Sun™ StorEdge™ A7000 DataShare Facility System Administrator's and User's Guide

THE NETWORK IS THE COMPUTER™

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900 USA
650 960-1300 Fax 650 969-9131

Part No.: 805-4366-11
December 1998, Revision A

Send comments about this document to: docfeedback@sun.com

Contents

Preface xv

What is DataShare Facility? xv

How This Book Is Organized xvi

General Information xvi

Administration xvi

Applications xvii

Additional Information xvii

Using UNIX Commands xvii

Documentation Conventions xviii

Shell Prompts xix

Related Documentation xix

Sun Documentation on the Web xx

Sun Welcomes Your Comments xx

1. Product Description 1-1

Introduction 1-2

Year 2000 Compliance 1-3

Mainframe DataShare Capability 1-3

DASD Tools Alternate Access to Mainframe Volumes 1-5

Open Systems DataShare Capability 1-6

Bidirectional Information Exchange 1-8

2. DataShare Facility and the StorEdge A7000 Architecture 2-1

Introduction 2-2

DataShare Facility Architecture 2-3

3. Planning and Configuration 3-1

DataShare Facility Considerations 3-2

Virtual Disk Size 3-2

Extended Storage Managers 3-2

Failover and High Availability 3-2

Mainframe DataShare Capability 3-3

▼ To Determine the Number of MFDSF Volumes 3-3

▼ To Configure the Mainframe 3-4

▼ To Configure the A7000 3-4

▼ To Configure the Open Systems 3-5

Open Systems DataShare Capability 3-6

Storage Allocation and DataShare Facility 3-6

▼ To Determine Concurrent Transfer Areas 3-7

▼ To Configure the A7000 3-8

Open Systems Configuration 3-8

Mainframe Configuration 3-8

4. Special Considerations 4-1

Mainframe DataShare Capability (MFDSF) 4-2

Concurrent Access 4-2

Dataset Characteristics - MVS 4-2

Mainframe Software Data Compression 4-3

Multivolume Datasets 4-3

Performance 4-3

Cache Synchronization 4-4

System Managed Storage (SMS) 4-4

Volume Characteristics 4-4

VSE Limitations 4-5

Dataset Characteristics 4-5

AF=CKD 4-6

AF=RAWRO 4-6

AF=RAWRW and DataShare Open Backup 4-6

DASD Tools 4-7

Mainframe Issues 4-7

Database Issues 4-8

Open Systems DataShare Capability (OSDSF) 4-9

Additional Mainframe Considerations 4-9

Concurrent Access 4-9

Dataset Characteristics 4-9

Synchronization 4-10

Volume Characteristics 4-10

5. DataShare Facility Applications 5-1

Introduction 5-2

Eliminating Data Replication and Movement 5-2

Mainframe Usage 5-4

▼ To Use Mainframe DataShare Capability 5-4

Function Consolidation 5-5

Function Offload 5-5

Mainframe Usage 5-7

▼ To Use the Mainframe DataShare Capability 5-7

Open Systems Usage 5-7

▼ To Use the Open Systems Capability 5-7

6. Best Business Practices 6-1

Example — Mainframe DataShare Capability (MFDSF) with MVS 6-2

Mainframe Environment 6-2

Open System Environment 6-2

▼ To Plan Replacement of Current Method 6-2

▼ To Configure the A7000 6-3

▼ To Configure the Open Systems 6-4

▼ To Use the MFDSF Volumes 6-4

On the Sun Ultra Sparc 6-4

On the mainframe 6-4

On the Sun Ultra Sparc 6-5

7. Open System Platform-Specific Issues 7-1

HP 9000 / HP_UX 7-2

Intel / Windows NT 7-2

RS6000 / AIX 7-3

Sun / Solaris 7-3

A. Sample `dsf.cf` File A-1

Example of a `dsf.cf` File A-2

B. Calculating Data Capacity B-1

Mainframe DataShare Capability B-2

Open Systems DataShare Capability B-3

C. DASD Tools UNIX Man Pages C-1

Man Pages C-1

D. DASD Tools Windows NT Man Pages D-1

Man Pages D-1

Glossary GL-1

Index IN-1

Code Samples

CODE EXAMPLE 6-1 `DB_load.sh` - PSEUDO scripting 6-5

CODE EXAMPLE A-1 Sample `dsf.cf` File A-2

Figures

FIGURE 1-1 Mainframe DataShare Capability 1-4

FIGURE 1-2 Open Systems DataShare Capability 1-7

FIGURE 1-3 Bidirectional Data Exchange 1-8

FIGURE 2-1 Product Architecture 2-2

FIGURE 2-2 DataShare Facility Architecture 2-4

FIGURE 2-3 Mainframe DataShare Capability Architecture 2-5

FIGURE 2-4 Open System DataShare Capability Architecture 2-6

FIGURE 3-1 3390 Track Characteristics 3-6

FIGURE 5-1 Traditional Data Replication 5-3

FIGURE 5-2 Eliminating Data Replication with DataShare 5-3

FIGURE 5-3 Function Offload Without and With DataShare Facility 5-6

Tables

TABLE P-1 Documentation Conventions xviii

TABLE P-2 Shell Prompts xix

TABLE P-3 Related Documentation xix

TABLE B-1 Actual Data Capacity (Mainframe DataShare Capability) B-2

TABLE B-2 Actual Data Capacity (Open Systems DataShare Capability) B-3

Preface

The *DataShare Facility System Administrator's & User's Guide* describes the Sun™ StorEdge™ A7000 DataShare™ product and provides configuration and usage information. These instructions are designed for an experienced system administrator and application users.

What is DataShare Facility?

The DataShare Facility (DSF) enables data stored on the StorEdge™ A7000 to be shared easily and effectively among IBM-compatible mainframes and open system platforms.

With DSF, hosts can access non-native information using existing native access mechanisms (existing read commands and software interfaces) requiring no software additions to the mainframe or open systems host.

The DataShare Facility eliminates inefficient and costly bulk data movements between platforms through its ability to provide common shared information access. In addition, this shared access capability can be used by multiple platforms simultaneously, easing and sometimes eliminating the logistic problems associated with creating multiple copies of data in separate processing environments.

DataShare Facility comprises the following parts:

- Mainframe DataShare capability.
- Open systems DataShare capability.
- DASD Tools as an alternative way for open system hosts to access mainframe volumes.

How This Book Is Organized

This document supplies information to administrators and application users in a logical manner and is divided into sections specific to these areas.

The audience for this manual consists of the following groups:

- Mainframe administrators (system programmers)
- Open system administrators
- Mainframe users
- Open system users

In areas of this document where it is applicable, the audience needing the specific information is indicated.

General Information

These chapters contain DataShare Facility information for Mainframe and Open System administrators and users.

Chapter 1 describes the DSF capabilities.

Chapter 2 describes how DSF fits into the StorEdge A7000 architecture.

Administration

These chapters are for those responsible for installing, configuring, and maintaining the DataShare Facility on the mainframe host, open system host, and StorEdge A7000.

Chapter 3 contains planning and configuration information.

Chapter 4 describes considerations for system administrators and users to keep in mind when using DSF.

Applications

These chapters are for DataShare Facility users.

Chapter 5 describes how to use DSF.

Chapter 6 contains application examples for specific configurations.

Chapter 7 describes issues that you must be aware of when using specific open system platforms.

Additional Information

The following appear at the end of the manual and are useful to administrators and users.

Appendix A contains a sample dsf.cf file.

Appendix B contains information to help calculate data capacity.

Appendix C contains hardcopies of the DASD Tools UNIX man pages.

Appendix D contains hardcopies of the DASD Tools Windows NT man pages.

Glossary includes definitions of DSF, mainframe, and open system terms.

Index

Using UNIX Commands

This document may not contain information on basic UNIX® commands and procedures such as shutting down the system, booting the system, and configuring devices.

Refer to your UNIX documentation for this information.

Documentation Conventions

TABLE P-1 Documentation Conventions

Typeface or Symbol	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output.	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail`.
`AaBbCc123`	What you type, when contrasted with on-screen computer output.	`%` **`su`** `Password:`
AaBbCc123	Book titles, new words or terms, words to be emphasized. Command-line variable; replace with a real name or value.	Read Chapter 6 in the *User's Guide*. These are called *class* options. You *must* be `root` to do this. To delete a file, type `rm` *filename*.
[ ]	In syntax, brackets indicate that an argument is optional.	In this example, `/etc/dsfadmin` is the only part of the syntax that *must* be entered. `/etc/dsfadmin [ -v [` *device*]
`com(n)`	The form `command(number)`, where the number in parentheses ranges from 1 through 6 and is followed by letters, indicates the presence of an online reference man page.	`dsfadmin(1DSF)`

Shell Prompts

TABLE P-2 Shell Prompts

Shell	Prompt
C shell	*machine_name*%
C shell superuser	*machine_name*#
Bourne shell and Korn shell	$
Bourne shell and Korn shell superuser	#

Related Documentation

TABLE P-3 Related Documentation

Application	Title	Part Number
Release	*DataShare Facility Release Notes*	online
Reference	*Direct Access Storage Device (DASD) Manager User's Guide*	805-4884-10
Reference	*SCSI Target Emulation Release Notes*	online
Reference	*Simulation of Count-Key-Data (SIMCKD) Release Notes*	online

Sun Documentation on the Web

The `docs.sun.com`[SM] web site enables you to access Sun technical documentation on the Web. You can browse the `docs.sun.com` archive or search for a specific book title or subject at:

`http://docs.sun.com`

Sun Welcomes Your Comments

We are interested in improving our documentation and welcome your comments and suggestions. You can email your comments to us at:

`docfeedback@sun.com`

Please include the part number of your document in the subject line of your email.

CHAPTER **1**

Product Description

This chapter describes DataShare Facility and its capabilities and contains the following sections:

- Introduction—page 1-2
- Mainframe DataShare Capability—page 1-3
- Open Systems DataShare Capability—page 1-6

Introduction

The DataShare Facility (DSF) enables data stored on the StorEdge A7000 to be shared easily and effectively among IBM-compatible mainframes and open system platforms. DataShare provides an effective alternative to many of the inefficient information transfer and access methods that are common in today's heterogeneous computing environments.

Efficient cross-platform data sharing is achieved between channel-attached mainframes and SCSI-attached open system hosts. With DataShare Facility, hosts can access non-native information using existing native access mechanisms, requiring no software additions to the mainframe or open system host. Using A7000 configuration, the originating host grants availability of specific volumes to specific other hosts. Restricting user access to these shared volumes on the specific host is managed through that host's software.

Through DataShare, information transfer between platforms can be eliminated by providing direct information access to the platform requiring it. Further, through direct information access, processes that are currently being duplicated on many platforms can be consolidated into the environment where the function is accomplished most efficiently and effectively.

The following DSF primitives are used in different combination to provide DSF capabilities:

- Access functions (AF) interpret volume structures
- Presentation functions (PF) specify the format to be presented (for example, ISO-9660)
- Conversion functions (CF) modify the information format

DSF comprises the following parts:

- Mainframe DataShare capability (MFDSF) to read mainframe volumes and present fixed block datasets in a native format to open systems hosts.
- Open systems DataShare capability (OSDSF) to read open system raw disks and present them as datasets to the mainframe.
- DASD Tools as an alternative way for open system hosts to access mainframe volumes.

Year 2000 Compliance

DataShare (a) will not produce errors in date data related to the year change from December 31, 1999 to January 1, 2000 and (b) will handle leap years correctly.

The two digit year field in a mainframe file date is interpreted as follows before the year information is placed in the ISO-9660 image:

- Based on the year 2000 if the number is in the range of 00 to 68
- Based on the year 1900 if the number is in the range of 69 to 99

Mainframe DataShare Capability

The Mainframe DataShare capability provides tools that allow existing mainframe volumes (3380, 3390-1, or 3390-2) residing on an A7000 to be read directly by SCSI-attached open systems platforms.

Qualified SCSI-attached open system hosts have read-access to Multiple Virtual Storage (MVS) and Virtual Storage Extended (VSE) physical-sequential, fixed-record/blocked (QSAM) datasets through the SCSI target connection. In this case, the specified mainframe volumes appear as ISO-9660 (CD-ROM) devices to the open system host. Datasets within the target mainframe volume appear as files within the ISO-9660 file system, allowing any native open system command, GUI, or programmatic interface capable of accessing ISO-9660 volumes and files to access mainframe information directly as though it were native, local data. Support is provided for System Managed Storage (SMS) and multivolume datasets. When using this DataShare capability, the mainframe writes the dataset(s) to one or more mainframe volumes, and then the open system host accesses these datasets. Do not write to these mainframe volumes while the open system host is accessing them.

Multiple open systems can access the same mainframe volume concurrently. Primitives for application synchronization between mainframes and the open system(s) accessing this information are not currently supplied within the DataShare Facility environment and must be coordinated by the participating systems/applications, if required.

By default, data in a physical sequential dataset appears to the open system host exactly as it was written by the mainframe system (for example, EBCDIC). As in homogenous information sharing, applications (mainframe writer/open system reader) must cooperate on structure and format of the information being exchanged.

The following figure illustrates the mainframe DataShare capability.

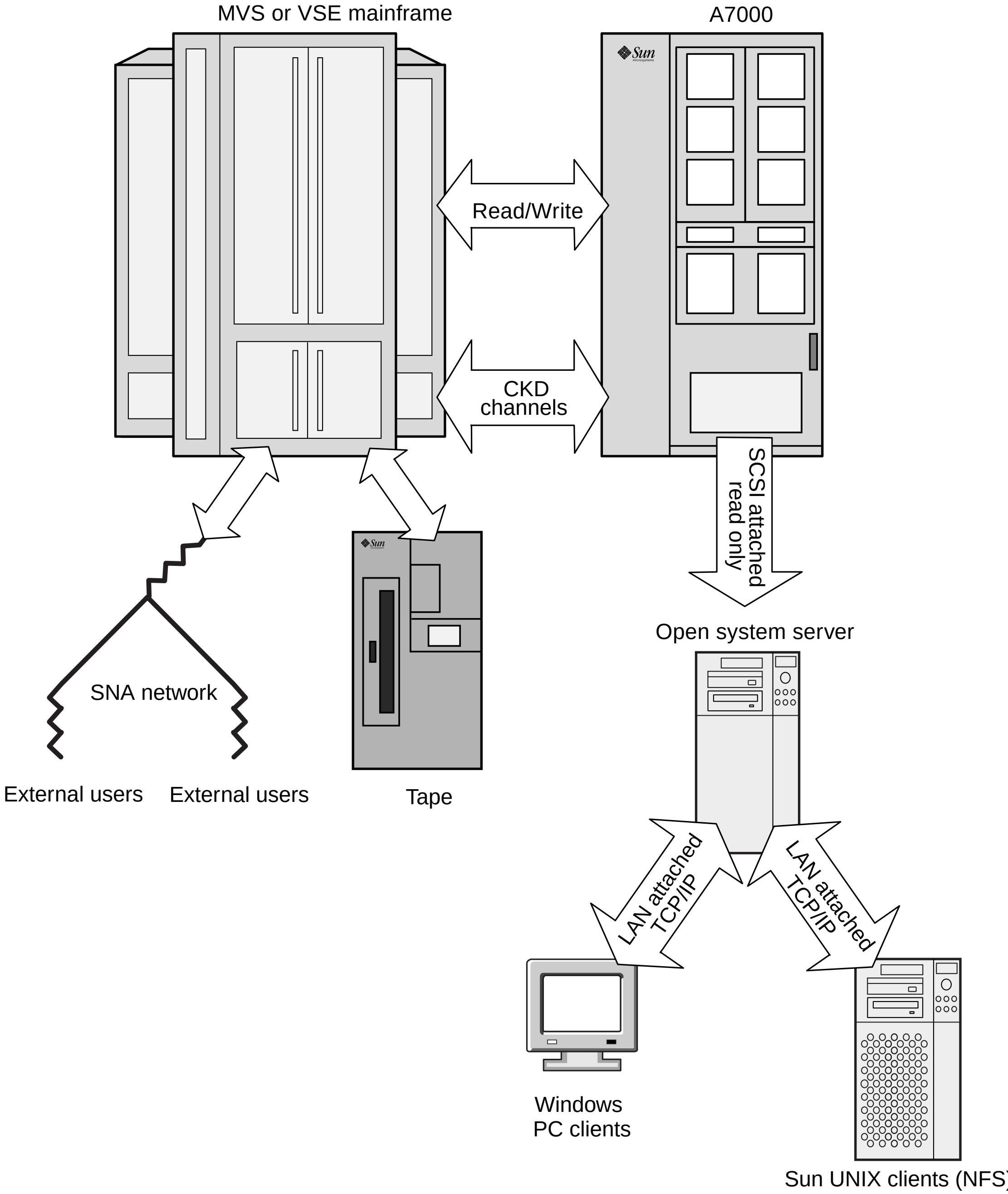

FIGURE 1-1 Mainframe DataShare Capability

In some limited applications, it is useful to convert the dataset structure externally from the source and destination systems. The DataShare environment provides conversion functions for this purpose. These functions perform common conversions transparently as data is accessed by the open system host. The same mainframe volume can appear as multiple ISO-9660 volumes to an open system, each with or without a specific conversion being applied to the same input data. These functions are configured on a volume, not dataset, basis. Character conversion functions are a one-to-one character mapping using either the default EBCDIC to ASCII mapping or custom mapping. Additionally, record delimiter inclusion (for example, fixed record to carriage return/line feed) is provided.

These conversion routines are for convenience. Conversion functions are often faster on the mainframe or on the open system rather than on the A7000 processor.

DASD Tools Alternate Access to Mainframe Volumes

The DASD Tools enables the open system host to access a mainframe volume providing a stream of data to an application.

Note – These volumes cannot be accessed using open system native access methods.

The qualified attached open system hosts have read access to MVS physical-sequential fixed-block and variable block QSAM, PDS, and VSAM datasets or VSE physical-sequential fixed-block and variable block QSAM datasets through SCSI Target Emulation (STE).

The tools provide the following:

- Directory/VTOC listings of the mainframe volume
- Dataset/volume characteristics
- Sequential read capability

DASD Tools contains the following commands.

When using VSE, the output for some of these commands may be degraded because some information is not supplied in the VTOC.

- `dasd_cat`
- `dasd_db_init` (no Windows NT support)
- `dasd_file`
- `dasd_ls`
- `dasd_pds` (no VSE support)
- `dasd_retrieve` (no Windows NT support)
- `dasd_vol`
- `sms_cat` (no Windows NT support)
- `vsam_alt_cat` (no VSE support)
- `vsam_cat` (no VSE support)

See Appendix C for the DASD Tools UNIX man pages and Appendix D for the Windows NT man pages.

Open Systems DataShare Capability

The open systems DataShare capability enables resident open system disk images on the A7000 to be read directly by mainframes.

These disk images appear to the MVS and VSE mainframes as though the entire logical disk is a read-only physical-sequential fixed-record/blocked (QSAM) dataset within a single 3390-3 or 3380-K volume. The logical record size, block size, VOLSER, and dataset name representing this disk area are preconfigured on the A7000 processor at initialization.

The size of the open system logical disk, which appears as a dataset to the mainframe, is limited by the following:

Limit	Description
minimum	Minimum logical disk supported by the open system.
maximum	Unlimited. If this size is greater than the maximum dataset size representable on one 3390-3 or 3380-K at mainframe blocksize, the open system logical disk can be presented as multiple mainframe volumes.

The open system host writes the information, which requires mainframe access, directly to the pertinent raw disk area on the A7000 processor, ignoring open system file system semantics. UNIX systems provide this *raw* interface as a standard I/O semantic. Other operating systems have varying degrees of support for these raw I/O primitives.

By default, data within a logical disk area appears exactly as it was written by the open system (for example, ASCII). The open systems and mainframe applications must cooperate on coordination, structure, and format of the information being exchanged. As with the mainframe DataShare capability, optional one-to-one character conversion is supported from the default ASCII to EBCDIC map or custom map.

Because there is no portable, writeable file system support on open systems at this time, no open system file level access is supported by this product. The open systems DataShare capability is best suited to large information transfer applications.

The following figure illustrates the open systems DataShare capability.

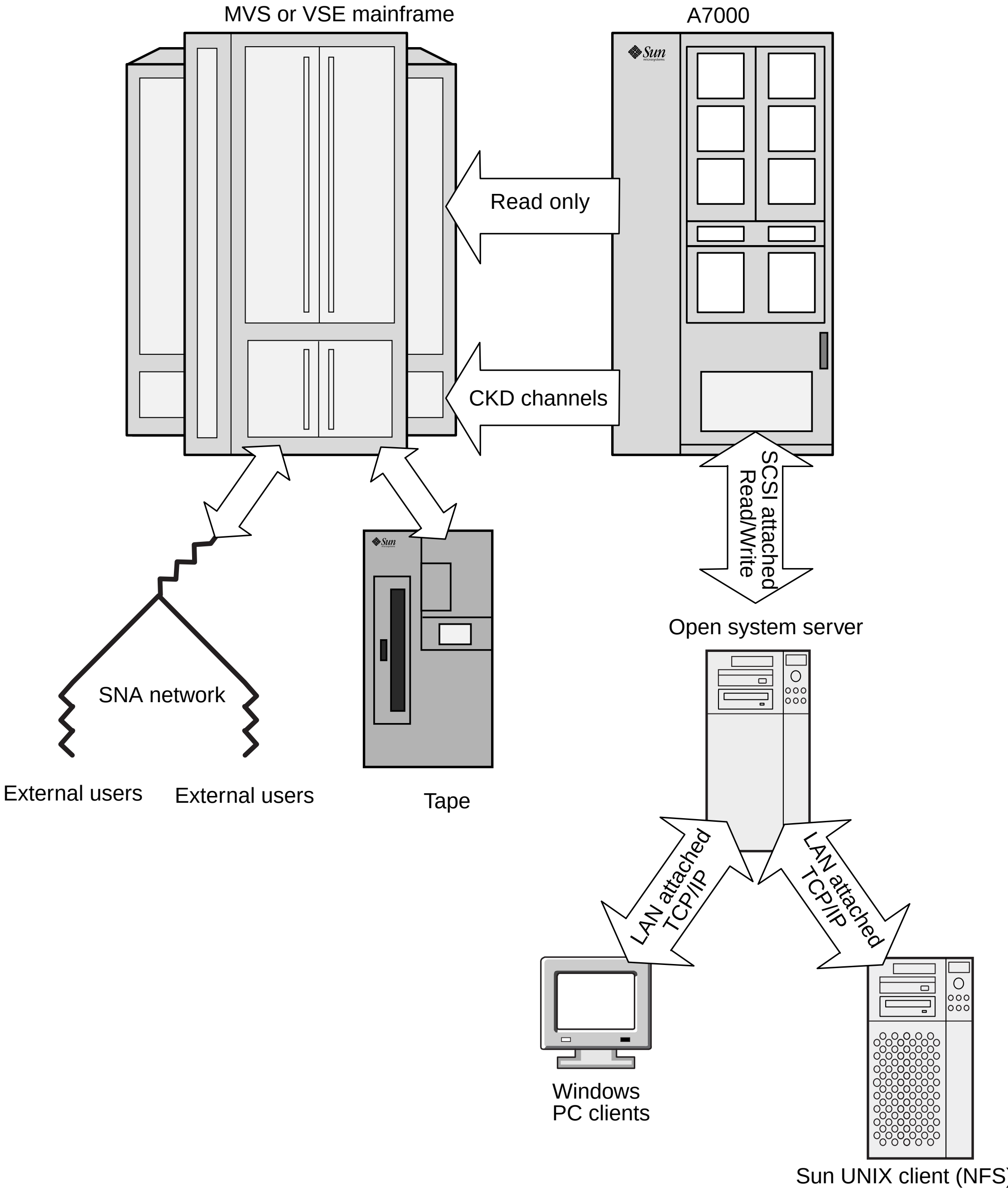

FIGURE 1-2 Open Systems DataShare Capability

Bidirectional Information Exchange

Using mainframe and open systems DataShare capabilities, information can be shared in a secure environment allowing bidirectional information exchange (eliminating the need for data transfer), and providing the foundation for very powerful application solutions.

The following figure illustrates bidirectional information exchange.

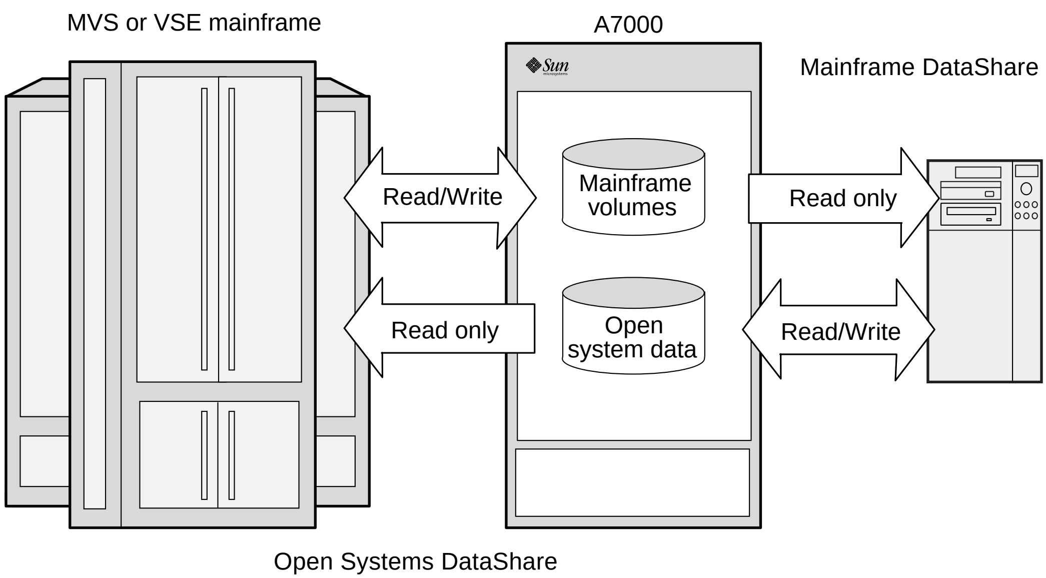

FIGURE 1-3 Bidirectional Data Exchange

CHAPTER **2**

DataShare Facility and the StorEdge A7000 Architecture

This chapter uses text and figures to describe the DataShare Facility architecture and how it is integrated into the StorEdge A7000 architecture.

- Introduction—page 2-2
- DataShare Facility Architecture—page 2-3

Introduction

The StorEdge A7000 architecture includes the Simulation of Count-Key-Data (SIMCKD) and SCSI Target Emulation (STE) facilities, which allow the simultaneous emulation of Count-Key-Data (CKD) mainframe volumes using BMC and ESCON Channels, and Fixed Block Architecture (FBA) open systems volumes across SCSI II Channels. SIMCKD and STE share a common cache and a head / disk assembly (HDA) pool within the A7000. For a specified volume, information is physically stored on the HDAs and staged into cache in a format that is optimized for the native emulation being provided.

The following figure illustrates the product architecture and the relationship between the cache, HDA pool, and SIMCKD and STE emulation facilities.

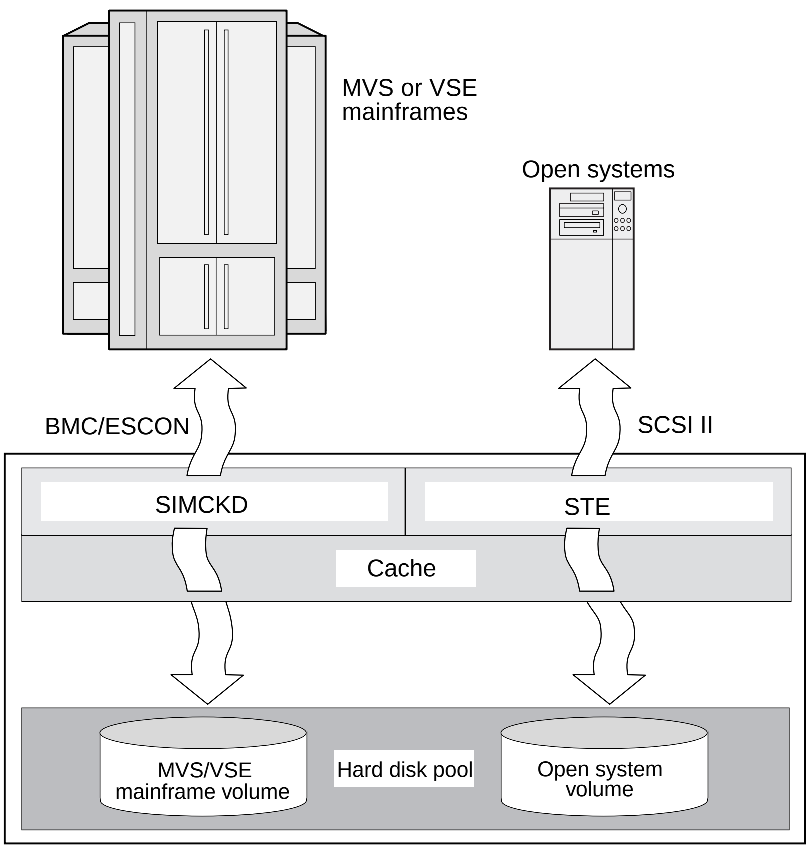

FIGURE 2-1 Product Architecture

DataShare Facility Architecture

The DataShare Facility (DSF) is an integrated function within the A7000 architecture that is configured to access the HDA pool providing non-native information through the cache to an emulation different from the one that created it. The STE / CKD layer is then used to make data stored through CKD / SCSI emulation available to open system hosts.

DSF acts directly on stored data in real time as a non-native host reads information. Host platform configuration of non-native devices needing to be shared is consistent with the native emulation (for example, STE for open systems). File system appearance for non-native information is consistent with a native supported file system for the host requiring access (for example, ISO-9660 (CD-ROM)).

FIGURE 2-2 illustrates the DSF architecture.

The DataShare Facility internal architecture comprises several functions. These functions provide access to heterogeneous information using homogeneous presentation methods without additional host-based software.

The DataShare Facility primitives include:

- *Data access* functions (AF), which interpret the volume structures to be supplied to a specific presentation or conversion function. Supported access functions include:
  - 3380 / 3390 MVS / VSE volumes
  - RAW, read only open system SCSI disk volumes
- *Data presentation* functions (PF), which enable delivery of accessed information to a specific emulation in a volume format consistent with that environment. Supported presentation functions include:
  - ISO-9660 (CD-ROM) volumes
  - CKD 3380-K or 3990-3 MVS / VSE volumes
- *Data conversion* / processing functions (CF), which modify the information format or content prior to the presentation function.
  Several conversion / processing functions can be linked to a given access and presentation function pair providing a powerful stream of conversions. Supported conversion / processing functions include:
  - EBCDIC to ASCII
  - ASCII to EBCDIC
  - One-to-one custom map character conversion
  - Fixed record to carriage return / line feed conversion
  - Fixed record to delimited record
  - Fixed record to custom delimited record

Multiple data conversion/presentation streams can be linked to the same *data access* function, creating several logical views of the same physical information from a single system. These multiple views are accessed by the host requesting access as multiple distinct volumes/file systems. You can also configure multiple hosts to access the same presentation function, allowing simultaneous read access of the same physical information by multiple systems.

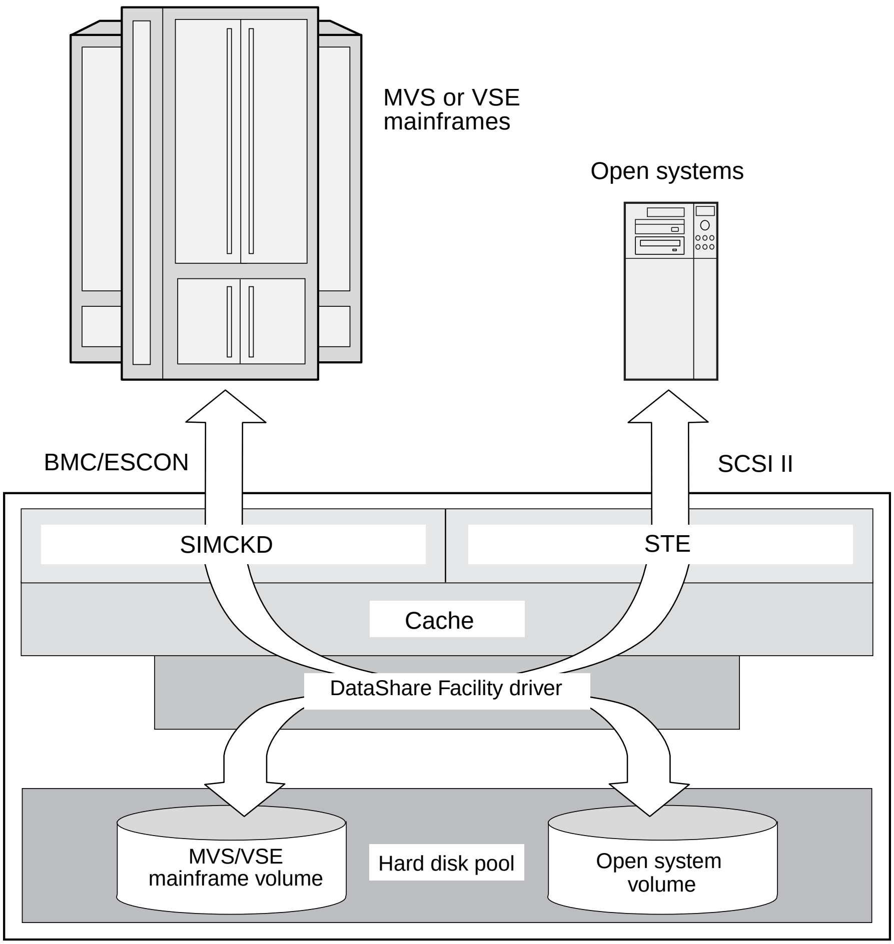

FIGURE 2-2 DataShare Facility Architecture

FIGURE 2-3 illustrates a typical mainframe DataShare capability architecture.

FIGURE 2-4 illustrates a typical open system DataShare capability architecture.

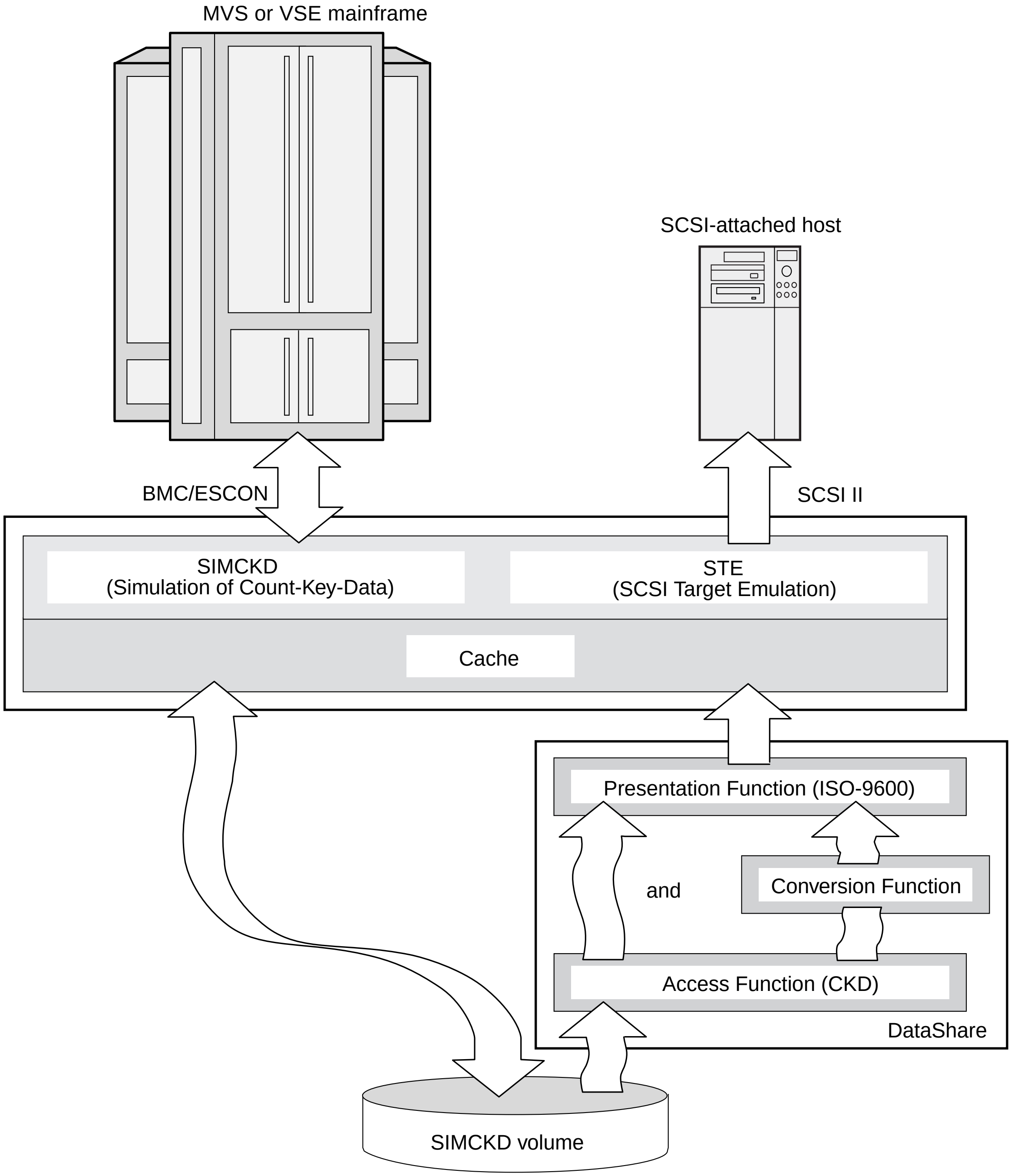

FIGURE 2-3 Mainframe DataShare Capability Architecture

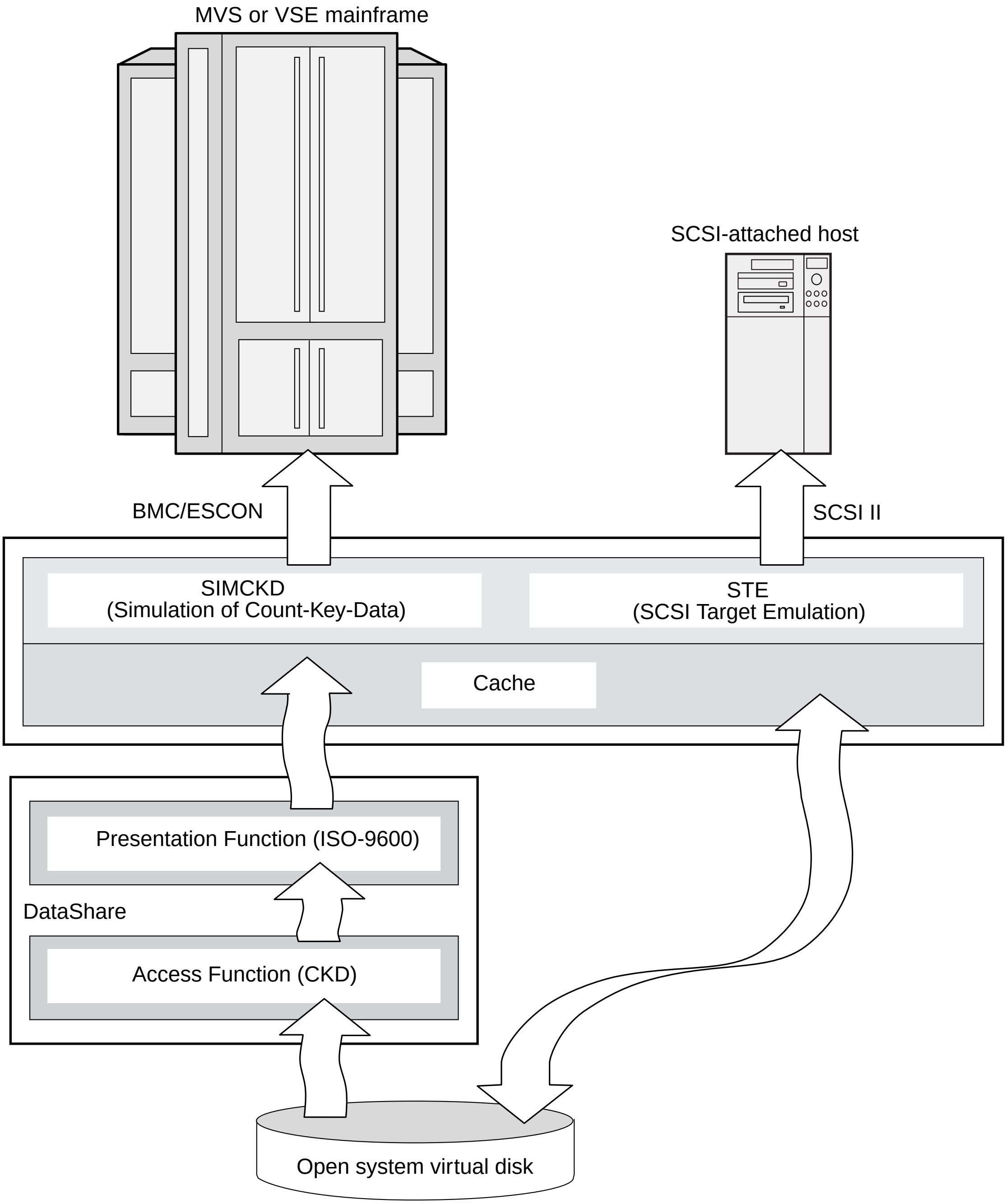

FIGURE 2-4 Open System DataShare Capability Architecture

CHAPTER **3**

Planning and Configuration

This chapter contains procedures for planning and configuration of the DataShare Facility capabilities.

- DataShare Facility Considerations—page 3-2
- Mainframe DataShare Capability—page 3-3
- Open Systems DataShare Capability—page 3-6

DataShare Facility Considerations

The following subsections contain useful information for planning and configuring the DataShare Facility. This information applies to the mainframe and open system DataShare capabilities.

Virtual Disk Size

Each open system may impose limitations on the following:

- Physical disk size
- Largest partition
- Largest file that it supports

Check your vendor documentation for details.

Extended Storage Managers

Many open system platforms have software that provides logical volume management. In this respect, an open system virtual disk residing on a StorEdge A7000 platform is like any other manufacturer's SCSI disk.

When using the DataShare Facility for open system data sharing back to the mainframe, there must be a one-to-one correspondence of the disk allocation on the open system host to the A7000 platform. That is, you must NOT use an Open System Extended Storage Manager for these A7000 virtual disks.

Failover and High Availability

Failover and high availability can be configured using a combination of A7000 and host software and hardware functionality. For example, an open system host can configure hardware failover, or use mirroring, to provide high availability.

Mainframe DataShare Capability

The following sections describe planning and setup of the mainframe DataShare capability (MFDSF) used to eliminate data movement. The mainframe is the originator/user and the open system is a read-only user.

Before configuring the mainframe or open system host, determine the number of MFDSF volumes that are required. To determine this, calculate the amount of data to be shared (written by the mainframe, read by the open system).

▼ To Determine the Number of MFDSF Volumes

Use these steps to determine the number of MFDSF volumes.

The following groups may be involved in this decision:

- Mainframe system programmers and users
- Open system administrators and users

1. **Calculate the maximum amount of data to be shared with the open system host.**
2. **Determine the mainframe volume type required (choose from any 3380, 3390-1, or 3390-2).**
3. **Use this** *rule of thumb* **for determining approximate** *data capacity* **per volume:**
   - 1.2 Gbytes if a 3380-K volume is used.
   - 1.5 Gbytes if a 3390-2 volume is used.

   Because of mainframe Count-Key-Data (CKD) architecture, actual volume *data capacity* varies depending on the user-specified blocksize and device type.

   See Appendix B for detailed data storage calculation information.
4. **Determine the number of mainframe (mf) volumes required for the shared data:**

   number of mf volumes=total mf shared data / mf volume capacity
5. **Determine the mainframe unit address and VOLSERs to be shared.**
6. **Determine SCSI Target Emulation (STE) channel usage.**

Note – A shared mainframe volume cannot hold more than 2 Gbytes of data available to the open system host because of some open system hosts' restrictions.

▼ To Configure the Mainframe

This procedure is for mainframe system programmers.

1. **Establish a unique esoteric name for the MFDSF volumes, excluding them from other common esoteric names.**

2. **Set up security options to extend write access to only specified MFDSF users or applications.**

3. **Plan to schedule applications so all mainframe write activity is completed before the open system access begins.**

4. **Ensure that the volume initialization and the VTOC size is appropriate. Use the default VTOC size of one cylinder.**

5. **Ensure that the MFDSF volume is initialized and online.**

▼ To Configure the A7000

This procedure is for the person doing the A7000 configuration.

Prerequisite: The A7000 mainframe volume is initialized and online and open system physical connections are established.

1. **Configure SCSI Target Emulation (STE) to open system hosts, create the DSF device entries, configure CKD Access/ISO-9660 presentation functions, and configure conversion function, if required.**

2. **Start the DSF, SIMCKD, and STE components.**

▼ To Configure the Open Systems

Note – An open system reboot may be required.

This procedure is for open system administrator.

1. **Configure the open system host to recognize the A7000 virtual disks and CD-ROMs.**

Note – A reboot of the open system host is often required before new SCSI devices are recognized. Refer to the open system host vendor documentation.

---

2. **Create CD-ROM volume mount points. See "Multivolume Datasets" on page 4-3 and"System Managed Storage (SMS)" on page 4-4.**

3. **Mount the CD-ROMs if necessary.**

Open Systems DataShare Capability

The following sections describe planning and configuration of the open systems DataShare capability (OSDSF) used to eliminate data movement. The source is an open system and the destination is a mainframe.

Storage Allocation and DataShare Facility

DASD Manager’s default setup of the open systems capability of the DataShare Facility (DSF) sets half-track blocking with one record per block, providing maximum capacity and good performance.

This configuration provides actual data capacity up to 2.58 Gbytes (3390-3) or 1.67 Gbytes (3380-K) of open systems data in a single mainframe volume. Alternate block and record lengths can be specified during DSF configuration to satisfy specific application requirements.

For open system, the blocksize must be a multiple of the backend disk drive geometry (512 bytes). Because of the mainframe restrictions, block size must be less than 32K bytes.

With the open systems DataShare capability, consider application requirements when setting blocksize (BLKL) and logical record length (LRECL).

The following figure illustrates a 3390 track characteristic for an open systems DSF capability dataset. There are two blocks of approximately 27K bytes each. Each block has an integer number of logical records. For example, if the record size is 512 and the blocksize is 27648, there are 54 512-byte records.

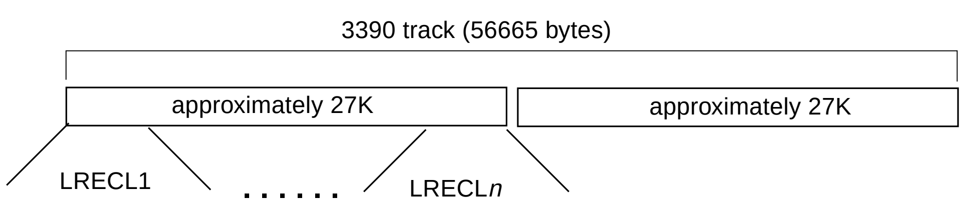

FIGURE 3-1 3390 Track Characteristics

▼ To Determine Concurrent Transfer Areas

Follow these steps to determine the number and size of concurrent transfer areas for the applications.

The following groups may be involved:

- Mainframe system programmers and users
- Open system administrators and users

1. **Calculate the maximum amount of data to be transferred.**

2. **Determine the preconfigured block and logical record size best suited for the applications.**

3. **Determine the volume type defined to the mainframe (that is, 3390-3 or 3380-K).**

4. **Use this** *rule of thumb* **for determining approximate** *data capacity* **per volume:**

   - 1.7 Gbytes if a 3380-K volume is used.
   - 2.5 Gbytes if a 3390-3 volume is used.

   Because of mainframe Count-Key-Data (CKD) architecture, actual volume *data capacity* varies depending on the user-specified blocksize.

   See Appendix B for detailed data storage calculation information.

5. **Determine the number of mainframe (mf) volumes:**

   number of mf volumes=total shared open system data / mf volume capacity

6. **Determine collaboration between open system host and mainframe applications (if required). That is,**

   - Method for passing control information from the open system to the mainframe
   - End of dataset determination by the mainframe application (for example, test for RECORD contents or provide mainframe application with RECORD count)
   - Notification to the mainframe when open system writes are completed
   - Notification to the open system host when information has been processed and can be reused

▼ To Configure the A7000

This procedure is for the person doing the A7000 configuration.

Prerequisite: All existing mainframe volumes allocated on the A7000 are offline.

1. **Allocate A7000 storage for the open system devices.**
2. **Configure SCSI Target Emulation (STE) to open system hosts providing a phantom header/tail, if required. See Chapter 7 for phantom header/tail requirements.**
3. **Configure the DataShare Facility devices to mainframe hosts (open system virtual disk, VOLSER, dataset name and LRECL/BLKL (BLKSIZE) factor, raw access function, CKD presentation function, and conversion functions, if required).**
4. **Activate the new DataShare Facility devices, STE devices, and SIMCKD.**

Open Systems Configuration

This procedure is for the open systems administrator.

Configure the open system host to recognize the A7000 virtual disks.

Note – A reboot of the open system host is often required before new SCSI devices are recognized. Refer to the open system host vendor documentation.

Mainframe Configuration

Confirm that the mainframe I/O generation is consistent with the requirements.

Note – The existing mainframe volumes allocated on the A7000 must be taken offline for a short time before the new devices can be brought online.

CHAPTER 4

Special Considerations

This chapter contains considerations that are important for the DataShare Facility administrator and user.

- Mainframe DataShare Capability (MFDSF)—page 4-2
- VSE Limitations—page 4-5
- Open Systems DataShare Capability (OSDSF)—page 4-9

Mainframe DataShare Capability (MFDSF)

The following subsections contain useful information to keep in mind when using the mainframe DataShare capabilities of the DataShare Facility. See “VSE Limitations” later in this chapter for VSE information that can impact MFDSF.

Concurrent Access

- Multiple open system hosts can simultaneously access a DSF device.
- Multiple readers can read a mainframe data shared volume or dataset.
- Mainframe writes must not occur to volumes that are being read by an open system host. See “Cache Synchronization” later in this chapter.

Dataset Characteristics - MVS

Dataset	Description
Size	Reported size is actual dataset size, unless the dataset has embedded short blocks. If a dataset has embedded short blocks, reported size is larger than actual size, and the content is NULL character filled after the end of data.
Dataset Names	When the data presentation function is ISO-9660, dataset names that are longer than 30 characters are truncated to 29 characters. A truncated file name begins with an underscore character (_), followed by the last 29 characters of the dataset name. The dot characters (.) in the dataset name are translated as underscore (_) characters to conform to the ISO-9660 standard. The volume sequence number is appended to the file name in the file extension field.
Types	Physical Sequential (QSAM) only, either Fixed Record / Unblocked or Fixed Record / Blocked.
Create Date	Files presented in the data presentation function have a create date associated with the create date of the dataset.
Open System Access	Fixed block datasets can be read sequentially or randomly.

Mainframe Software Data Compression

- Do not use mainframe software data compression on volumes that will be data shared unless a corresponding decompression application is available for the open system host.

Multivolume Datasets

- All volumes that may contain an element of a multivolume dataset must be on the StorEdge A7000 platform and be accessible by the open system host.
- The ISO-9660 presentation maps the dataset sequence number of each member to the file name extension after a dot (.) character (for example, `file.1` and `file.2`).
- On a UNIX system, the mainframe volumes associated with a group can be mounted under a common directory. We suggest that the mount point under this common directory be the VOLSER relating it to the mainframe data location. You can use shell wildcards to serially obtain all members of the multivolume dataset.

  UNIX Example:
  If DSF devices are mounted to the common directory `/MFDSFgroup1`, do the following to list all members of the multivolume dataset in this group:

```
# ls  /MFDSFgroup1/*/dataset.*
```

Similarly, you can use standard UNIX library calls for programmatic access to the datasets. Other non-UNIX operating systems provide similar functions.

Performance

DSF conversion functions are for convenience and may impact performance. Because conversion functions are often faster on the mainframe or open system host than on the A7000 processor, consider performing conversions on the mainframe or open system host based on resource availability.

Cache Synchronization

Cache is entirely coherent in the A7000, ensuring that the data presentation functions are always current with the latest data written by the mainframe. However, as most open systems cache disk data internally, a remount of the ISO-9660 image is required when data has been changed by the mainframe. If the mainframe has changed a volume VTOC that impacts the ISO-9660 image, an open system remount *must* occur; otherwise, data read by applications on the open system may not be current because of the ISO-9660 image change and local caching. See Chapter 7.

System Managed Storage (SMS)

- For UNIX-type open systems, members of an SMS volume group should be mounted to mount points under a common directory. Because you can use shell wildcards in the file path name (for example, `ls /MFDSFgroup1/*/dataset.*`) or programmatic access, you need not know the VOLSER of a specific volume that contains a dataset. An SMS pool of storage must be treated as one entity. If mainframe writes are occurring to an SMS pool, the open system host must not access any of these volumes. If open system reads are occurring, the mainframe must not write to this pool.
- An SMS pool of storage must be treated as one entity. If mainframe writes are occurring to an SMS pool, the open system host must not access any of these volumes. If open system reads are occurring, the mainframe must not write to this pool.
- If SMS pools are being used for MFDSF capability volumes, ensure that SMS initiated activity does not occur during open system read activity.

Volume Characteristics

Volume	Description
Volume Types	All 3380 and 3390 volumes are supported by the DataShare Facility with one exception. 3390-3 is excluded because some open system hosts are restricted to ISO-9660 images of 2 Gbytes.
VOLSER	Mainframe VOLSER is represented as an ISO-9660 volume ID.

VSE Limitations

The following subsections describe the limitations of VSE to keep in mind when using the mainframe DataShare capabilities of the DataShare Facility.

Dataset Characteristics

Dataset	Description
Size	Because VSE does not provide block length (BLKL), record length (RECL), and last track and record (LSTAR) information in the volume table of contents (VTOC), dataset size is calculated based on the maximum amount of information that can fit into the extents allocated to the dataset. In general, this means that the reported dataset size is larger than the actual size. Dataset content is NULL character filled after the actual end of data. When creating VSE datasets for use with mainframe DataShare capability, it is advisable to closely match extent size with actual data size and preferable to release any nonallocated space.
Dataset Names	When the data presentation function is ISO-9660, dataset names that are longer than 30 characters are truncated to 29 characters. A truncated file name begins with an underscore character (_), followed by the last 29 characters of the dataset name. The dot characters (.) in the dataset name are translated as underscore (_) characters to conform to the ISO-9660 standard. The volume sequence number is appended to the file name in the file extension field.
Types	Because VSE does not provide record format (RECFM) information in the VTOC, it is impossible to distinguish between Fixed and Variable length record datasets based on a VTOC entry; therefore, all Physical Sequential datasets are presented regardless of format. Applications should cooperate on structure and format of the information being exchanged.
Create Date	Files presented in the data presentation function have a create date associated with the creation date of the dataset.
Access	Read VSE datasets sequentially only. Random reads of datasets give indeterminate results.

AF=CKD

- Only physical sequential, fixed block/fixed record datasets are handled, but the VTOC does not provide enough information for DSF to determine record format. DSF tries to process any physical sequential dataset as fixed block/fixed record. If the dataset is not fixed block/fixed record, output may be of limited value. Applications should cooperate on structure and format of the information being exchanged.
- Dataset size must be calculated based on the maximum amount of information that can fit into the extents allocated to the dataset. In general, the reported dataset size is larger than the actual size. Dataset content is NULL character filled after the actual end of data. When creating VSE datasets for use with mainframe DataShare, closely match extent size with actual data size. Applications should agree on an end-of-file (EOF) format for the information being exchanged.
- EBCDIC to ASCII conversion is supported but delimiters are not (lack of record length information in the VTOC).
- Specify `VTOC=VSE` in the `/etc/dsf.cf` file.

AF=RAWRO

DataShare supplies record and block size information to the mainframe, but this information is ignored by VSE. Therefore, you must manually supply this information to the JCL or application used to access data presented with RAWRO.

AF=RAWRW and DataShare Open Backup

- The supported backup utility on VSE is FCOPY.
- Only full volume backups are supported.

DASD Tools

Because of VSE limitations, some of the DASD Tools functions are not supported, or have limited support, when using VSE.

Function	Descripiton
dasd_cat	The dasd_cat -irs, -eor, and -recfm options allow you to supply record length, delimiter, and format information.
dasd_db_init	Supported.
dasd_file	Limited support due to lack of VTOC information. Only the type of dataset (for example, physical sequential) is available.
dasd_ls	Limited support due to lack of VTOC information. Tracks-used information is not available.
dasd_pds	Not supported.
dasd_retrieve	Supported.
dasd_vol	Supported.
sms_cat	Supported.
vsam_alt_cat	Not supported.
vsam_cat	Not supported.

Mainframe Issues

- Ensure that all datasets that are to be presented to the open system are Physical Sequential.
- Determine the block size, record size, and number of records for each dataset. Record the information at the start of the dataset to be shared, in another dataset that can be shared, or manually.

Database Issues

- Do not attempt parallel loading from a single dataset.
- Some database loaders require end-of-line (EOL) characters after each record.
  - On loaders *with* the EOL requirement:

    `PF=ISO` An intermediate process (for example, `dd` command) must be used to insert EOLs.

    `dasd_cat` Use the `-irs` and `-eor` options and pipe the output to the loader.
  - On loaders *without* the EOL requirement:

    `PF=ISO` The dataset can be read directly.

    `dasd_cat` Pipe the output to the loader.

Open Systems DataShare Capability (OSDSF)

The following subsections contain useful information to keep in mind when using the open systems capability of the DataShare Facility.

Additional Mainframe Considerations

- When the DataShare Facility provides an open system virtual volume to the mainframe, the volume contains one or more datasets representing chunks of the open system raw disk. The volume free space is always zero.
- Mainframe writes to open system DSF volumes are discarded and no error condition is returned.

Concurrent Access

Only one open system host can have permission to write to a specific virtual disk during a block of time. On that host, the operating system or application is responsible for managing multiple writes to a virtual disk.

Dataset Characteristics

Dataset	Description
Dataset Names	Dataset names are preconfigured on the A7000.
Size	Size is determined by the DSF software configuration or by the physical size of the open system partition being shared. Block or record size is determined by the software configuration.
Types	The structure presented is always a Physical Sequential (QSAM) dataset.
Time Stamps	Time stamps are the date and time that the DataShare Facility device was enabled on the A7000.

Synchronization

The open system host applications or processes must ensure appropriate synchronization for shared devices. The open systems DataShare capability provides a consistent view of the information that is written to the A7000 virtual disk from the open system host to the mainframe.

Volume Characteristics

Volume	Description
Volume Types	All 3380-K and 3390-3 volumes are supported.
VOLSER	The VOLSER is assigned by the DSF software configuration.

CHAPTER 5

DataShare Facility Applications

This chapter describes the types of applications that can benefit from the DataShare Facility.

- Introduction—page 5-2
- Eliminating Data Replication and Movement—page 5-2
- Function Consolidation—page 5-5
- Function Offload—page 5-5

Introduction

DataShare Facility (DSF) is an effective solution to many existing data center problems and also provides new opportunities for operational efficiency. The following types of applications can benefit from the DataShare Facility:

- Elimination of data replication
- Function consolidation
- Function offload

Note – DSF is not intended to replace networks where the amount of shared data is small.

Continue using a network solution for those applications requiring small cross platform data access that can be handled easily using network hardware and software packages. DSF is an alternative for small cross platform access when any of the following conditions are present:

- Networks are a security concern
- No common network between mainframes and open systems is in place
- Transfer window is a concern because of the quantity and reliability of transfer

Eliminating Data Replication and Movement

Because computing environments often contain many different platforms, the transfer of transaction-based information from a mainframe to open system-based decision support systems or data warehouses is essential. In these environments, the timeliness and amount of detailed information transferred relates to the accuracy of business decisions that can be made on the data warehouse information. Without the DataShare Facility, timeliness and detailed data are at odds with the solutions used to address this transfer activity (i.e., network and tape).

The DataShare Facility eliminates the need to transfer the information between the operational system and the data warehouse prior to loading and enables the timely loading of transaction information into the data warehouse.

FIGURE 5-1 illustrates data replications without DSF. FIGURE 5-2 illustrates how DSF eliminates the transfer, allowing more detailed and frequent data warehouse updates.

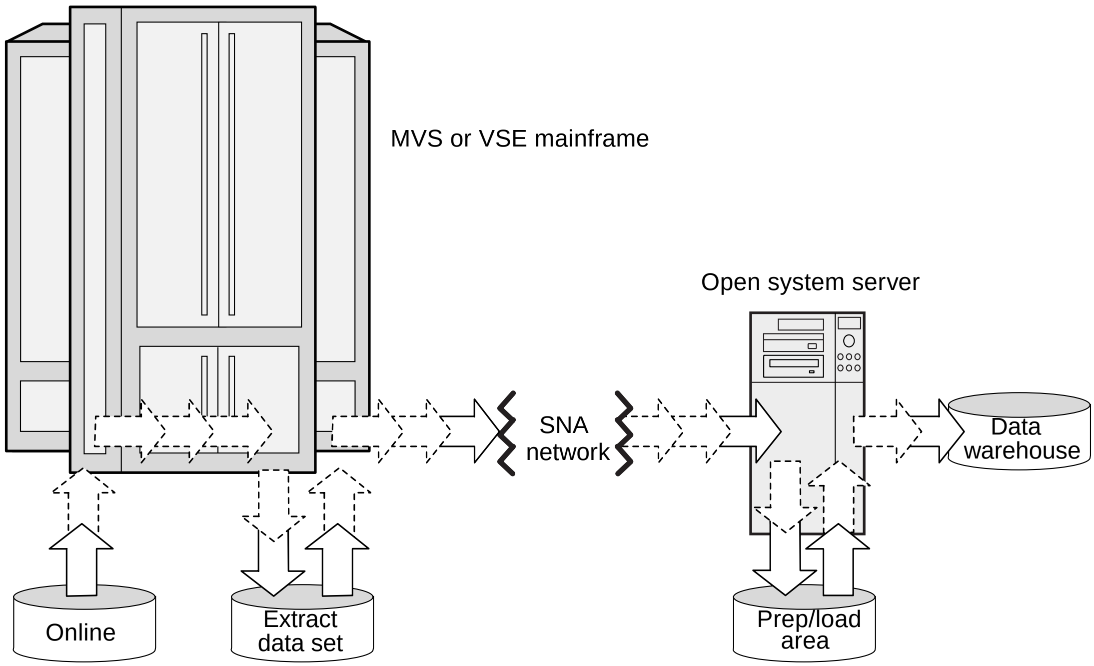

FIGURE 5-1 Traditional Data Replication

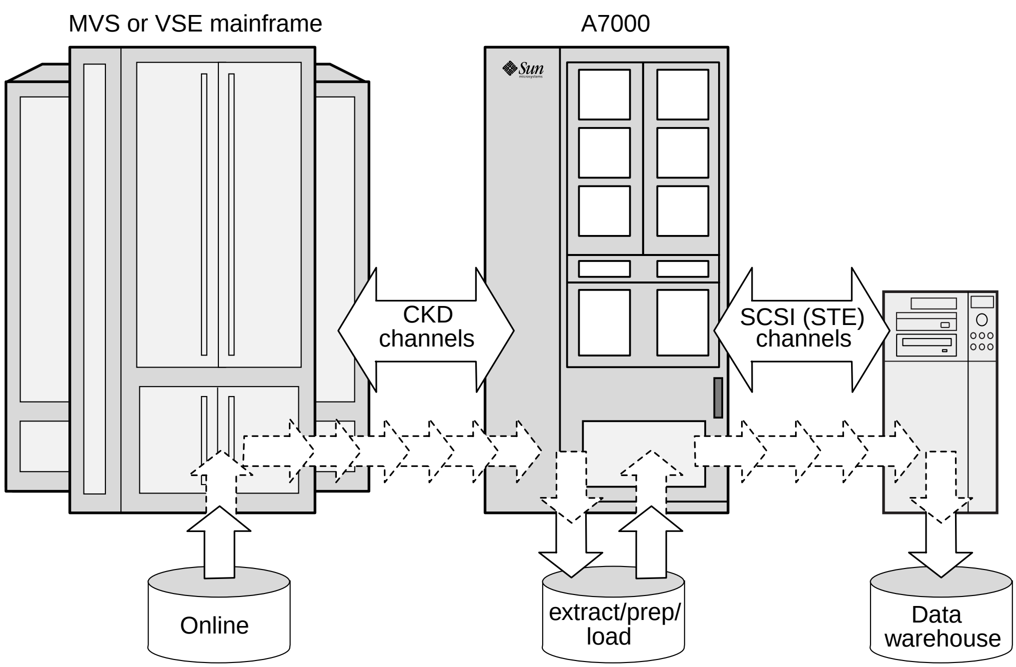

FIGURE 5-2 Eliminating Data Replication with DataShare

Data replication in a data warehouse load/update in a network transfer environment *without* the DataShare Facility is typically done in three steps:

1. Operational data extraction
2. Data transfer
3. Data warehouse load/update

Maintaining the information on both systems duplicates storage resources, causes significant time loss, and makes detailed information transfer impractical.

Using the DataShare Facility, the information extracted from the mainframe data base is written directly to a StorEdge A7000 mainframe volume and accessed directly by the data warehouse system. The data transfer is eliminated. Data can be loaded directly into the data warehouse by standard data base loaders using the DataShare Facility to access the extracted information on shared volumes.

Mainframe extraction volumes are configured for the mainframe DataShare capability and are read as ISO-9660 volumes by the loader on the data warehouse open system. For performance reasons, we recommend that conversion (if required) be done as part of the mainframe extraction or data warehouse loader function in this high throughput application.

Mainframe Usage

Sequential datasets on shared mainframe volumes to the open system hosts are accessed in the same manner as nonshared mainframe volumes to other mainframes. To synchronize mainframe and open system applications, mainframe I/O activity for a shared volume must be quiescent prior to open system access.

▼ To Use Mainframe DataShare Capability

Mainframe Read/Write Access:

1. **Stop open system access to the CD-ROM image.**
2. **Ensure the MFDSF volume is online.**
3. **Initiate the mainframe application writes to the QSAM datasets (fixed block only) on the MFDSF volume.**

Open System Read Access:

1. **Stop mainframe write access to this volume and ensure there is no further write access until completion of the open system read activity.**
2. **Enable the open system read access. Remount the ISO-9660 image (unmount, mount). On an NT system, an access to the ISO-9660 image refreshes the image.**
3. **Start the open system application reads.**

Note – Other methods can be used to serialize volume access between systems.

Function Consolidation

Multisystem environments lack direct, efficient access to storage resources between systems, causing duplication of specific functions on every system. DataShare Facility can aid in function consolidation.

Some of the other areas in which DataShare Facility can help with function consolidation are:

- Internet servers
- Network access
- World Wide Web access

Function Offload

Data processing batch jobs can often benefit from additional hosts accomplishing some of the subcomponents of the process. Benefits can be seen in the following areas:

- Increased process speed (same data, less time)
- Increased process throughput (more data, same time)
- Decreased process cost (MIPS and license cost)
- Parallel processing advantages

These benefits can be realized only in *homogenous* mainframe shared Direct Access Storage Device (DASD) systems or in applications requiring minimal data transfer and access between subprocesses because of the inherent overhead and latency of data transfer between process steps.

The following figure illustrates how this latency impacts overall process time without the DataShare Facility and how this latency is eliminated when using the DataShare Facility, gaining the same benefits between heterogeneous systems that can be achieved in *homogenous* mainframe shared Direct Access Storage Device (DASD) systems.

The following procedure is an example of mainframe offload to an open system using DataShare Facility:

1. Store input to the offload process (data and control information) from the mainframe process onto the A7000 mainframe storage.
2. Access input as an ISO-9660 file system from an open system access process using the mainframe DataShare capability.
3. Execute the process on the open system platform.
4. Store output (data and control information) from the open system onto the A7000 open system raw storage.
5. Access process output from the mainframe using the open systems DataShare capability.

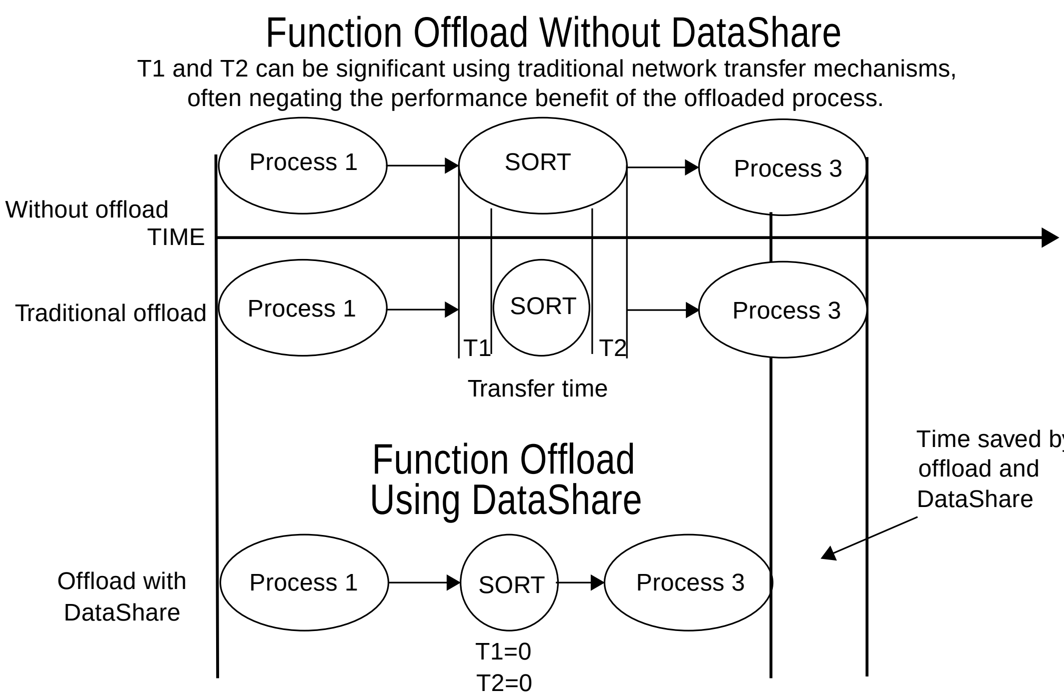

FIGURE 5-3 Function Offload Without and With DataShare Facility

Mainframe Usage

Volumes that are shared between a mainframe and open system hosts are handled as nonshared volumes by the mainframe. To synchronize mainframe and open system applications, we recommend that mainframe I/O activity for a shared volume be quiescent prior to open system access.

▼ To Use the Mainframe DataShare Capability

Mainframe Read/Write Access:

1. **Stop open system access to the CD-ROM image.**
2. **Ensure the MFDSF volume is online.**
3. **Initiate the mainframe application writes to the QSAM datasets (fixed block only) on the MFDSF volume.**

Open System Read Access:

4. **Stop mainframe write access to this volume and ensure there is no further write access until completion of the open system read activity.**
5. **Enable the open system read access. Remount the CD-ROM image (unmount, mount), or on an NT system, refresh the CD-ROM image.**
6. **Start the open system application reads.**

Open Systems Usage

Shared open system volumes are accessed by the mainframe as nonshared mainframe volumes. One or more sequential datasets, each containing all of the open system's information specified by the defined Access Functions, are presented to the mainframe. See the sample `dsf.cf` file in Appendix A for more information on Access Functions.

▼ To Use the Open Systems Capability

Do the following from the open system:

1. **Write information to the A7000 virtual disk using open system native raw I/O semantics. The following example is on a UNIX open system host:**

   write_application **`|dd of=/dev/`***SP_virt_disk* **`cbs=`***xx* **`conv=ebcdic`**

2. **Notify the mainframe when the information is available, and specify how to determine end of file.**

Do the following from the mainframe:

3. **The mainframe application reads from the dataset on the shared volumes specifying the preconfigured block and logical record sizes and determines the end of file as specified in Step 2.**

4. **Notify the open system when the virtual disk can be overwritten.**

CHAPTER **6**

Best Business Practices

This chapter contains best business practices for specific configurations, which will help you understand how to plan and set up DataShare Facility (DSF).

- Example — Mainframe DataShare Capability (MFDSF) with MVS—page 6-2

Example — Mainframe DataShare Capability (MFDSF) with MVS

The Sun Ultra Sparc currently uses a network connection to transfer 10 Gbytes of mainframe DB/2 extract data to the Sun, and then loads that data into an Oracle data base.

The time required to do the transfer is considered excessive (~ 4hrs) and with an expected 5% growth of data per month, the current transfer method is too complicated and takes too long.

Mainframe Environment

- IBM 9021
- MVS/ESA
- DB/2
- TCP/IP network to the Sun and the TCP/IP REXEC and RPC are available

Open System Environment

- Sun Ultra Sparc
- Solaris
- Oracle (the data base currently resides on local Sun disks)
- SCSI Fast and Wide Controllers

▼ To Plan Replacement of Current Method

The following steps describe a planning approach for replacing the current network transfer method with DSF sharing of the mainframe volumes containing the DB/2 extract data.

1. **Calculate the maximum amount of data to be shared.**

   Initially there are 10 Gbytes. By the end of the year — 10 Gbytes adjusted for 5% growth per month = 10 * 1.795.

2. **Use 3390-2 mainframe volumes. Determine the number of mainframe volumes required.**

   Mainframe 3390-2 actual data capacity (rule of thumb) = 1.5 Gbytes

   Number of mainframe 3390-2 volumes = 18/1.5 = 12 volumes

3. **Determine the mainframe unit address.**

   Unit addresses 300-305 and 410 - 415 are designated as MFDSF volumes.

4. **Determine how these volumes will be made available to the extract applications.**

   Use a unique volume esoteric designation (`MFDSF`). In this example, do not define an SMS group for these volumes.

5. **Determine the number of open system logical disks required.**

   # of open system logical disks required = # of mf volumes required for the shared data = 12

6. **Determine the number of open system SCSI channels required.**

   # of open system SCSI Channels required = # of open system logical disks required / Max LUNs per channel = 12/16 (Fast and Wide can support up to 16) = 1 channel required

   We have elected to spread the I/O over two channels to enhance performance. Each channel will be configured for six logical units (LUNs).

Note – Because the 300 address ranges are on `dsp1` and the 410 address ranges are on `dsp2`, two physical cables are required.

▼ To Configure the A7000

This procedure is for the person doing the A7000 configuration.

1. **Establish the relationship between the mainframe unit addresses and the A7000 physical devices.**

2. **Configure the STE and DSF devices using DASD Manager.**

   The DSF devices are configured as read-only CD-ROM volumes and made available to the open system host.

3. **Start the DSF and STE components.**

▼ To Configure the Open Systems

1. **Create the CD-ROM mount points.**

   This example has a common directory for all the mount points associated with the MFDSF volumes. We chose the name MFDSF so that it relates to the mainframe esoteric name for the volumes.

```
# mkdir /MFDSF
# mkdir /MFDSF/VDSF01 /MFDSF/VDSF02 .... /MFDSF/VDSF0C
```

2. **Create scripts, specific to the open system host, to facilitate mounting and** *unmounting* **of these CD-ROM volumes.**

```
# mount_MFDSF
# umount_MFDSF
```

▼ To Use the MFDSF Volumes

On the Sun Ultra Sparc

- **Unmount the CD-ROM:**

```
# umount_MFDSF
```

On the mainframe

- **Run the JCL on the mainframe to extract the DB/2 information for the Sun Ultra Sparc. Use volumes assigned to** `UNIT=MFDSF`**. When all writing to these volumes is complete, use the mainframe network submission REXEC to submit a job to the Sun Ultra Sparc, which begins the data base load process.**

On the Sun Ultra Sparc

The example contains DB_load.sh scripting.

CODE EXAMPLE 6-1 DB_load.sh - PSEUDO scripting

```
# remount the MF DataShared Volumes
       mount_MFDSF

# Prepare the data base for the load
    DB_load_prep

# Load all of the tables
    for TABLE in .........
    do
# For each part of a multivolume dataset for TABLE
    for TABLE_PART in /MFDSF/*/${TABLE}.*
    do
     DB_loader TABLE_CTL_FILE $TABLE_PART

# Where: DB_loader is the data base loader application
# TABLE_PART is the next part of the DSN to be loaded
# TABLE_CTL_FILE is the DB loader control file for
# this $TABLE
    done
    done

# Build the appropriate indexes for all tables
    DB_build_indexes

# DB load complete ........
```

CHAPTER 7

Open System Platform-Specific Issues

The chapter contains information that you must be aware of when using DataShare Facility with various platforms.

- HP 9000/HP_UX—page 7-2
- Intel/Windows NT—page 7-2
- RS6000/AIX—page 7-3
- Sun/Solaris—page 7-3

HP 9000 / HP_UX

Mainframe DSF	Open System DSF	Size Limitation
Synchronization: ISO-9660 image refresh. Remount the file system. File name issues: No known file name issues. Record delimiter: line feed	Raw I/O support. Standard UNIX (e.g., `dd` or `cat`) Phantom Header: None Phantom Tail: None	Maximum file size is 2 Gbytes. Maximum file system size is 4 Gbytes. Maximum partition size is 4 Gbytes. Maximum disk size is unknown, but is at least 10 Gbytes.

Intel / Windows NT

Mainframe DSF	Open System DSF	Size Limitation
Synchronization: ISO-9660 image refresh. File name issues: truncation extensions Record delimiter: carriage return line feed	Raw I/O support. Phantom Header: None Phantom Tail: None	No known size limitations.

RS6000 / AIX

Mainframe DSF	Open System DSF	Size Limitation
Synchronization: ISO-9660 image refresh. Remount the file system. File name issues: none Record delimiter: line feed	Raw I/O support. Standard UNIX (e.g., `dd` or `cat`) Phantom Header: None Phantom Tail: None	Maximum file size is 2 Gbytes. Maximum file system size is 2 Gbytes. Maximum partition size is 4 Gbytes. Maximum disk size is unknown.

Sun / Solaris

Mainframe DSF	Open System DSF	Size Limitation
Synchronization: ISO-9660 image refresh Remount the file system. File name issues: none Record delimiter: line feed	Raw I/O support. Standard UNIX (e.g., `dd` or `cat`) Phantom Header: 8192 blocks Phantom Tail: 8192 blocks	Maximum file size is 2 Gbytes. Maximum file system size is unlimited. Maximum partition size is unlimited. Maximum disk size is unknown.

APPENDIX **A**

Sample `dsf.cf` File

This appendix contains a sample DataShare Facility configuration file.

- Example of a `dsf.cf` File—page A-2

Example of a `dsf.cf` File

```
#
# Sample DataShare Facility configuration file (/etc/dsf.cf).
#
# A line starting with “#START OF SAMPLES” marks the start of the
# samples. A line starting with “#END OF SAMPLES” marks the end of the
# samples.
#
# Lines starting with ‘#’ are comments and are ignored.
#
# Each dsf.cf entry defines a data PRESENTATION.
#
# The PRESENTATION definition begins with “dsfn”, where n is the device
# number in hex, examples dsf0,..9,A-F,10 etc..
# The maximum number of dsf devices over all nodes is defined
# by the /etc/system file parameter “dsf”, which is in decimal.
# The last DSF device is always allocated to the dsf daemon which
# manages synchronization of devices.
#
# The PRESENTATION definition ends with the keyword “END” (uppercase
# required).
#
# The definition contains KEYWORDS with VALUES, and optional ARGUMENTS.
#
# Mandatory keywords are:
# ======================
#
# MASTER= subsystem ID (2 | 3) or (4 | 5)
# AF= Access Function(s) used to access the data for this presentation.
# PF= Presentation Function used to present the data.
#
# Optional keywords are:
# =====================
# SLAVE= other node id (3 | 2) or (5 | 4)
# CF= Conversion Function(s) to use to process the data.
#
# Allowable keyword / value / arguments:
# =====================================
# The following table describes allowable keyword / value / arguments.
# Arguments must be enclosed in parenthesis “()”, and at least one
# space should separate the parenthesis from the KEYWORD value.
#
# Optional arguments are marked with an “[]”.
#
```

```
# Arguments within parenthesis must be comma-separated.
#
# KEYWORD  VALUE(S)       ARGUMENTS             COMMENTS
#
# MASTER=  2 | 3 or                             Master node id
#                                               (subsystem id).  4 | 5
#                                               Must be specified.
#
# SLAVE=   3 | 2 or          none               Slave node id
#          5 | 4                                Must be specified.
#
# AF=      _access_function_                    Function(s) used to get
#                                               data.  There are a
#                                               maximum of 8 AF per
#                                               presentation function.
#
#          CKD            (DISK=/dev/rdsk/xyz   CKD format, provide disk
#                                               path.
#                          [,VTOC=VSE])         indicates that the VTOC
#                                               was written by VSE instead
#                                               of MVS. NOTE: If VSE,
#                                               CF=DELIM should NOT be used
#                                               VSE does not provide
#                                               BLKL or RECL.
#                          [,INFO=filename])    Causes an extra file to
#                                               be created on the ISO9660
#                                               image containing
#                                               complete VTOC
#                                               information.  DSN, DSORG,
#                                               RECFM, BLKL and RECL.
#
#          {RAWRO | RAWRW}                      RAWRO is read only disk
#                                               data, RAWRW is read and
#                                               write disk data, use
#                                               RAWRO for OSDSF
#                                               capability, use RAWRW
#                                               for DataShare Open Backup
#                         (DISK=/dev/rdsk/xyz   Path to disk data.
#                                               The same disk may be
#                                               defined in more than one
#                                               AF in a dsf entry.
#                          ,NAME=xyz            Entry name as it appears
#                                               in the presentation
#                                               function.  Name must
#                                               comply with rules for
#                                               its final usage.  For
#                                               example, if PF will be
#                                               CKD, then translation
#                                               from lower to upper case
```

```
#                                                  will be performed, but
#                                                  the name must be a valid
#                                                  MF DSN.
#                          [,START=nnnn]           Start location in
#                                                  sectors, the default is
#                                                  0.
#                          [,BLKL=n]               Block length (BLKSIZE)
#                                                  in bytes.  Must be less
#                                                  than 32K.  Must be a
#                                                  multiple of 512 bytes &
#                                                  an integer multiple of
#                                                  LRECL. Default depends
#                                                  on PF & options.
#
#            NOTE: When using RAW AF, you must specify an LRECL and NUMREC
#                   or a SIZE parameter. This is no longer optional.
#
#                          ,LRECL=n,NUMREC=n       Logical record length in
#                                                  bytes.
#                                                  NUMREC is the number of
#                                                  logical records.
#                                                  If NUMREC is specified,
#                                                  the SIZE is calculated
#                                                  based on BLKL & LRECL
#                                  OR
#                          SIZE=nnnn               SIZE must be specified.
#                                                  SIZE, if specified, is in
#                                                  sectors, and must never exceed the
#                                                  size that can be supported
#                                                  by he PF(see table below).
#
#          UFS           (DISK=directory path) Unix file system format
#                                                  access. Specified directory
#                                                  path should identify the
#                                                  root of the directory
#                                                  structure to be used.
#                                                  The directory structure
#                                                  can not be more than eight
#                                                  levels deep.
#
# CF=      _conversion_function_                   Conversion function (see
#                                                  below)
#
#   [(AFlist)] DELIM      [(a,b,...c)]             Delimiter characters
#                                                  (see below) Default is
#                                                  <LF>, octal 012.
#                                                  NOTE: Don't use if
#                                                  VTOC=VSE. VSE doesn't
#                                                  provide BLKL/RECL.
```

```
#
#   [(AFlist)] ETOA       [([MAP=/fullpath][,]  EBCDIC to ASCII
#                                               conversion. The MAP
#                                               option can specify the
#                                               full path to an
#                                               alternate ETOA map.
#
#
#   [(AFlist)] ATOE       [([MAP=/fullpath][,]  ASCII to EBCDIC
#                                               conversion. The MAP
#                                               option can specify the
#                                               full path to an
#                                               alternate ATOE map.
#
#
# PF=      _presentation_function_              Function to use to
#                                               present data
#
#          ISO9660        [(VDISK=NN)]          ISO9660 CD-ROM emulation
#                                               following ISO-9660
#                                               specification. Optional
#                                               VDISK argument specifies
#                                               vdisk number for hosts
#                                               requiring a SCSI media
#                                               change notification. NT
#                                               requires VDISK option.
#
#
#          CKD            (DTYPE={3380|3390},   Type of CKD disk to
#                                               emulate.
#                          VOLSER=xxxxxx        Volume VOLSER (6 chars)
#                          [,BLKL=n]            Block length in bytes.
#                                               Must be less than 32K.
#                                               Overrides volume level
#                                               default. Only use with
#                                               AF=RAW*. Must be a
#                                               multiple of 512 bytes
#                                               and an integer multiple
#                                               of LRECL.
#                          [,LRECL=n]           Logical record length
#                                               in bytes. Overrides
#                                               volume level default.
#                                               Only use with AF=RAW*.
#                          [,HLQ=XXXXXXXX.[YYYYYYYY.[ZZZZZZZZ.]]])
#                                               High level qualifier to
#                                               be prepended to NAME.
#                                               Up to 3 levels, and up
#                                               to 8 chars, per level.
#                                               Must end in a “.”
```

```
# Notes on AF functions and arguments:
# ===================================
#
# A dsf definition can specify multiple access functions. The sum of
# all data from each AF will be used in the data presentation.
#
# RAW Access Functions, RAWRO and RAWRW
# When configured for RAWRW, writes to the VOLSER track and VTOC track
# are ignored (but no errors are generated). Other writes modify disk
# data as appropriate.
# When configured for RAWRO, all writes are ignored. No errors are
# generated.
#
#   BLKL, LRECL, NUMREC & SIZE Arguments for RAWRO and RAWRW Access
#   Function:
#
#   for CKD data presentation, it may be desirable to explicitly
#   configure MVS VTOC parameters for data being sourced from a raw disk
#   partition. When specified, the BLKL and LRECL parameters will
#   appear in the VTOC entry corresponding to the data from the RAW*
#   access function. When multiple RAW* access functions are used in
#   the same presentation, each access function may be individually
#   configured.
#   The NUMREC argument may be used to indicate the number of
#   Logical Records that have valid data.
#   If this value is provided, the LSTAR value in the VTOC entry is
#   modified accordingly. If NUMREC and SIZE is not specified, the
#   dataset size will be calculated to represent the entire disk
#   partition.
#
#
# Notes on CF functions and arguments:
# ===================================
#
# In the event that a configuration consists of data from multiple
# access functions, the optional “AFlist” can be specified in the CF
# specification. For example, CF=(1,3)ETOA may be used to indicate
# which of the access functions the conversion process is to be used
# with, where 1 corresponds to the first AF, etc. If no “AFlist” is
# specified, the conversion function is applied to data from all access
# functions.
#
#   DELIM arguments:
#
#   the optional argument to DELIM should be a comma separated list of
#   the three digit octal ascii codes for the characters to be used as
#   record separators. For instance, if the <CR> <LF>characters should
#   be used (typical for PCs), the argument should appear as:
#   CF=DELIM (015,012)
```

```
#   the default value is (012), <LF>, which is adequate for most UNIX
#   systems.
#   DELIM should NOT be used if the volume was created by a VSE system.
#   The VSE VTOC information does not contain the record length
#   information that is required for this option to function.
#
#   ETOA/ATOE arguments:
#
#   the optional argument to the ETOA/ATOE conversion functions should
#   be a full path to a 256 byte data file. The value of each character
#   (byte) in the file should represent the desired translated character
#   for that character value. For instance, the converted value of
#   character value octal [000] should appear as the first byte in the
#   file. For each unique usage of a conversion map, 64KB of kernel
#   memory is allocated.
#
#   An example of usage of this argument is:
#     CF=ETOA (MAP=/usr/install/dsf/custom/cmap1)
#   The default ETOA/ATOE maps are from a proposed ANSI standard.
#   Several alternative maps have been provided on the distribution.
#   These are documented in the MAPS(5DSF) manual page.
#
#
# Notes on PF functions and arguments:
# ===================================
#
#   CKD PF BLKL and RECL
#
#   In the event that BLKL and LRECL is not available from the Access
#   Function layer, the BLKL and LRECL for each dataset will be assigned
#   according to the following table:
#
#        DEFAULT
# DTYP  BLKL/RECL TRACK CAPACITY  VOLUME DATA CAPACITY
# 3380   23040     45 KB(46080)   2654*15*45K=1749.46MB=1.708GB=3582900 sectors
#
# 3390   27648     54 KB(55296)   3338*15*54K=2640.41MB=2.578GB=5407560 sectors
#
#
#START OF SAMPLES
# Sample dsf.cf configurations:
#
# Mainframe Datashare (MFDSF):
# ===========================
#
#   iso9660 presentation of raw ebcdic data:
#   dsf0 MASTER=2 SLAVE=3 AF=CKD (DISK=/dev/rdsk/14d6) PF=ISO9660 END
#
#   iso9660 presentation of converted ebcdic data:
```

```
#   dsp2 is the primary server for the disk 14d6
#   dsf1
#     MASTER=3 SLAVE=2
#     AF=CKD (DISK=/dev/rdsk/14d6) CF=DELIM CF=ETOA PF=ISO9660 END
#
#   iso9660 presentation of data for a NT host:
#   at the end of each record add <LF> <CR>,
#   converted ebcdic data to ascii and
#   notify vdisk14 of a media change when the MF data is modified.
#   Where 14 is the vdisk number specified in the /etc/ste.cf,
#   providing access of /dev/rdsk/dsf2 to the NT.
#   dsf2
#     MASTER=2 SLAVE=3
#     AF=CKD (DISK=/dev/rdsk/14d6)
#     CF=DELIM (015,012) CF=ETOA
#     PF=ISO9660 (VDISK=14)
#   END
#
#   iso9660 presentation of data from multiple CKD volumes with
#   individual ETOA mapping.
#   dsf3
#     MASTER=2 SLAVE=3
#     AF=CKD (DISK=/dev/rdsk/14d6)
#     AF=CKD (DISK=/dev/rdsk/14db)
#     CF=DELIM
#     CF=(1)ETOA (MAP=/usr/install/dsf/custom/etoa1.map)
#     CF=(2)ETOA (MAP=/usr/install/dsf/custom/etoabtl.map)
#     PF=ISO9660
#   END
#
#   iso9660 presentation of data from VSE Volume.
#   Note: for VSE, no DELIM processing.
#   dsf4
#     MASTER=2 SLAVE=3
#     AF=CKD (DISK=/dev/rdsk/14db,VTOC=VSE)
#     CF=ETOA
#     PF=ISO9660
#   END
#
#
# Datashare Open Backup:
# ==================================
#
#   CKD presentation of a single open system partition
#   dsf10
#     MASTER=2 SLAVE=3
#     AF=RAWRW (DISK=/dev/rdsk/14de,NAME=unix.data.ascii.dsk14de)
#     PF=CKD (DTYPE=3390,VOLSER=UN14DE)
#   END
```

```
#
#   CKD presentation of a open system partition of 6GB(vp1,
#   12582912 sectors).
#   6GB is larger than the data space available on a 3390-3.
#   The maximum data capacity for a 3390-3 with a 27K BLKL(default) is
#   ~2640.41MB, which is 5407560 sectors (512 bytes). See table above.
#   dsf11
#     MASTER=2 SLAVE=3
#     AF=RAWRW (DISK=/dev/rdsk/vp1,NAME=unix.data.vp1.part1,
#               START=0, SIZE=5407560)
#     PF=CKD (DTYPE=3390,VOLSER=UVP1P1)
#   END
#   dsf12
#     MASTER=2 SLAVE=3
#     AF=RAWRW (DISK=/dev/rdsk/vp1,NAME=unix.data.vp1.part2,
#               START=5407560, SIZE=5407560)
#     PF=CKD (DTYPE=3390,VOLSER=UVP1P2)
#   END
#   dsf13
#     MASTER=2 SLAVE=3
#     AF=RAWRW (DISK=/dev/rdsk/vp1,NAME=unix.data.vp1.part3,
#               START=10815120, SIZE=5407560)
#     PF=CKD (DTYPE=3390,VOLSER=UVP1P3)
#   END
#   Specifying alternate values for BLKL and LRECL overrides these
#   defaults, and will change the track and volume capacity.
#
#
# Open Systems Datashare (OSDSF):
# ==============================
#   CKD presentation of an open system partition with configured
#   BLKL/LRECL
#       Note: BLKL < 32K and LRECL*N = BLKL (80*320=25600)
#   dsf20
#     MASTER=2 SLAVE=3
#     AF=RAWRO
#        (DISK=/dev/rdsk/14d7,NAME=unix.data.ebcdic.dsk14d7,BLKL=25600,
#         LRECL=80,NUMREC=10000)
#     PF=CKD (DTYPE=3390,VOLSER=UN14D7)
#   END
#
#   CKD presentation of an open system partition with configured
#   BLKL/LRECL and conversion from ASCII to EBCDIC
#   dsf21
#     MASTER=2 SLAVE=3
#     AF=RAWRO
#        (DISK=/dev/rdsk/14d7,NAME=unix.data.ebcdic.dsk14d7,BLKL=25600,
#         LRECL=80,NUMREC=10000)
#     CF=ATOE
```

```
#     PF=CKD (DTYPE=3390,VOLSER=UN14D7)
#   END
#
#   CKD presentation of multiple open system partitions with BLKL/LRECL
#       part[1,2,3,4] are parts of a partition.
#       The parts would normally not overlap, so START and SIZE describe
#       different sections of one partition.
#
#       part1 has been configured as 100 tracks,
#             each track is 3*16K blocks which gives a SIZE of 2400
#             sectors
#
#   dsf22
#     MASTER=2 SLAVE=3
#     AF=RAWRO (DISK=/dev/rdsk/14da,NAME=unix.data.part1,
#               BLKL=16384,LRECL=512,SIZE=2400)
#     AF=RAWRO (DISK=/dev/rdsk/14da,NAME=unix.data.part2,
#               BLKL=8192,LRECL=1024,START=2400,SIZE=2400)
#     AF=RAWRO (DISK=/dev/rdsk/14da,NAME=unix.data.part3,
#               BLKL=31744,LRECL=124,START=4800,NUMREC=500)
#     AF=RAWRO (DISK=/dev/rdsk/14da,NAME=unix.data.part4,
#               BLKL=20480,LRECL=20480,START=4922,SIZE=120)
#     PF=CKD (DTYPE=3390,VOLSER=UNIXT2)
#   END
#
#
#END OF SAMPLES
```

CODE EXAMPLE A-1 Sample `dsf.cf` File

APPENDIX **B**

Calculating Data Capacity

This appendix contains tables for calculating the actual data capacity for a volume containing specific dataset characteristics.

- Mainframe DataShare Capability—page B-2
- Open Systems DataShare Capability—page B-3

Mainframe DataShare Capability

Use the table below to determine the approximate capacity per volume based on the approximate block size that will be used. This is standard mainframe storage planning.

Always plan conservatively. That is, the amount of mainframe data that will be shared per volume type should be less than 2/3 of the maximum volume capacity of the mainframe volume.

TABLE B-1 Actual Data Capacity (Mainframe DataShare Capability)

TYPE	Cylinders DATA	Heads/ CYL	Block Size	# of Blocks per track	Actual Data Capacity (ADC)	Max O.S. Data in One MF Volume	Max # of 512 Byte Sectors
3380-K	2654	15	16K	2	1244MB	= 1.214GB	= 2547840
3380-K*	2654	15	22.5K	2	1749.46MB	= 1.708GB	= 3582900
3390-2	2225	15	16K	3	1564.45MB	= 1.527GB	= 3204000
3390-2*	2225	15	27K	2	1760MB	= 1.71GB	= 3604500

* indicates maximum capacity

Open Systems DataShare Capability

Use the table below to determine the approximate capacity per volume based on the block size required. This is standard mainframe storage planning.

Always plan conservatively. That is, the open system virtual disk should be less than 2/3 of the maximum volume capacity of the mainframe volume.

TABLE B-2 Actual Data Capacity (Open Systems DataShare Capability)

TYPE	Cylinders DATA	Heads/ CYL	Block Size	# of Blocks per track	Actual Data Capacity (ADC)	Max O.S. Data in One MF Volume	Max # of 512 Byte Sectors
3380-K	2654	15	16K	2	1244MB	= 1.214GB	= 2547840
3380-K*	2654	15	22.5K	2	1749.46MB	= 1.708GB	= 3582900
3390-3	3338	15	16K	3	2347.03MB	= 2.292GB	= 4806720
3390-3*	3338	15	27K	2	2640.41MB	= 2.578GB	= 5407560

* indicates maximum capacity

APPENDIX **C**

DASD Tools UNIX Man Pages

This appendix contains hardcopies of the DASD Tools UNIX man pages.

Man Pages

- dasd_cat(1DSF)—page C-3
- dasd_db_init(1DSF)—page C-8
- dasd_file(1DSF)—page C-11
- dasd_ls(1DSF)—page C-13
- dasd_pds(1DSF)—page C-15
- dasd_retrieve(1DSF)—page C-18
- dasd_vol(1DSF)—page C-20
- sms_cat(1DSF)—page C-22
- vsam_alt_cat(1DSF)—page C-24
- vsam_cat(1DSF)—page C-30
- dsftab(4DSF)—page C-37
- dsftab_db(DSF)—page C-38
- maps(5DSF)—page C-39

APPENDIX **D**

DASD Tools Windows NT Man Pages

This appendix contains hardcopies of the DASD Tools Windows NT man pages.

Man Pages

- `dasd_cat(1DSF)`—page D-3
- `dasd_file(1DSF)`—page D-7
- `dasd_ls(1DSF)`—page D-9
- `dasd_pds(1DSF)`—page D-11
- `dasd_vol(1DSF)`—page D-14
- `vsam_alt_cat(1DSF)`—page D-16
- `vsam_cat(1DSF)`—page D-22
- `maps(5DSF)`—page D-29

Glossary

access functions DataShare Facility functions that interpret the emulated volume structures.

BLKL DataShare Facility keyword for the mainframe BLKSIZE.

BLKSIZE The maximum length, in bytes, of a data block.

BMC Block Mux Channel.

CKD Count-Key-Data, which is used by mainframes.

conversion functions DataShare Facility functions that modify the information format or content before presentation to another host.

DASDMGR Direct Access Storage Device Manager, which are utilities used to configure storage on the StorEdge A7000 system.

data replication Reproduction of data on various platforms.

datasets Mainframe terminology for files.

DataShare capabilities Functions that are available through DataShare. That is, mainframe and open systems DataShare capabilities.

DSF DataShare Facility.

dsp1 and dsp2 Default names for the two subsystems on the A7000.

ESCON Enterprise System Connect, which is a method used to connect from a mainframe to an A7000 system.

esoteric Identifier by which a group of devices (usually tape and disk) are known. For example, you can use an esoteric name to request space from a specific pool of devices without having to know the volume serial number (VOLSER) of any specific device in the pool. Administrators can define security permissions at the esoteric level.

files Open system terminology for mainframe datasets.

fixed block A dataset structure in which blocks are a fixed length not exceeding 32K.

fixed record length Data record containing fixed-length fields.

HDA Head disk assembly.

IEBGENER IBM standard utility that copies files from one location to another. This utility is similar to the UNIX dd, cat, or cp commands.

ISO-9660 The standard on which CD-ROMs are based.

logical disks Virtual disks.

LRECL Logical record.

MVS Multiple Virtual Storage (mainframe operating system).

OEMI Other Equipment Manufacturer Interface.

open systems Hardware and software that permit easy access to various vendor products providing application portability, scalability, and interoperability using approved standards. An open system is the opposite of a proprietary or vendor-specific implementation.

phantom header/tail Certain open systems initiators require space at the beginning and/or end of each disk for header information; the required size and location (at the beginning or end of the disk) are system-specific. To prevent these initiators from writing their header information over the contents of a shared device, you must prepend a “phantom” header and/or append a “phantom” tail to the initiator’s view of the contents of the disk and store the initiator’s header information there. This area is always read/write and is independent of the access rights of the data shared region.

presentation functions DataShare Facility functions that enable delivery of accessed information to a specific emulation.

QSAM Queued Sequential Access Method (mainframe access method).

quiescent No reads or writes being performed, and cache has been destaged to disk.

record format Fixed, fixed block, variable block, and variable block spanned.

SCSI Target Emulation Hardware and software that allow an A7000 system to appear to open system initiators as one or more SCSI target disks.

SIMCKD Simulation of Count-Key-Data.

STE SCSI Target Emulation.

virtual disk The view provided by STE of the SCSI target disk from a SCSI initiator’s perspective.

VM Virtual Machine (mainframe operating system).

VOLSER Volume serial number used by the mainframe operating system.

volume Mainframe terminology for a logical disk. From a mainframe perspective, it is a logical representation of a physical address.

VSE Virtual Storage Extended (mainframe operating system).

VTOC Volume Table of Contents (mainframe directory).

Index

A

access functions, GL-1
allocation
  storage, 3-6
applications
  DataShare Facility, 5-1
  eliminating data replication / movement, 5-2
  examples, 6-1
    mainframe DSF capability, 6-2
  function consolidation, 5-5
architecture
  DataShare Facility, 2-3, 2-4
  illustration, 2-2
  StorEdge A7000, 2-2

B

best practices, 6-1
bidirectional information exchange, 1-8
  illustration, 1-8

C

cache synchronization, 4-4
concurrent access
  mainframe, 4-2
  open systems, 4-9
configuration
  A7000, 3-4
  DataShare Facility, 2-5
  dsf.cf file, A-2
  mainframe capability, 3-4
  open systems, 3-5
  open systems capability, 3-8
consolidation, function, 5-5
conversion functions, GL-1

D

DASD Tools, 1-5
  commands, 1-5
  UNIX Man Pages, C-1
  VSE limitations, 4-7
  Windows NT Man Pages, D-1
DASDMGR, GL-1
data
  access functions, 2-3
  capacity, B-1
  compression, 4-3
  conversion / processing functions, 2-3
  extraction, 5-4
  presentation functions, 2-3
  replication and movement
    eliminating, 5-2
  transfer, 5-4
dataset characteristics
  open systems, 4-9
datasets
  VSE, 4-5
DataShare Facility
  applications, 5-1
  architecture illustration, 2-4
  configuration, 2-5
  function offload, 5-5
  mainframe, 1-3

overview, 1-2
disk size, virtual, 3-2
dsf.cf file, A-1

E

examples
applications, 6-1
extended storage managers, 3-2

F

failover, 3-2
function
consolidation, 5-5
offload, 5-5
offload illustration, 5-6
functions
access, GL-1
conversion, GL-1
data access, 2-3
data conversion / processing, 2-3
data presentation, 2-3
presentation, GL-2

H

high availibilty, 3-2
HP 9000 / HP_UX platform issues, 7-2

I

information exchange
bidirectional, 1-8
Intel / Windows NT platform issues, 7-2
ISO-9660, GL-2

L

limitations of VSE, 4-5

M

mainframe capability
administration considerations, 4-2
application examples, 6-2
calculating data capacity, B-2
configuration, 3-4
determining volumes, 3-3
illustration, 1-4
overview, 1-3
planning, 3-3
setup, 5-7
usage, 5-4, 5-7
mainframe considerations
concurrent access, 4-2
multivolume sets, 4-3
MVS dataset, 4-2
performance, 4-3
software data compression, 4-3
synchronization, 4-4
system managed storage (SMS), 4-4
volume characteristics, 4-4
VSE limitations, 4-5
man pages
DASD tools (UNIX), C-1
DASD Tools (Windows NT), D-1
multivolume sets
mainframe, 4-3
MVS, 4-2
MVS dataset
mainframe
datasets, 4-2

O

offload, function, 5-5
open systems capability
calculating data capacity, B-3
concurrent transfer areas, 3-7
configuration, 3-8
illustration, 1-7
overview, 1-6
planning, 3-7
storage allocation, 3-6
usage, 5-7
open systems considerations
concurrent access, 4-9
dataset characteristics, 4-9

mainframe actions, 4-9
synchronization, 4-10
volume characteristics, 4-10
overview
DataShare Facility, 1-2
mainframe capability, 1-3
open systems capability, 1-6

P

performance
mainframe, 4-3
phantom
header, GL-2
tail, GL-2
planning
mainframe capability, 3-3
open systems capability, 3-7
platform issues
HP 9000 / HP_UX, 7-2
Intel / Windows NT, 7-2
RS6000 / AIX, 7-3
Sun, 7-3
presentation functions, GL-2

R

RS6000 / AIX platform issues, 7-3

S

SCSI Target Emulation, GL-2
and DataShare Facility, 2-2
SIMCKD, GL-2
and DataShare Facility, 2-2
STE, GL-2
storage
allocation
and DataShare Facility, 3-6
virtual disk size, 3-2
extended
managers, 3-2
Sun platform issues, 7-3
synchronization
cache, 4-4
mainframe, 4-4
open systems, 4-10
system managed storage (SMS)
mainframe, 4-4

U

usage
mainframe capability, 5-4, 5-7
open systems capability, 5-7

V

virtual disk size, 3-2
volume characteristics
mainframe, 4-4
open systems, 4-10
VSE limitations, 4-5
AF=CKD, 4-6
AF=RAWRO, 4-6
AF=RAWRW, 4-6
DASD Tools, 4-7
database issues, 4-8
dataset characteristics, 4-5
mainframe issues, 4-7

Y

Y2K compliance, 1-3
year 2000 compliance, 1-3