSER,
//            RECORG=ES,SPACE=(CYL,1),AVGREC=K,LRECL=256
//*
//* ALLOCATE DATA SETS
//*
//ALOC02   EXEC PGM=IEFBR14,COND=(0,NE)
//OVDS     DD DSN=&HLQ..OVDS.V&VOLSER,DISP=(,CATLG,CATLG),
//            UNIT=3390,VOL=SER=&VOLSER,
//            SPACE=(ABSTR,(17,13)),EXPDT=99365,
//            DCB=(DSORG=PS,RECFM=FBS,LRECL=512,BLKSIZE=27648)
//OVDTA00  DD DSN=&HLQ..OVDTA.V&VOLSER..P00,DISP=(,CATLG,CATLG),
//            UNIT=3390,VOL=SER=&VOLSER,
//            SPACE=(CYL,3337,,CONTIG),EXPDT=99365,
//            DCB=(DSORG=PS,RECFM=F,LRECL=512,BLKSIZE=512)
//*
//* ZAP VTOC
//*
//ZAP01    EXEC PGM=AMASPZAP,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
```

```
//SYSLIB   DD DSN=FORMAT4.DSCB,DISP=OLD,
//            UNIT=3390,VOL=SER=&VOLSER,DCB=(KEYLEN=44)
//SYSIN    DD DDNAME=ZAPSYSIN
//*
//         PEND
//*
//* ==> CHANGE FOLLOWING EXEC STATEMENT TO APPROPRIATE VALUES
//* ==>    HLQ    - DATA SET NAME HIGH-LEVEL QUALIFIER
//* ==>    VOLSER - VOLUME SERIAL NUMBER
//* ==>
//* ==> NOTES:
//* ==> - USERID MUST HAVE CREATE AUTHORITY FOR THE HLQ
//* ==> - VOLSER MUST BE MOUNTED AND ONLINE
//*
//  EXEC PROC=BLDMOD3N,HLQ=hlq,VOLSER=volser
//*
//ZAP01.ZAPSYSIN DD *
   CCHHR 0000000102      FMT5
   VER 0000 05050505
   VER 0004 0000000000
   CCHHR 0000000104      OVDS
   VER 0062 000000       DS1LSTAR TTR
   VER 0065 E5A2         DS1TRBAL
   VER 0069 01           DS1EXT1 EXTENT TYPE
   VER 006A 00           DS1EXT1 EXTENT SEQUENCE NUMBER
   VER 006B 0000000D     DS1EXT1 LOWER LIMIT CCHH
   VER 006F 0001000E     DS1EXT1 UPPER LIMIT CCHH
   REP 0062 001002
   REP 0065 0000
   CCHHR 0000000106      OVDTA
   VER 0062 000000       DS1LSTAR TTR
   VER 0065 E5A2         DS1TRBAL
   VER 0069 81           DS1EXT1 EXTENT TYPE
   VER 006A 00           DS1EXT1 EXTENT SEQUENCE NUMBER
   VER 006B 00020000     DS1EXT1 LOWER LIMIT CCHH
```

```
   VER 006F 0D0A000E     DS1EXT1 UPPER LIMIT CCHH
   REP 0062 C38630
   REP 0065 0000
   ABSDUMPT 0000000101 000000010A
//*
```

scsin9b.jcl

The following JCL must be run for non-SMS 3390 Mod 9 (SCSIB) volumes. This job allocates the five data sets that fill the volume for OS/390 use; the VVDS, OVDS control areas and three OVDTA data sets that encapsulates the SCSI data. See the comments contained in the JCL file for detailed instructions.

```
//jobcard  JOB ‘accounting info’
//*
//* ALLOCATES DATA SETS AND ZAPS VTOC
//* ON NON-SMS SCSIB (3390-9) VOLUMES
//*
//BLDMOD9N PROC
//*
//*
//* CREATE VVDS
//*
//ALOC01   EXEC PGM=IEBR14
//TEMPVSAM DD DSN=&TEMPVSAM,DISP=(,DELETE,DELETE),
//            UNIT=3390,VOL=SER=&VOLSER,
//            RECORG=ES,SPACE=(CYL,1),AVGREC=K,LRECL=256
//*
//* ALLOCATE DATA SETS
//*
//ALOC02   EXEC PGM=IEFBR14,COND=(0,NE)
//OVDS     DD DSN=&HLQ..OVDS.V&VOLSER,DISP=(,CATLG,CATLG),
//            UNIT=3390,VOL=SER=&VOLSER,
//            SPACE=(ABSTR,(17,13)),EXPDT=99365,
//            DCB=(DSORG=PS,RECFM=FBS,LRECL=512,BLKSIZE=27648)
//OVDTA00  DD DSN=&HLQ..OVDTA.V&VOLSER..P00,DISP=(,CATLG,CATLG),
//            UNIT=3390,VOL=SER=&VOLSER,
//            SPACE=(CYL,4369,,CONTIG),EXPDT=99365,
```

```
//             DCB=(DSORG=PS,RECFM=F,LRECL=512,BLKSIZE=512)
//OVDTA01  DD DSN=&HLQ..OVDTA.V&VOLSER..P01,DISP=(,CATLG,CATLG),
//             UNIT=3390,VOL=SER=&VOLSER,
//             SPACE=(CYL,4369,,CONTIG),EXPDT=99365,
//             DCB=(DSORG=PS,RECFM=F,LRECL=512,BLKSIZE=512)
//OVDTA02  DD DSN=&HLQ..OVDTA.V&VOLSER..P02,DISP=(,CATLG,CATLG),
//             UNIT=3390,VOL=SER=&VOLSER,
//             SPACE=(CYL,1277,,CONTIG),EXPDT=99365,
//             DCB=(DSORG=PS,RECFM=F,LRECL=512,BLKSIZE=512)
//*
//* ZAP VTOC
//*
//ZAP01    EXEC PGM=AMASPZAP,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//SYSLIB   DD DSN=FORMAT4.DSCB,DISP=OLD,
//             UNIT=3390,VOL=SER=&VOLSER,DCB=(KEYLEN=44)
//SYSIN    DD DDNAME=ZAPSYSIN
//*
//          PEND
//*
//* ==> CHANGE FOLLOWING EXEC STATEMENT TO APPROPRIATE VALUES
//* ==>    HLQ    - DATA SET NAME HIGH-LEVEL QUALIFIER
//* ==>    VOLSER - VOLUME SERIAL NUMBER
//* ==>
//* ==> NOTES:
//* ==> - USERID MUST HAVE CREATE AUTHORITY FOR THE HLQ
//* ==> - VOLSER MUST BE MOUNTED AND ONLINE
//*
//  EXEC PROC=BLDMOD9N,HLQ=hlq,VOLSER=volser
//*
//ZAP01.ZAPSYSIN DD *
   CCHHR 0000000102      FMT5
   VER 0000 05050505
   VER 0004 0000000000
   CCHHR 0000000104      OVDS
```

```
VER 0062 000000       DS1LSTAR TTR
VER 0065 E5A2         DS1TRBAL
VER 0069 01           DS1EXT1 EXTENT TYPE
VER 006A 00           DS1EXT1 EXTENT SEQUENCE NUMBER
VER 006B 0000000D     DS1EXT1 LOWER LIMIT CCHH
VER 006F 0001000E     DS1EXT1 UPPER LIMIT CCHH
REP 0062 001002
REP 0065 0000
CCHHR 0000000106      OVDTA00
VER 0062 000000       DS1LSTAR TTR
VER 0065 E5A2         DS1TRBAL
VER 0069 81           DS1EXT1 EXTENT TYPE
VER 006A 00           DS1EXT1 EXTENT SEQUENCE NUMBER
VER 006B 00020000     DS1EXT1 LOWER LIMIT CCHH
VER 006F 1112000E     DS1EXT1 UPPER LIMIT CCHH
REP 0062 FFFE30
REP 0065 0000
CCHHR 0000000107      OVDTA01
VER 0062 000000       DS1LSTAR TTR
VER 0065 E5A2         DS1TRBAL
VER 0069 81           DS1EXT1 EXTENT TYPE
VER 006A 00           DS1EXT1 EXTENT SEQUENCE NUMBER
VER 006B 11130000     DS1EXT1 LOWER LIMIT CCHH
VER 006F 2223000E     DS1EXT1 UPPER LIMIT CCHH
REP 0062 FFFE30
REP 0065 0000
CCHHR 0000000108      OVDTA02
VER 0062 000000       DS1LSTAR TTR
VER 0065 E5A2         DS1TRBAL
VER 0069 81           DS1EXT1 EXTENT TYPE
VER 006A 00           DS1EXT1 EXTENT SEQUENCE NUMBER
VER 006B 22240000     DS1EXT1 LOWER LIMIT CCHH
VER 006F 2720000E     DS1EXT1 UPPER LIMIT CCHH
REP 0062 4AD230
REP 0065 0000
```

```
   ABSDUMPT 0000000101 000000010A
//*
```

scsisa.jcl

The following JCL must be run for SMS 3390 Mod 3 (SCSIA) or 3390 Mod 9 (SCSIB) volumes. This job writes the VTOC and VTOCIX, and initialized the volume for OS/390 use. See the comments contained in the JCL file for detailed instructions.

```
//jobcard JOB ‘accounting info’
//*
//* FORMATS MVS SMS SCSIA (3390-3) OR SCSIB (3390-9)
//* VOLUME WITH VTOC AND VTOC INDEX
//*
//* ==> CHANGE UNITADDRESS AND VOLID TO APPROPRIATE VALUES
//*
//STEP01   EXEC PGM=ICKDSF,REGION=4M,PARM=’NOREPLYU’
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 INIT UNITADDRESS(unitaddress) -
      VOLID(volid) VERIFY(*NONE*) -
      VTOC(0,1,1) -
      INDEX(0,2,1) -
      STORAGEGROUP -
      NOMAP
/*
```

scscis3b.jcl

The following JCL must be run for SMS 3390 Mod 3 (SCSIA) volumes. This job allocates the three data sets that fill the volume for OS/390 use; the VVDS, OVDS control areas and OVDTA data set that encapsulates the SCSI data. See the comments contained in the JCL file for detailed instructions.

```
//jobcard  JOB 'accounting info'
//*
//* ALLOCATES DATA SETS, ZAPS VTOC AND UPDATES VTOC INDEX
//* ON SMS SCSIA (3390-3) VOLUMES
//*
//BLDMOD3S PROC
//*
//* CREATE VVDS
//*
//ALOC01   EXEC PGM=IEFBR14
//TEMPDS                                                      DD
DSN=&HLQ.TEMPDS.V&volser.DATASET,DISP=(,DELETE,DELETE),
//            UNIT=3390,VOL=SER=&VOLSER,SPACE=(CYL,1,,CONTIG)
//*
//* ALLOCATE DATA SETS
//*
//ALOC02   EXEC PGM=IEFBR14,COND=(0,NE)
//OVDS     DD DSN=&HLQ..OVDS.V&VOLSER,DISP=(,CATLG,CATLG),
//            UNIT=3390,VOL=SER=&VOLSER,
//            SPACE=(TRK,1,,CONTIG),EXPDT=99365,
//            DCB=(DSORG=PS,RECFM=FBS,LRECL=512,BLKSIZE=27648)
//OVDTA00  DD DSN=&HLQ..OVDTA.V&VOLSER..P00,DISP=(,CATLG,CATLG),
//            UNIT=3390,VOL=SER=&VOLSER,
//            SPACE=(TRK,1,,CONTIG),EXPDT=99365,
//            DCB=(DSORG=PS,RECFM=F,LRECL=512,BLKSIZE=512)
//*
//* ZAP VTOC
//*
//ZAP01    EXEC PGM=AMASPZAP,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
```

```
//SYSLIB   DD DSN=FORMAT4.DSCB,DISP=OLD,
//            UNIT=3390,VOL=SER=&VOLSER,DCB=(KEYLEN=44)
//SYSIN    DD DDNAME=ZAPSYSIN
//*
//* UPDATE VTOC INDEX
//*
//INDX01   EXEC PGM=ICKDSF,COND=(0,NE),PARM='NOREPLY'
//SYSPRINT DD SYSOUT=*
//DD       DD DSN=SYS1.VTOCIX.&VOLSER,DISP=SHR,
//            UNIT=3390,VOL=SER=&VOLSER
//SYSIN    DD DDNAME=DSFSYSIN
//*
//         PEND
//* ==> CHANGE FOLLOWING EXEC STATEMENT TO APPROPRIATE VALUES
//* ==>    HLQ    - DATA SET NAME HIGH-LEVEL QUALIFIER
//* ==>    VOLSER - VOLUME SERIAL NUMBER
//* ==>
//* ==> NOTES:
//* ==> - USERID MUST HAVE CREATE AUTHORITY FOR THE HLQ
//* ==> - VOLSER MUST BE MOUNTED AND ONLINE
//*
//  EXEC PROC=BLDMOD3S,HLQ=hlq,VOLSER=volser
//*
//ZAP01.ZAPSYSIN DD *
   CCHHR 0000000102      FMT5
   VER 0000 05050505
   VER 0004 0000000000
   CCHHR 0000000104      OVDS
   VER 0062 000000       DS1LSTAR TTR
   VER 0065 E5A2         DS1TRBAL
   VER 0069 01           DS1EXT1 EXTENT TYPE
   VER 006A 00           DS1EXT1 EXTENT SEQUENCE NUMBER
   VER 006B 0000000D     DS1EXT1 LOWER LIMIT CCHH
   VER 006F 0000000D     DS1EXT1 UPPER LIMIT CCHH
   REP 0062 001002
```

```
   REP 0065 0000
   REP 006F 0001000E
   CCHHR 0000000106      OVDTA.P00
   VER 005E 88           SECONDARY ALLOCATION FLAG BYTE
   VER 0062 000000       DS1LSTAR TTR
   VER 0065 E5A2         DS1TRBAL
   VER 0069 01           DS1EXT1 EXTENT TYPE
   VER 006A 00           DS1EXT1 EXTENT SEQUENCE NUMBER
   VER 006B 0000000E     DS1EXT1 LOWER LIMIT CCHH
   VER 006F 0000000E     DS1EXT1 UPPER LIMIT CCHH
   REP 005E C8
   REP 0062 C38630
   REP 0065 0000
   REP 0069 81
   REP 006B 00020000
   REP 006F 0D0A000E
   ABSDUMPT 0000000101 000000010A
//*
//INDX01.DSFSYSIN DD *
  BUILDIX DDNAME(DD) OSVTOC
  BUILDIX DDNAME(DD) IXVTOC
//*
```

scsis9b.jcl

The following JCL must be run for SMS 3390 Mod 9 (SCSIB) volumes. This job allocates the five data sets that fill the volume for OS/390 use; the VVDS, OVDS control areas and three OVDTA data sets that encapsulates the SCSI data. See the comments contained in the JCL file for detailed instructions.

```
//jobcard  JOB 'accounting info'
//*
//* ALLOCATES DATA SETS, ZAPS VTOC AND UPDATES VTOC INDEX
//* ON SMS SCSIB (3390-9) VOLUMES
//*
//BLDMOD9S PROC
//*
//* CREATE VVDS
```

```
//*
//ALOC01   EXEC PGM=IEFBR14
//TEMPDS   DD DSN=&HLQ.TEMPDS.V&volser,DISP=(,DELETE,DELETE),
//            UNIT=3390,VOL=SER=&VOLSER,SPACE=(CYL,1,,CONTIG)
//*
//* ALLOCATE DATA SETS
//*
//ALOC02   EXEC PGM=IEFBR14,COND=(0,NE)
//OVDS     DD DSN=&HLQ..OVDS.V&VOLSER,DISP=(,CATLG,CATLG),
//            UNIT=3390,VOL=SER=&VOLSER,
//            SPACE=(TRK,1,,CONTIG),EXPDT=99365,
//            DCB=(DSORG=PS,RECFM=FBS,LRECL=512,BLKSIZE=27648)
//OVDTA00  DD DSN=&HLQ..OVDTA.V&VOLSER..P00,DISP=(,CATLG,CATLG),
//            UNIT=3390,VOL=SER=&VOLSER,
//            SPACE=(TRK,1,,CONTIG),EXPDT=99365,
//            DCB=(DSORG=PS,RECFM=F,LRECL=512,BLKSIZE=512)
//OVDTA01  DD DSN=&HLQ..OVDTA.V&VOLSER..P01,DISP=(,CATLG,CATLG),
//            UNIT=3390,VOL=SER=&VOLSER,
//            SPACE=(TRK,1,,CONTIG),EXPDT=99365,
//            DCB=(DSORG=PS,RECFM=F,LRECL=512,BLKSIZE=512)
//OVDTA02  DD DSN=&HLQ..OVDTA.V&VOLSER..P02,DISP=(,CATLG,CATLG),
//            UNIT=3390,VOL=SER=&VOLSER,
//            SPACE=(TRK,1,,CONTIG),EXPDT=99365,
//            DCB=(DSORG=PS,RECFM=F,LRECL=512,BLKSIZE=512)
//*
//* ZAP VTOC
//*
//ZAP01    EXEC PGM=AMASPZAP,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//SYSLIB   DD DSN=FORMAT4.DSCB,DISP=OLD,
//            UNIT=3390,VOL=SER=&VOLSER,DCB=(KEYLEN=44)
//SYSIN    DD DDNAME=ZAPSYSIN
//*
//* UPDATE VTOC INDEX
//*
```

```
//INDX01   EXEC PGM=ICKDSF,COND=(0,NE),PARM='NOREPLY'
//SYSPRINT DD SYSOUT=*
//DD       DD DSN=SYS1.VTOCIX.&VOLSER,DISP=SHR,
//            UNIT=3390,VOL=SER=&VOLSER
//SYSIN    DD DDNAME=DSFSYSIN
//*
//         PEND
//*
//* ==> CHANGE FOLLOWING EXEC STATEMENT TO APPROPRIATE VALUES
//* ==>    HLQ    - DATA SET NAME HIGH-LEVEL QUALIFIER
//* ==>    VOLSER - VOLUME SERIAL NUMBER
//* ==>
//* ==> NOTES:
//* ==> - USERID MUST HAVE CREATE AUTHORITY FOR THE HLQ
//* ==> - VOLSER MUST BE MOUNTED AND ONLINE
//*
//  EXEC PROC=BLDMOD9S,HLQ=hlq,VOLSER=volser
//*
//*
//ZAP01.ZAPSYSIN DD *
   CCHHR 0000000102      FMT5
   VER 0000 05050505
   VER 0004 0000000000
   CCHHR 0000000104      OVDS
   VER 0062 000000       DS1LSTAR TTR
   VER 0065 E5A2         DS1TRBAL
   VER 0069 01           DS1EXT1 EXTENT TYPE
   VER 006A 00           DS1EXT1 EXTENT SEQUENCE NUMBER
   VER 006B 0000000D     DS1EXT1 LOWER LIMIT CCHH
   VER 006F 0000000D     DS1EXT1 UPPER LIMIT CCHH
   REP 0062 001002
   REP 0065 0000
   REP 006F 0001000E
   CCHHR 0000000106      OVDTA.P00
   VER 005E 88           SECONDARY ALLOCATION FLAG BYTE
```

```
VER 0062 000000       DS1LSTAR TTR
VER 0065 E5A2         DS1TRBAL
VER 0069 01           DS1EXT1 EXTENT TYPE
VER 006A 00           DS1EXT1 EXTENT SEQUENCE NUMBER
VER 006B 0000000E     DS1EXT1 LOWER LIMIT CCHH
VER 006F 0000000E     DS1EXT1 UPPER LIMIT CCHH
REP 005E C8
REP 0062 FFFE30
REP 0065 0000
REP 0069 81
REP 006B 00020000
REP 006F 1112000E
CCHHR 0000000107      OVDTA.P01
VER 005E 88           SECONDARY ALLOCATION FLAG BYTE
VER 0062 000000       DS1LSTAR TTR
VER 0065 E5A2         DS1TRBAL
VER 0069 01           DS1EXT1 EXTENT TYPE
VER 006A 00           DS1EXT1 EXTENT SEQUENCE NUMBER
VER 006B 00010000     DS1EXT1 LOWER LIMIT CCHH
VER 006F 00010000     DS1EXT1 UPPER LIMIT CCHH
REP 005E C8
REP 0062 FFFE30
REP 0065 0000
REP 0069 81
REP 006B 11130000
REP 006F 2223000E
CCHHR 0000000108      OVDTA.P02
VER 005E 88           SECONDARY ALLOCATION FLAG BYTE
VER 0062 000000       DS1LSTAR TTR
VER 0065 E5A2         DS1TRBAL
VER 0069 01           DS1EXT1 EXTENT TYPE
VER 006A 00           DS1EXT1 EXTENT SEQUENCE NUMBER
VER 006B 00010001     DS1EXT1 LOWER LIMIT CCHH
VER 006F 00010001     DS1EXT1 UPPER LIMIT CCHH
REP 005E C8
```

```
   REP 0062 4AD230
   REP 0065 0000
   REP 0069 81
   REP 006B 22240000
   REP 006F 2720000E
   ABSDUMPT 0000000101 000000010A
//*
//INDX01.DSFSYSIN DD *
  BUILDIX DDNAME(DD) OSVTOC
  BUILDIX DDNAME(DD) IXVTOC
//*
```

This page intentionally left blank

READER'S COMMENT FORM

Manual Name: ______________________________ Manual PN: __________

Please check or fill in the items; adding explanations/comments in the space provided.

Which of the following terms best describes your job?

☐ Field Engineer	☐ Manager	☐ Programmer	☐ Systems Analyst
☐ Engineer	☐ Mathematician	☐ Sales Representative	☐ Systems Engineer
☐ Instructor	☐ Operator	☐ Student/Trainee	☐ Other (explain below)

How did you use this publication?

☐ Introductory text	☐ Reference manual	☐ Student/Trainee	☐ Instructor text

☐ Other (explain) ______________________________

Did you find the material easy to read and understand? ☐ Yes ☐ No (explain below)

Did you find the material organized for convenient use? ☐ Yes ☐ No (explain below)

Specific criticisms (explain below):

Clarifications on pages ______________________________

Additions on pages ______________________________

Deletions of pages ______________________________

Errors on pages ______________________________

Explanations and other comments:

Note: *Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form. If you would like a reply, please supply your name and address on the reverse side of this form.*

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A.

TO COMPLY WITH POSTAL REGULATIONS, FOLD EXACTLY ON DOTTED LINES AND TAPE (DO NOT STAPLE)

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 2 LOUISVILLE, CO U.S.A.

POSTAGE WILL BE PAID BY ADDRESSEE

STORAGE TECHNOLOGY CORPORATION
ENTERPRISE DISK INFORMATION DEVELOPMENT
ONE STORAGETEK DRIVE
LOUISVILLE CO 80028-2201

FOLD HERE AND TAPE DO NOT STAPLE FOLD HERE AND TAPE

If you would like a reply, please print:

Your Name: ______________________________

Company Name: ______________ Department: ______________

Street Address: ______________________________

____

Cut or Fold
Along Line