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

THE NETWORK IS THE COMPUTER™

A Sun Microsystems, Inc. Business
901 San Antonio Road
Palo Alto, CA 94303-4900 USA
1 650 960-1300 fax 1 650 969-9131

Part No.: 805-4366-10
Revision B, June 1998

Send comments about this document to: `smcc-docs@sun.com`

Contents

Code Samples

Figures

Tables

Preface

The *DataShare Facility System Administrator's & User's Guide* describes the Sun™ StorEdge™ DataShare™ Facility 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 and write 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
- Backup/restore DataShare capability

In addition, we offer a separate product that when installed on the open system host enhances the DSF capability and broadens the types of applications served:

- Mainframe DataShare Facility

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 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 considerations for application users to keep in mind when using DSF.

Chapter 8 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 printed copies of the A7000 online man pages.

Appendix B contains a sample dsf.cf file.

Appendix C contains information detailing storage sizing.

Appendix D contains mainframe JCL examples for the backup / restore DataShare capability.

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.

See 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 [ /dev/rdsk/dsfn ] ] [ -f` *filename*]
`com(n)`	The form `command(number)`, where the number in parentheses ranges from 1 through 8 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
Installation	*DataShare Facility Installation Notes*	online
Reference	*DataShare Facility Quick Reference*	805-4367
Reference	*Direct Access Storage Device (DASD) Manager User's Guide*	805-4884
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 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:

smcc-docs@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

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. The DataShare Facility 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 the DataShare Facility, 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 (e.g., backup/restore) can be consolidated into the environment where the function is accomplished most efficiently and effectively.

The following DataShare Facility 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

DataShare Facility 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.
- Backup / restore DataShare capability (BRDSF) to read open system raw disks, back up the volume, and then restore the volume.

FIGURE 1-1 illustrates a typical environment that may benefit from the DataShare Facility.

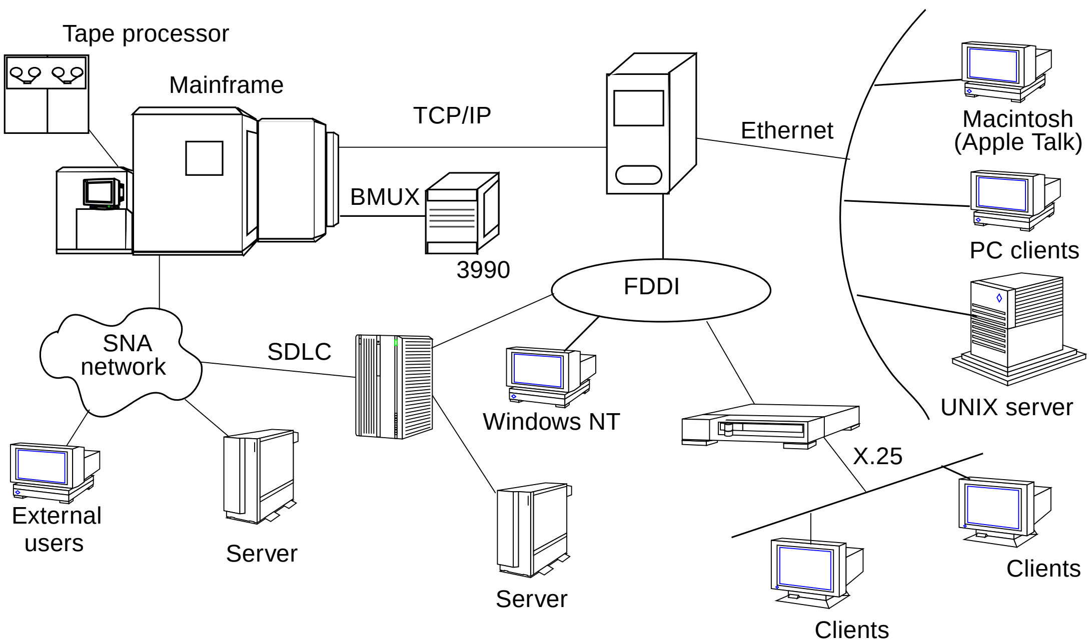

FIGURE 1-1 An Environment That Would Benefit From DataShare Facility

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) 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.

FIGURE 1-2 illustrates the mainframe DataShare capability.

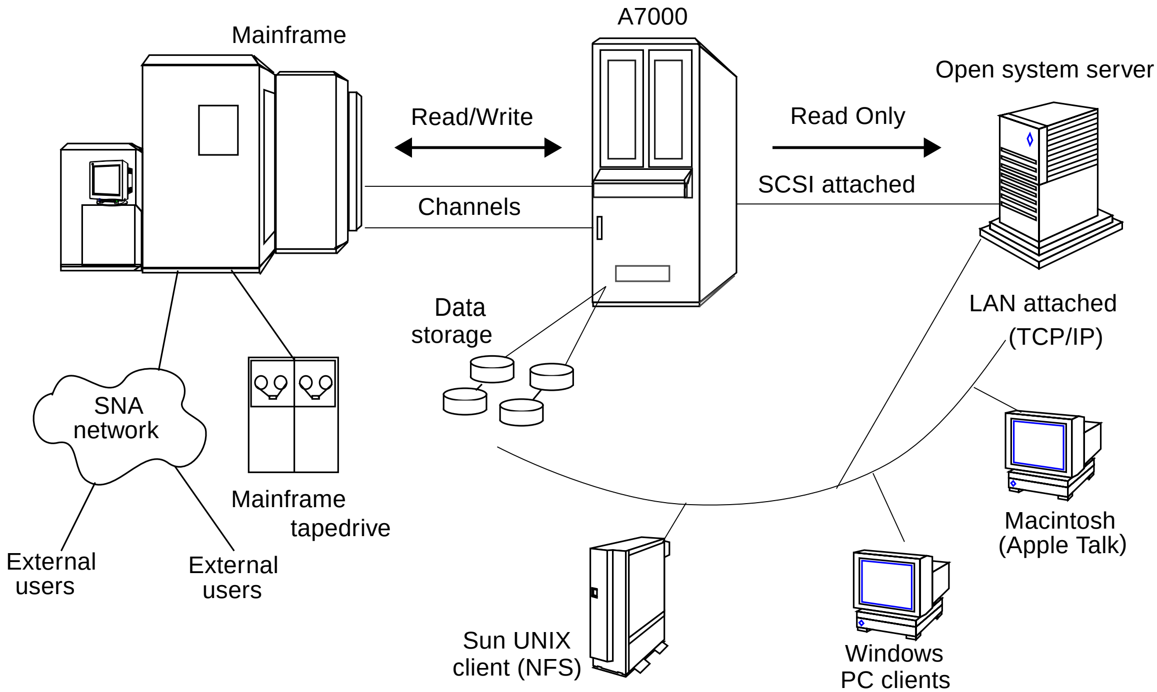

FIGURE 1-2 Mainframe DataShare Capability

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.

Alternative Access of Mainframe Volumes to Open System Hosts Using the Mainframe DataShare Facility Product

There are also unique tools available for the open system host (provided by a separate product) that enable a mainframe volume to be accessed and that provide a stream of data to an application.

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

In this case, the qualified SCSI-attached open system hosts have read access to MVS physical-sequential fixed-block and variable block QSAM datasets through SCSI Target Emulation (STE). Writes to other datasets on the same mainframe volume can occur concurrently but performance is impacted.

The tools provide the following:

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

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 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.

FIGURE 1-3 illustrates the open systems DataShare capability.

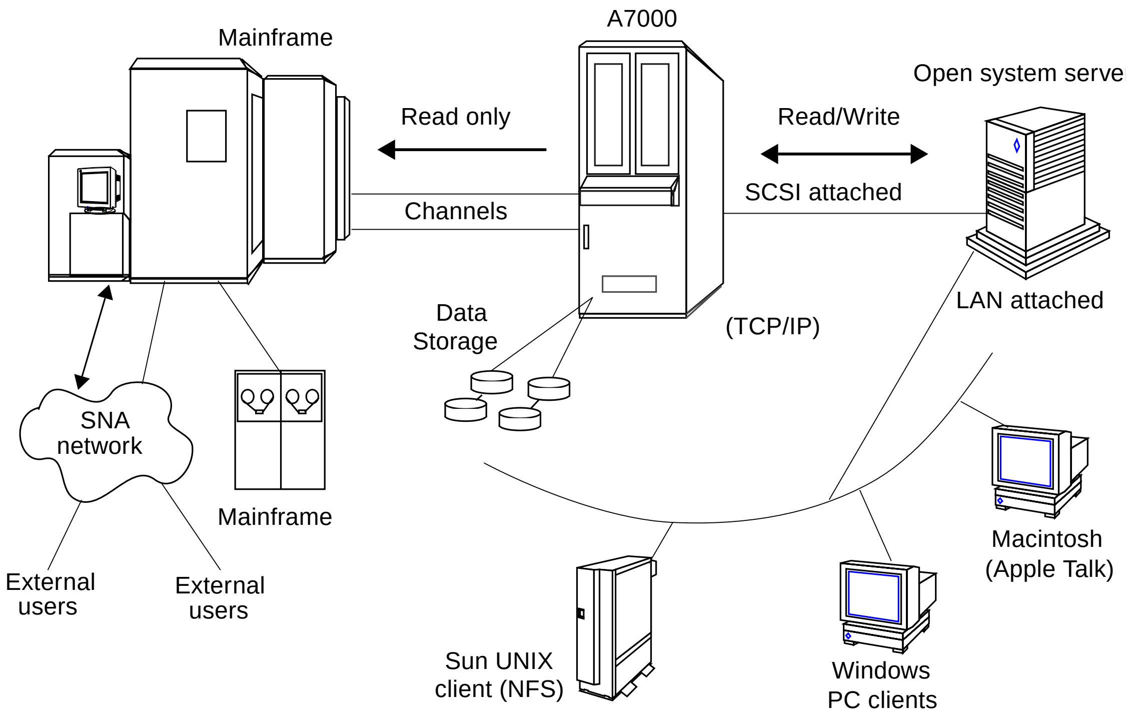

FIGURE 1-3 Open Systems DataShare Capability

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.

FIGURE 1-4 illustrates bidirectional information exchange.

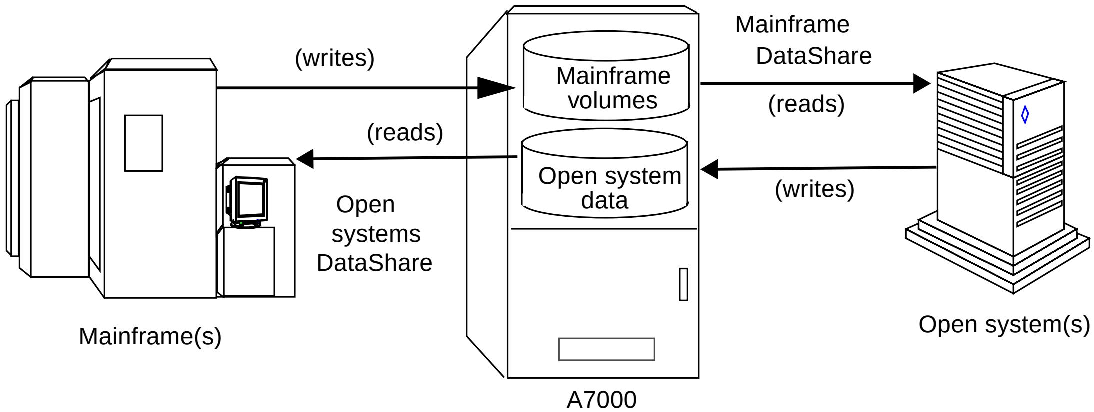

FIGURE 1-4 Bidirectional Data Exchange

Backup / Restore DataShare Capability

The backup / restore DataShare capability allows you to apply existing mainframe backup / restore software, peripherals, and procedures directly to A7000 resident nonmainframe volumes. The mainframe can back up mainframe volumes as well as open system volumes.

A7000 resident open system and network volumes appear to the MVS mainframes as though the entire volume is a read / write physical-sequential fixed-record / blocked (QSAM) dataset within one or more 3380-K or 3390-3 volumes. The VOLSER and dataset names representing each volume are preconfigured on the A7000 processor at initialization. You can use any MVS mainframe backup / restore product that supports 3380-K or 3390-3 sequential datasets to back up and restore these volumes without modification. For example, Fast Dump Restore (FDR), Data Facility Data Set Services (DFDSS), and IEBGENR.

Note – Only full volume backup and restore are supported at this time.

FIGURE 1-5 illustrates the backup / restore DataShare capability.

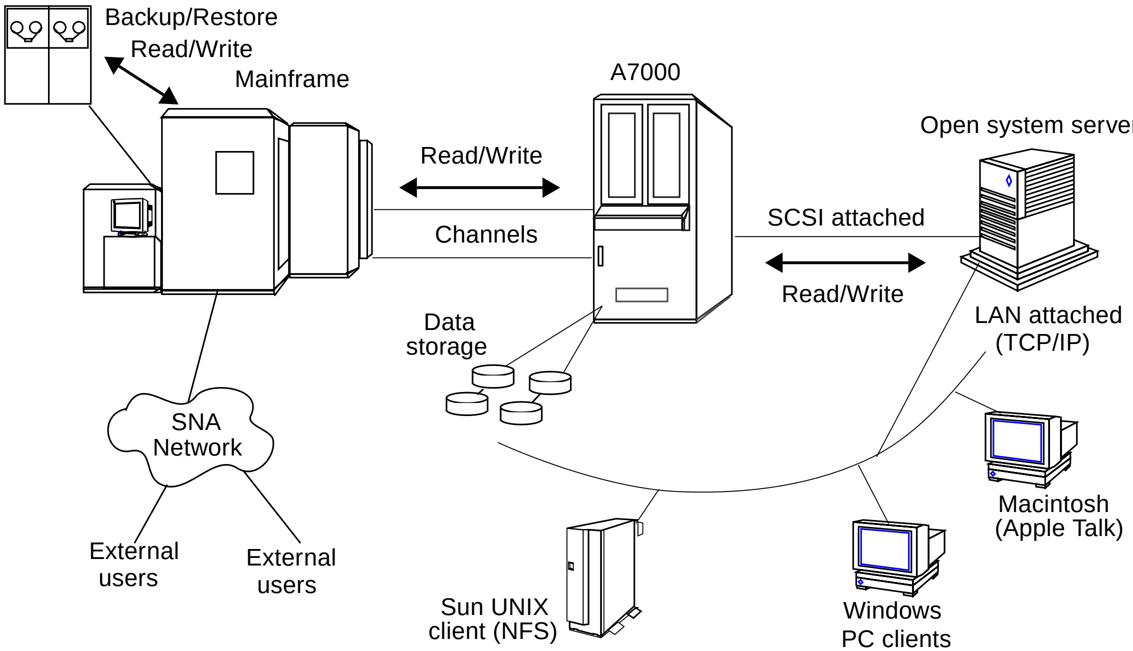

FIGURE 1-5 Backup/Restore DataShare Capability

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

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.

FIGURE 2-1 illustrates the product architecture and the relationship between the cache, HDA pool, and SIMCKD and STE emulation facilities.

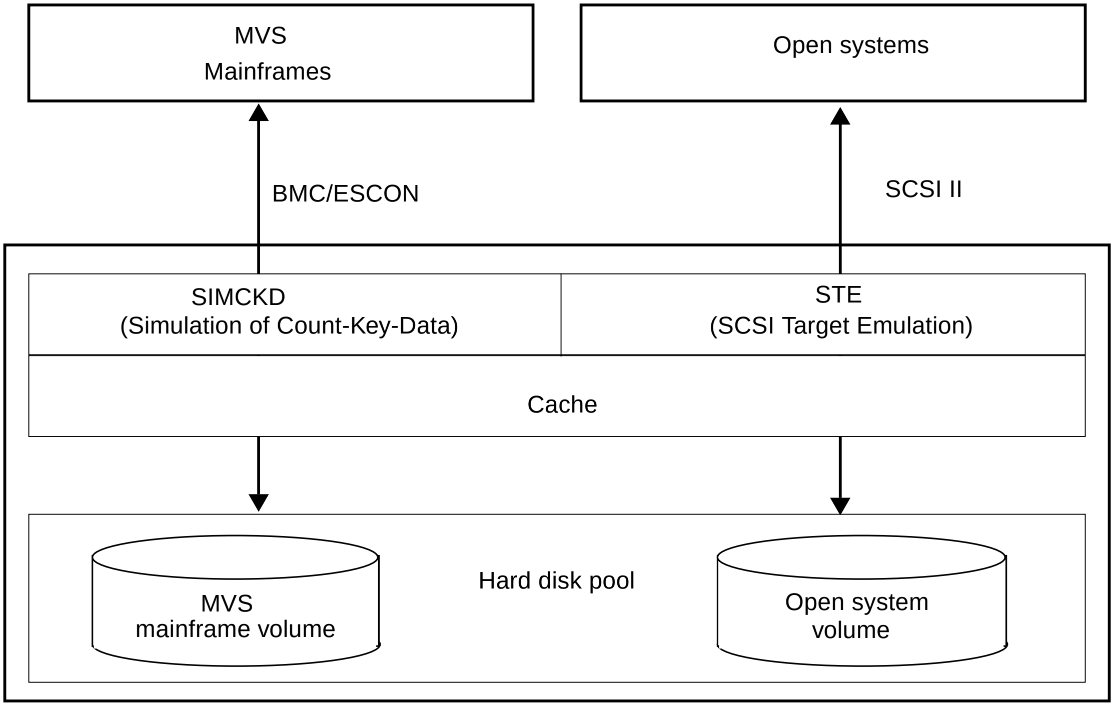

FIGURE 2-1 Product Architecture

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 layer is then used to make data stored through CKD 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.

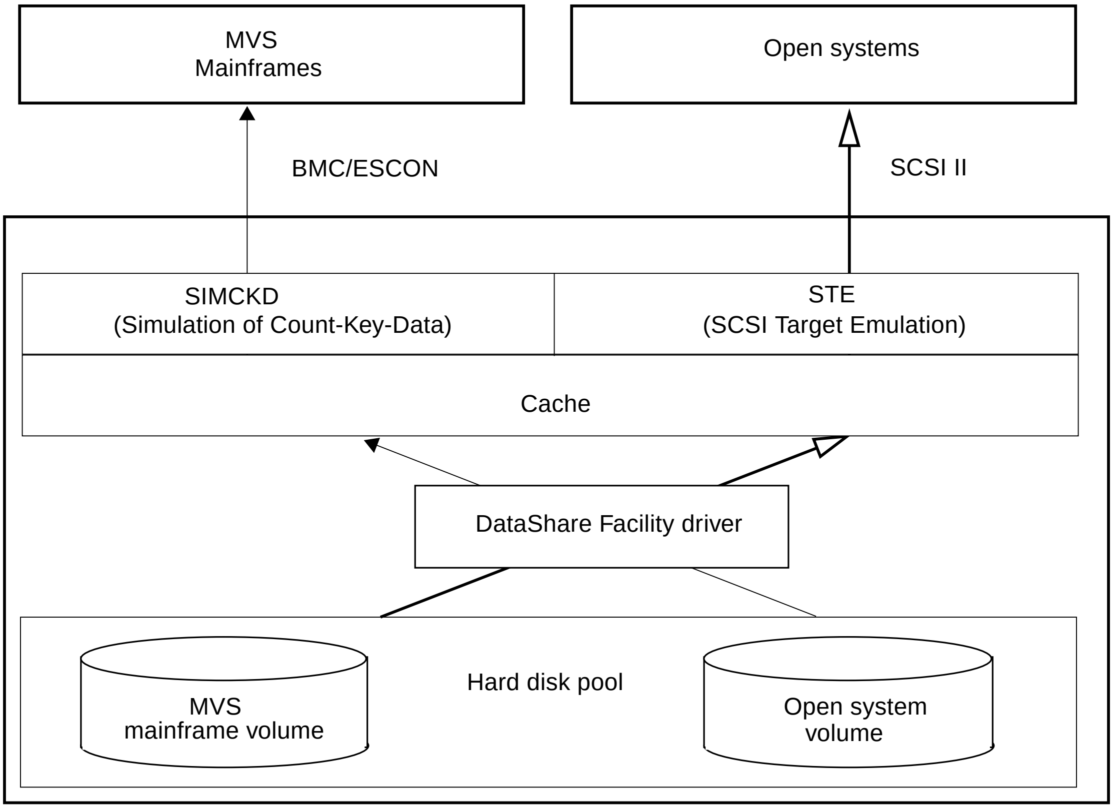

FIGURE 2-2 DataShare Facility Architecture

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 CKD MVS volumes
  - RAW, read only or read/write 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 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
  - One-to-one custom map character conversion
  - Fixed record to carriage return / line feed conversion
  - Fixed record to 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 2-3 illustrates a typical mainframe DSF capability architecture.

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

FIGURE 2-5 illustrates a typical backup / restore DSF capability architecture.

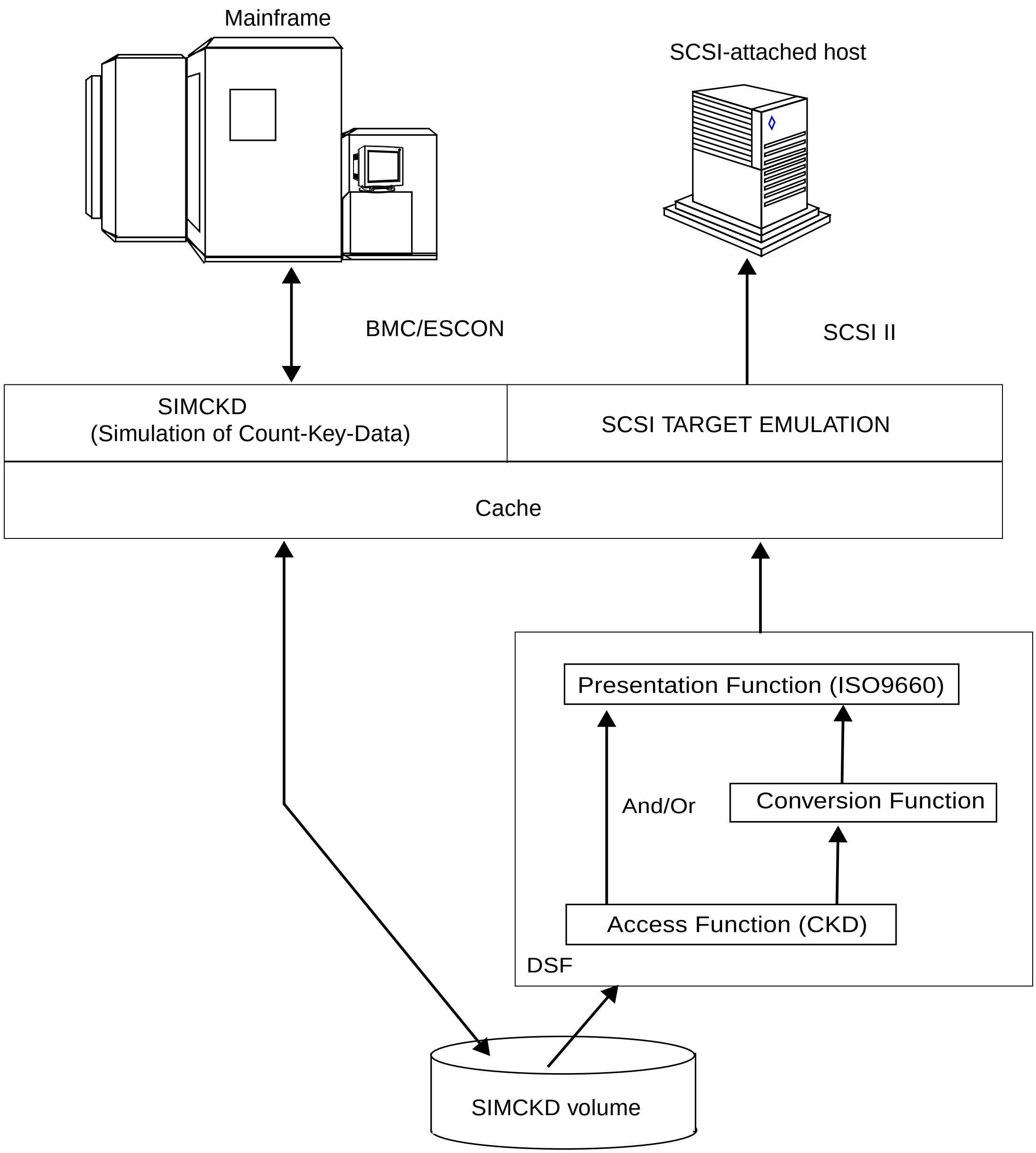

FIGURE 2-3 Mainframe DSF Capability Architecture

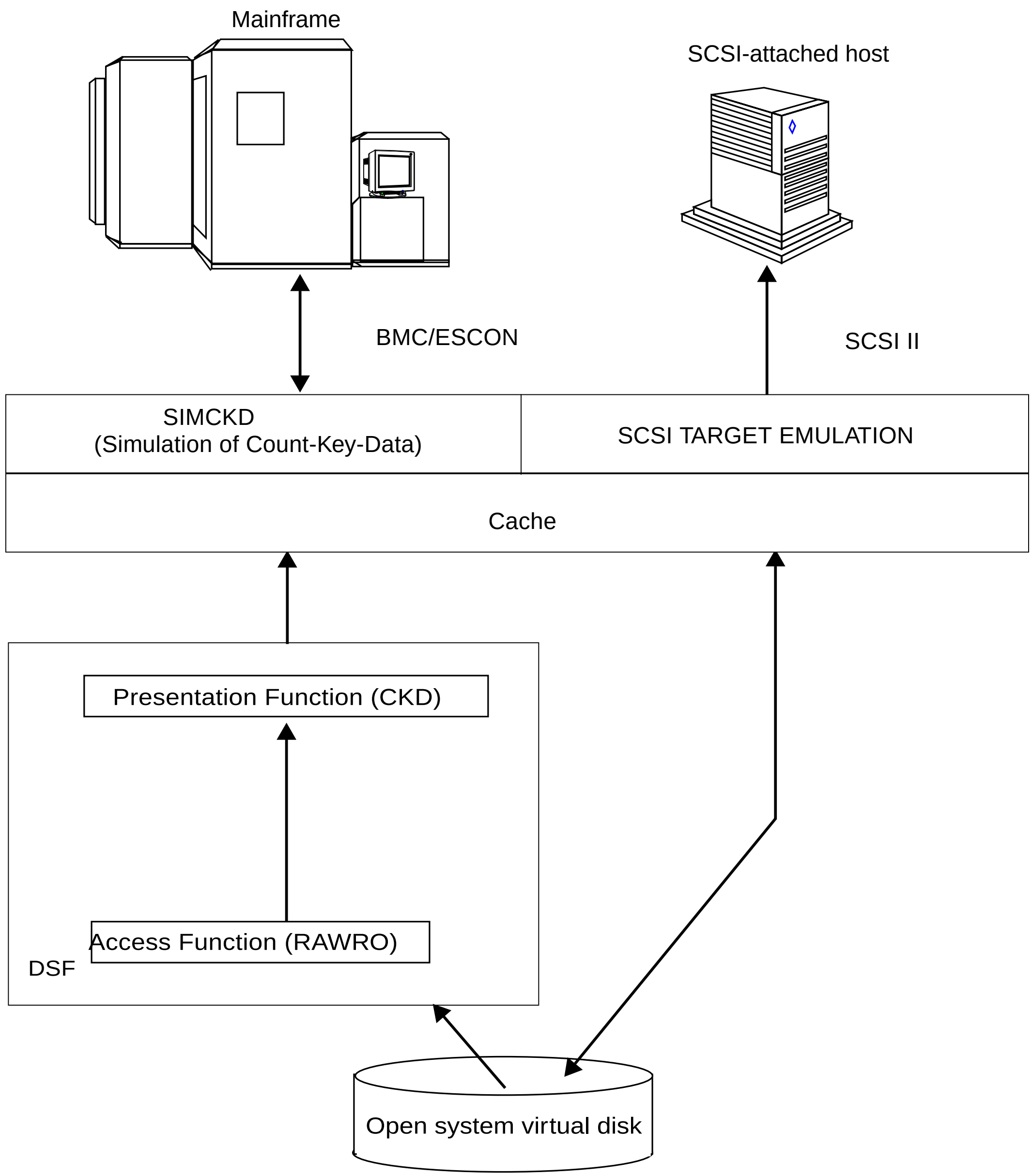

FIGURE 2-4 Open System DSF Capability Architecture

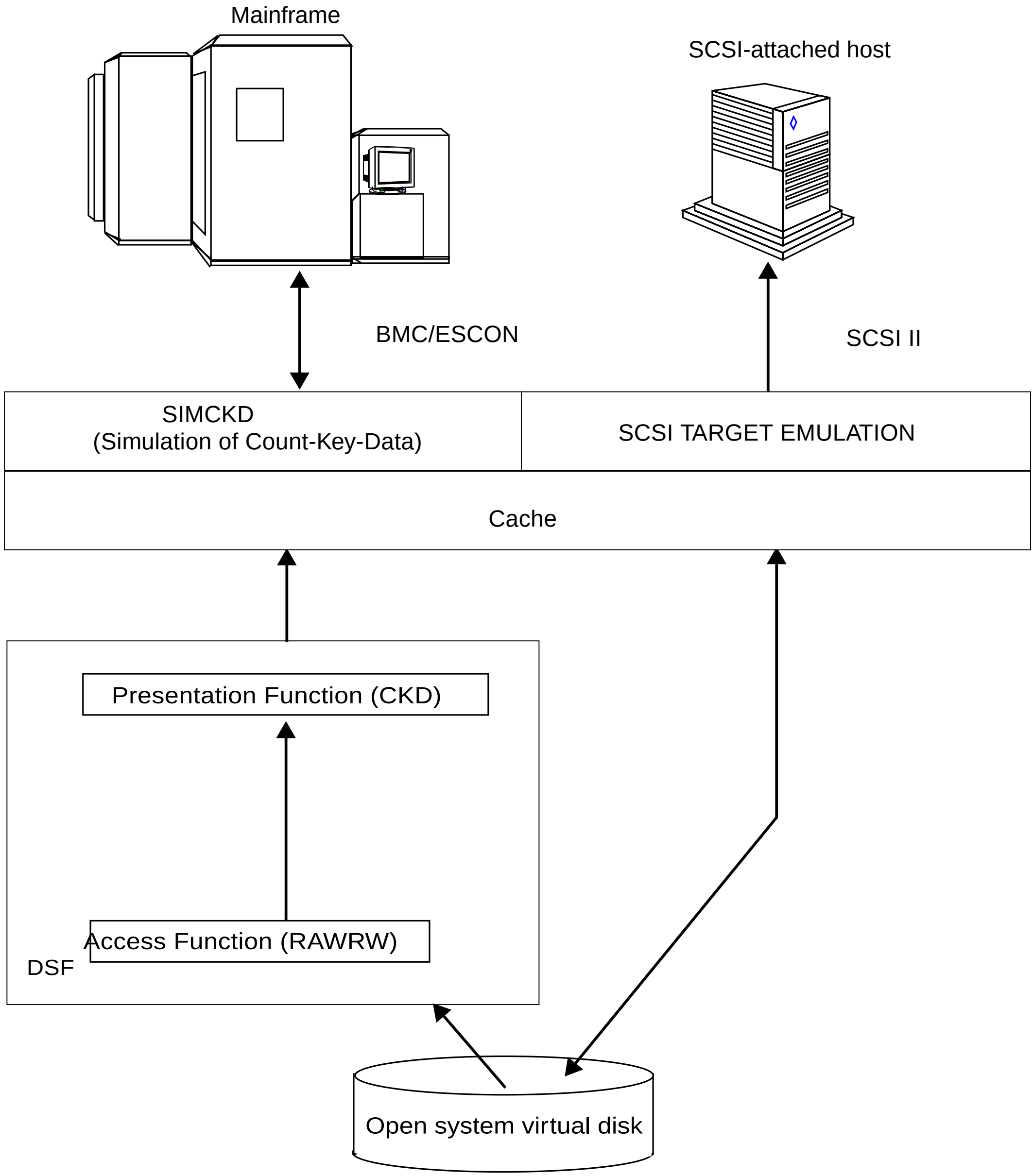

FIGURE 2-5 Backup/Restore DSF Capability Architecture

CHAPTER 3

Planning and Configuration

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

DataShare Facility Considerations

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

Storage Allocation and DataShare Facility

DASD Manager's default setup of the open systems and backup/restore capabilities 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.

Optimum mainframe sharing capacity is achieved when a multiple of the blocksize fits the geometry of the disk being presented, either 3380-K or 3390-3.

Optimum performance is achieved when the block size is large.

For open system and backup/restore, 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 backup/restore DataShare capability, use the defaults to maximize capacity and performance. Multiple mainframe volumes can be allocated for open system logical disks of greater capacity than can fit in a single mainframe volume.

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

Figure 3-1 illustrates the default 3390 track characteristics for a backup/restore DSF capability dataset. There are two blocks of 27K bytes each. Within each block there is one 27K byte logical record.

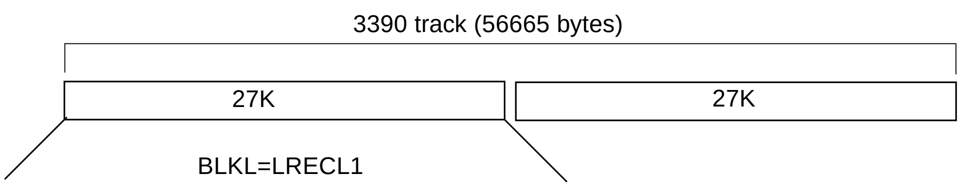

FIGURE 3-1 3390 Backup/Restore Capability Dataset Characteristics

Figure 3-2 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 80 and the blocksize is 27600, there are 345 80 byte records.

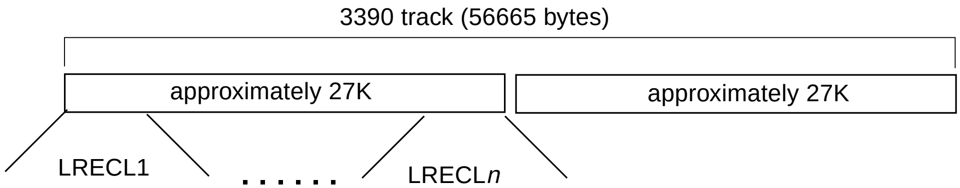

FIGURE 3-2 3390 Open Systems Capability Dataset Characteristics

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.

   Refer to Appendix C 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 for the MFDSF volumes, excluding them from other common esoterics.**
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 unless the volume will be used for small datasets.**

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 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. Refer to "Multivolume Datasets" on page 4-2 and"System Managed Storage (SMS)" on page 4-3.**

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.

▼ 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:**

   - 1.2 Gbytes if a 3380-K volume is used.
   - 2.2 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.

   Refer to Appendix C 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. Refer to Chapter 8 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 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.

Backup/Restore DataShare Capability

The following sections describe planning, setup, and usage of the backup/restore DataShare capabilities (BRDSF) for an open system disk on the A7000.

▼ To Determine the BRDSF Volumes

The following groups may be involved:

- Mainframe system programmers
- Open system administrators

1. **Determine which open system disks must be backed up and record the size of each.**

2. **Determine the 3390-3 volumes required per open system disk.**

3. **Use this** *rule of thumb* **for determining approximate capacity per 3390-3 volume:**

   - 2.5 Gbytes for each 3390-3 volume.

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

   Refer to Appendix C for detailed data storage calculation information.

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

   number of mf volumes = sum of mf volumes required for each open system disk

---

Note – Sixteen is the maximum number of DSF entries (in the `dsf.cf` file), which reference an individual storage device.

Caution – Take existing A7000 resident mainframe volumes offline while the new devices are brought online.

▼ To Configure the A7000

This procedure is for the person doing the A7000 configuration.

Prerequisite: A7000 storage and cable connections to the open system host(s) are already established.

1. **Configure the DataShare Facility devices to mainframe hosts (open system virtual disk, VOLSER, dataset name, default LRECL/BLKL (BLKSIZE)), and set access function and read/write raw access (RAWRW) CKD presentation function.**

   No conversion functions are specified.

2. **Start the DSF and Simulation of Count-Key-Data (SIMCKD) components.**

Open Systems Configuration

This was completed when the open system virtual disks were allocated.

Mainframe Configuration

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

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

CHAPTER **4**

Administration Considerations

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

Mainframe DataShare Capability (MFDSF)

The following subsections contain useful information to keep in mind when using the mainframe DataShare capabilities of the DataShare Facility. Refer to Chapter 7 for additional details.

Concurrent Access

- Multiple open system hosts can simultaneously access a DataShare Facility 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.

Dataset Characteristics - MVS

- When the data presentation function is ISO-9660, dataset names that are longer than 30 characters are truncated to 29 characters and an underscore character (_) is added to the beginning of the 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.

  In this example, the two dataset names are different. Because of the truncation rules, the characters at the beginning of the names are truncated creating two dataset names that are identical.

  ```
  BPAUL.TEST01.DATASET.FOR.DSF.TESTING.DATA.1
  ARIEL.TEST02.DATASET.FOR.DSF.TESTING.DATA.1
  ```

  Both are truncated to:

  ```
  _ATASET_FOR_DSF_TESTING_DATA_1
  _ATASET_FOR_DSF_TESTING_DATA_1
  ```

Note – Ensure that multiple dataset names will not be the same if truncated, as only one of the datasets will be visible from the ISO-9660 presentation.

- Only Physical Sequential (QSAM) datasets are presented to the open system host (either fixed record / unblocked or fixed record / blocked).

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.
- 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 CD-ROM image, an open system remount *must* occur; otherwise, data read by applications on the open system may not be current because of the CD-ROM image change and local caching. Refer to Chapter 8.

System Managed Storage (SMS)

- 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.

Open Systems and Backup/Restore Capabilities

The following subsections contain useful information to keep in mind when using the open systems and backup/restore capabilities of the DataShare Facility.

Additional Mainframe Considerations

- The mainframe may require specific actions (for example, some mainframe backup utilities may require a generation data group (GDG) entry to enable a backup).
- Backup and restore volumes are read/write from the mainframe. All restores must be coordinated with the open system host administrator.
- 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.

Concurrent Access

- To ensure data consistency for backup/restore, the open system host must be write-quiescent during the backup and any writes that are pending in the open system cache must be flushed to the A7000 disk.

Dataset Characteristics

Dataset	Description
Dataset Names	Dataset names are preconfigured on the A7000.
Size	The 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 and 3390 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

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 mainframe network is in place
- The transfer window is a concern because of the quantity and reliability of transfer

Eliminating Data Replication / 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 the DataShare Facility and using the DataShare Facility. DataShare Facility eliminates the transfer, allowing more detailed and frequent data warehouse updates.

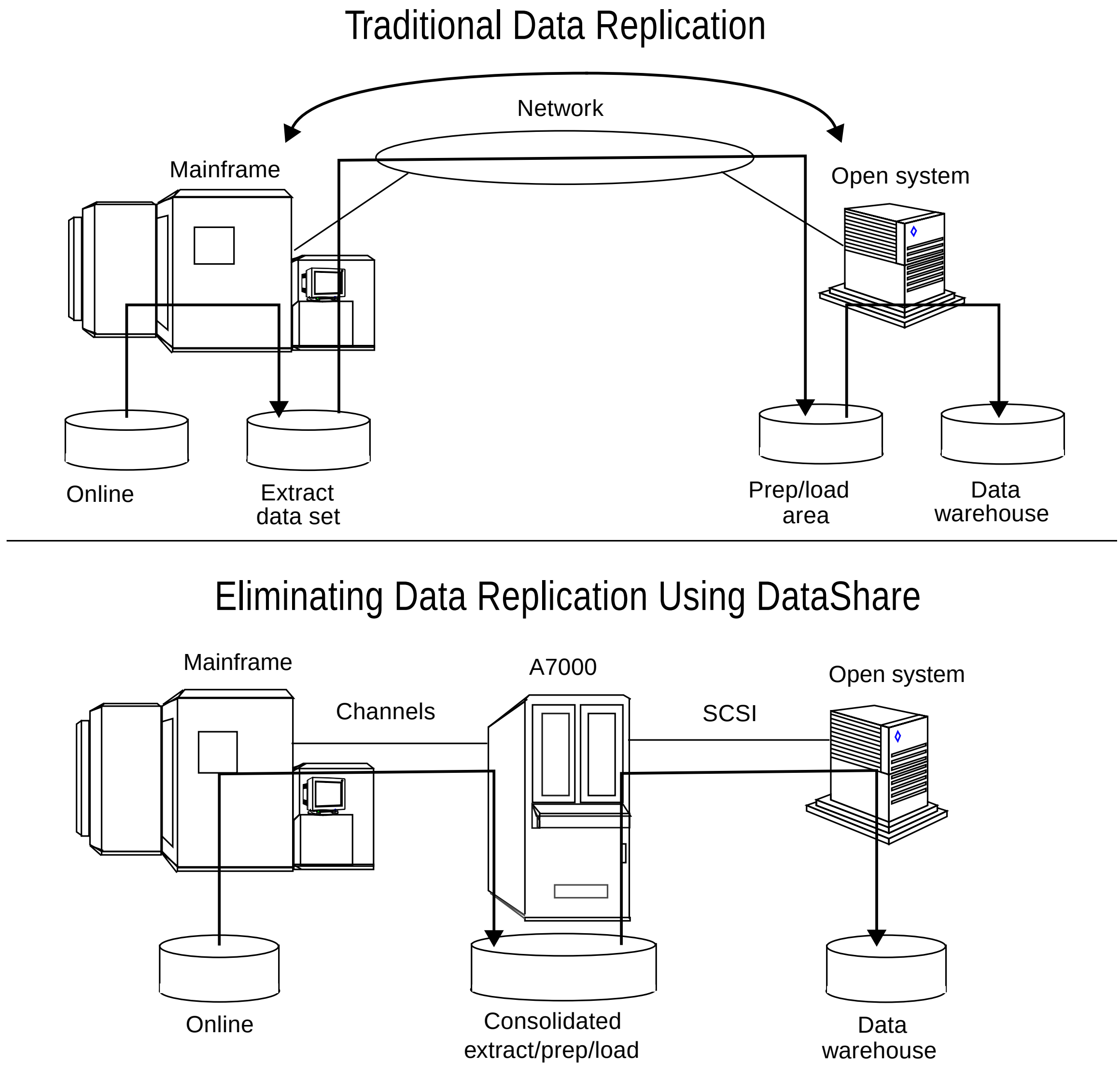

FIGURE 5-1 Eliminating Data Replication With DataShare Facility

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, we recommend that mainframe I/O activity for a shared volume 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 CD-ROM image (unmount, mount), or on an NT system, refresh the CD-ROM 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.

The following subsection describes function consolidation in the backup/restore and disaster recovery area.

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

- Printing
- Internet servers
- Network access
- World Wide Web access

Backup/Restore and Disaster Recovery

Standard storage maintenance and disaster recovery preparation are examples of function duplication. Each system has dedicated:

- Backup software and associated license fees
- Peripheral devices (tapes, transports, etc.)
- Procedures and processes
- Training costs
- Operators
- Disaster plans

FIGURE 5-2 illustrates the differences between traditional backup/restore/disaster recovery and backup/restore/disaster recovery using the DataShare Facility.

The A7000 with the Datashare Facility provides direct, cross platform access to shared storage resources, accomplishing a specified function efficiently and effectively.

The backup / restore DataShare capability consolidates this function on the mainframe on behalf of all attached systems. In FIGURE 5-2 on page 5-7, open system platforms requiring backup are configured for the backup / restore DataShare capability (read / write) for access by the mainframe. These full volumes are then accessed as mainframe 3390-3 or 3380-K datasets for backup operations.

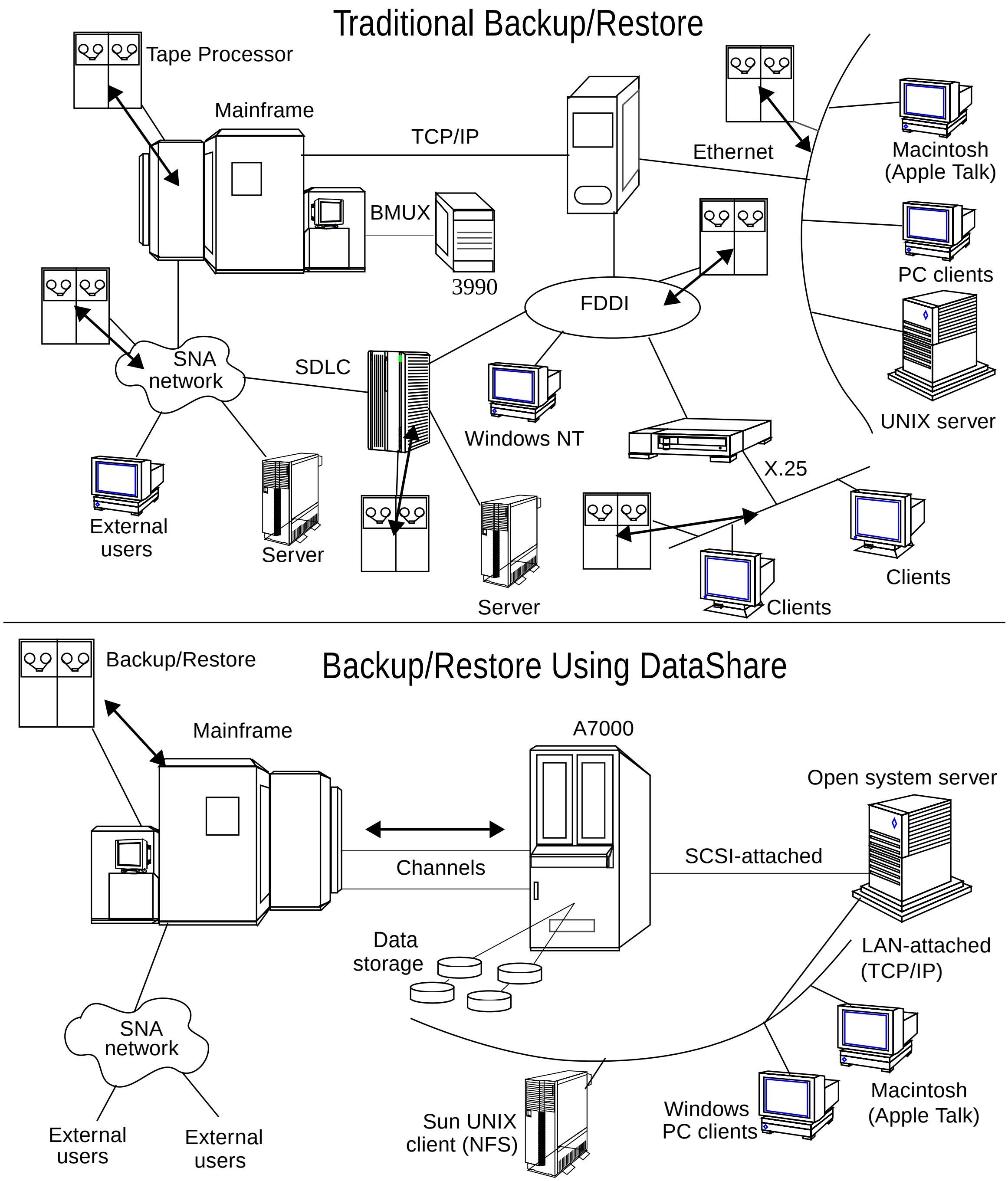

FIGURE 5-2 Backup/Restore Without and With DataShare Facility

Using the Backup/Restore Capability

Shared open system volumes are accessed by the mainframe as nonshared mainframe volumes. One sequential dataset containing all of the open system's information is presented to the mainframe.

▼ To Do Backups

This procedure is the suggested method for doing backups.

1. **To ensure data consistency, the open system virtual disk being backed up must be write-quiescent from the open system host, and all writes from the open system cache must be flushed to the A7000 virtual disk.**
   - On open systems running UNIX where the entire A7000 logical disk contains one or more file systems, unmount all of the file systems.
   - On the Windows NT system:

   a. **Select the drive to be backed up:**

   ```
   administrative_tools=>disk_administrator
   ```

   b. **Choose Do not assign drive letter from the tools to make the drive quiescent and unavailable to other users.**

   ```
   tools=>Assign_Drive_Letter (on NT 4.0)
   or
   tools=>Drive_Letter... (on NT 3.51)
   ```

2. **Perform the backup on the mainframe using standard MVS backup utilities.**

   Refer to the JCL examples in Appendix D.

3. **After the backup is complete, resume normal open system operation.**
   - On open systems running UNIX, mount the file systems.
   - On the Windows NT system:
     - If the NT system was shut down, reboot.

- Otherwise, return to the tools, select the drive that was backed up, and reassign the previous drive letter. This takes immediate effect.

```
administrative_tools=>disk_administrator
tools=>Assign_Drive_Letter (on NT 4.0)
or
tools=>Drive_Letter... (on NT 3.51)
```

▼ To Do Restores

This procedure is the suggested method for doing restores.

1. **Ensure that the open system virtual disk is quiescent from the open system host and is not mounted or being accessed.**

2. **Ensure that the target virtual disk has been formatted/partitioned identically to the one from which the backup was made.**
   - On open systems running UNIX:
     - Use the open systems host's disk administration tools. Refer to the open system vendor documentation.
     - On UNIX systems where the entire A7000 logical disk contains one or more file systems, *unmount* all of the file systems.
   - On the Windows NT system:

     All the data is laid down exactly as it was on the source disk. If changes are required, use the partition/format options within the administrative tools.

3. **Ensure that the target virtual disk is quiescent from the open system host and information is flushed to the virtual disk.**
   - On open systems running UNIX where the entire A7000 logical disk contains one or more file systems, unmount all of the file systems.
   - On the Windows NT system:

   a. **Select the drive to be restored:**

```
administrative_tools=>disk_administrator
```

   b. **Choose `Do not assign drive letter` from the tools to make the drive quiescent and unavailable to other users.**

```
tools=>Assign_Drive_Letter (on NT 4.0)
or
tools=>Drive_Letter... (on NT 3.51)
```

4. **Perform the restore on the mainframe using standard MVS backup utilities.**

   Refer to the JCL examples in Appendix D.

5. **Resume normal operation on the open system.**

   - On open systems running UNIX, mount the file systems.
   - On the Windows NT system:
     - If the NT was shut down, reboot.
     - Otherwise, return to the tools, select the drive that was backed up, and reassign the previous drive letter. This takes immediate effect..

```
administrative_tools=>disk_administrator
tools=>Assign_Drive_Letter (on NT 4.0)
or
tools=>Drive_Letter... (on NT 3.51)
```

Note – If the restore was made to a disk other than the original disk, the NT's disk administrator indicates that the restored disk does not have a signature. Reply **Yes** when asked if a signature should be written to that disk. If the original disk is still connected, you now have two disks with the same volume level and contents.

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.

FIGURE 5-3 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.

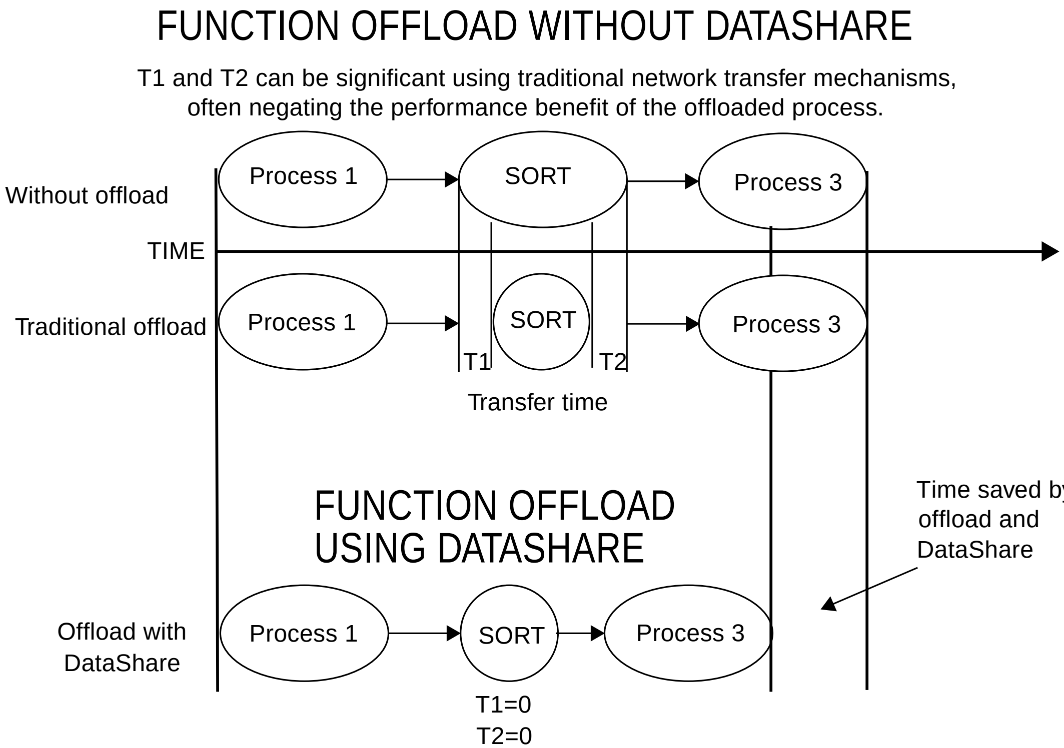

FIGURE 5-3 Function Offload Without and With DataShare Facility

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. Refer to the `dsf.cf(4DSF)` manual page 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

Application Examples

This chapter contains application examples of specific configurations that will help you understand how to plan and configure for DataShare Facility (DSF).

Example — Mainframe DSF Capability (MFDSF)

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

User Considerations

This chapter contains considerations that are important when using the DataShare Facility.

Mainframe DataShare Capability

The following subsections contain useful information to keep in mind when using the mainframe DataShare capabilities of the DataShare Facility.

Concurrent Access

- Multiple hosts can simultaneously access a DataShare Facility device.
- Multiple readers on the same host can read a mainframe data shared volume or dataset.
- Do not perform mainframe writes to volumes that are being read by an open system host.

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 and an underscore character (_) is added to the beginning of the 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.
	
	In this example, the two dataset names are different. Because of the truncation rules, the characters at the beginning of the names are truncated creating two dataset names that are identical.
	
	`BPAUL.TEST01.DATASET.FOR.DSF.TESTING.DATA.1`
	`ARIEL.TEST02.DATASET.FOR.DSF.TESTING.DATA.1`
	
	Both are truncated to:
	`_ATASET_FOR_DSF_TESTING_DATA_1`
	`_ATASET_FOR_DSF_TESTING_DATA_1`
	
	Note: Ensure that multiple dataset names will not be the same if truncated, as only one of the datasets will be visible from the ISO-9660 presentation.

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.

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 table of contents (VTOC) that affects the CD-ROM image, the open system must be remounted; otherwise, data read by applications on the open system may not be current because of the CD-ROM image change and local caching. Refer to Chapter 8.

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.

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.

Open Systems and Backup / Restore DataShare Capabilities

The following subsections contain useful information to keep in mind when using the open systems and backup / restore capabilities of the DataShare Facility.

Additional Mainframe Considerations

- The mainframe may require specific actions (for example, some mainframe backup utilities may require a generation data group (GDG) entry to enable a backup).
- Backup and restore volumes are read / write from the mainframe. All restores must be coordinated with the open system host administrator.
- When the DataShare Facility provides an open system virtual volume to the mainframe, the volume contains one or more datasets and the volume free space is always zero.

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.
- To ensure data consistency for backup/restore, the open system host must be write-quiescent and all open system cache references flushed to the A7000 virtual disk before the backup or restore.

Dataset Characteristics

Dataset	Description
Dataset Names	Dataset names are preconfigured on the A7000.
Size	The 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 **8**

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

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 Gbyes. 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 Phantom Tail: 8192	Maximum file size is 2 Gbytes. Maximum file system size is unlimited. Maximum partition size is unlimited. Maximum disk size is unknown.

APPENDIX **A**

DSF Man Pages

This appendix contains hardcopies of the DataShare Facility man pages. The most recent versions of the man pages are online. The online *DataShare Facility Release Notes* specify if there is a more recent version than the one in this appendix.

Man Pages

- `dsfadmin(1DSF)`
- `dsf.cf(4DSF)`
- `maps(5DSF)`

APPENDIX **B**

Sample `dsf.cf` File

This appendix contains a sample DataShare Facility configuration file.

Example of a `dsf.cf` File

CODE EXAMPLE B-1 Sample `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 hexadecimal, examples dsf0,..9,A-F,10 etc..
# The maximum number of dsf devices over all subsystems 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.
```

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

```
#
# 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 subsystem 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 subsystem ID
#                                               (subsystem ID).  4 | 5
#                                               Must be specified for
#                                               multisubsystem
#                                               configurations, optional
#                                               for GATEWAY
#                                               configurations. If not
#                                               specified, ID is
#                                               displayed as 0.
#
# SLAVE=   3 | 2 or          none               Slave subsystem ID
#          5 | 4                                Don't specify for a
#                                               GATEWAY or if the dsf
#                                               device will not be used
#                                               during failover.  If
#                                               not specified, ID is
#                                               displayed as 0.
```

CODE EXAMPLE B-1 Sample dsf.cf File

```
#
# 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.
#
#          {RAWRO | RAWRW}                          RAWRO is read only disk
#                                                   data, RAWRW is read and
#                                                   write disk data, use
#                                                   RAWRO for OSDSF
#                                                   capability, use RAWRW
#                                                   for BRDSF capability
#                         (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, translation
#                                                   from lower to upper case
#                                                   is 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.
#                          [,LRECL=n]               Logical record length in
#                                                   bytes.
#
```

CODE EXAMPLE B-1 Sample dsf.cf File

```
#                            [,{SIZE=nnnn|NUMREC=n}])
#                                                  SIZE or NUMREC can be
#                                                  specified. The default
#                                                  is the size of partition
#                                                  remaining after
#                                                  adjusting for the START
#                                                  offset.
#                                                  SIZE, if specified is in
#                                                  sectors, and should
#                                                  never exceed the size
#                                                  that can be supported by
#                                                  the PF(see table below).
#                                                  NUMREC is the number of
#                                                  logical records.
#                                                  If NUMREC is specified,
#                                                  the SIZE is calculated,
#                                                  (based on BLKL & LRECL).
#
# CF=      _conversion_function_                   Conversion function (see
#                                                  below)
#
#   [(AFlist)] DELIM      [(a,b,...c)]             Delimiter characters
#                                                  (see below)  Default is
#                                                  <LF>, octal 012.
#
#   [(AFlist)] ETOA       [([MAP=/fullpath][,]     EBCDIC to ASCII
#                                                  conversion.  The MAP
#                                                  option can specify the
#                                                  full path to an
#                                                  alternate ETOA 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.
```

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

```
#
#
#         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.
#                                              Default depends on PF &
#                                              options.
#                         [,LRECL=n]           Logical record length
#                                              in bytes.  Overrides
#                                              volume level default.
#                                              Use only with AF=RAW*.
#
# Notes on AF functions and arguments:
# ===================================
#
# A dsf definition can specify multiple access functions.  The sum of
# all data from each AF is 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
#   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.
```

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

```
#   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 is calculated to respresent the entire disk
#   partition.
#
#
# Notes on CF functions and arguments:
# ===================================
#
# If 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 are to
#   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.
#
#   ETOA arguments:
#
#   the optional argument to the ETOA conversion functions must
#   be a full path to a 256 byte data file.  The value of each character
#   (byte) in the file represents the desired translated character
#   for that character value.  For instance, the converted value of
#   character value octal [000] appears 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 maps are from a proposed ANSI standard.
#   Several alternative maps are provided on the distribution.
```

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

```
#   These are documented in the maps(5DSF) manual page.
#
#
# Notes on PF functions and arguments:
# ===================================
#
#   CKD PF BLKL and RECL
#
#   If the BLKL and LRECL are not available from the Access
#   Function layer, the BLKL and LRECL for each dataset are 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
# Note: A maximum of 16 references to an individual logical disk can be made
# within the dsf.cf file. This limits the maximum open system virtual disks
# that can be backed up using the backup/restore DataShare capability to
# 40 Gbytes (16 * 2.5 Gbytes).
#
#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 \fB/etc/ste.cf\fR,
```

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

```
#   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
#
#
#
# Backup / Restore Datashare (BRDSF):
# ==================================
#
#   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=3380,VOLSER=UN14DE)
#   END
#
#   CKD presentation of an open system partition of 6 Gbytes(vp1,
#   12582912 sectors).
#   6 Gbytes 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=1767792)
#     PF=CKD (DTYPE=3390,VOLSER=UVP1P3)
#   END
```

CODE EXAMPLE B-1 Sample dsf.cf File

```
#   Specifying alternate values for BLKL and LRECL overrides these
#   defaults, and changes the track and volume capacity.
#
#
#   OR, REDIFINITION of dsf13 above, but this time
#   let the SIZE default on the last part of a multivolume backup set.
#   dsf13
#     MASTER=2 SLAVE=3
#     AF=RAWRW (DISK=/dev/rdsk/vp1,NAME=unix.data.vp1.part3,
#               START=10815120)
#     PF=CKD (DTYPE=3390,VOLSER=UVP1P3)
#   END
#
# 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
#       file[1,2] are entire partitions, and
#       file[3,4] are partial parts of a partition.
#
#   dsf22
#     MASTER=2 SLAVE=3
```

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

```
#     AF=RAWRO (DISK=/dev/rdsk/14d7,NAME=unix.data.file1,
#               BLKL=25600,LRECL=80)
#     AF=RAWRO (DISK=/dev/rdsk/14d8,NAME=unix.data.file2,
#               BLKL=8192,LRECL=1024)
#     AF=RAWRO (DISK=/dev/rdsk/14d9,NAME=unix.data.file3,
#               BLKL=31744,LRECL=124,NUMREC=500)
#     AF=RAWRO (DISK=/dev/rdsk/14da,NAME=demo.data.file1,
#               BLKL=20480,LRECL=20480,SIZE=120)
#     PF=CKD (DTYPE=3390,VOLSER=UNIXT1)
#   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
#
#END OF SAMPLES
```

APPENDIX **C**

Calculating Data Capacity

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

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 C-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 C-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

Backup/Restore 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.

TABLE C-3 Actual Data Capacity (Backup/Restore 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	22.5K	2	1749.46MB	= 1.708GB	= 3582900
3390-3*	3338	15	27K	2	2640.41MB	= 2.578GB	= 5407560

* indicates maximum capacity

APPENDIX **D**

JCL Examples

This appendix contains samples of JCL for doing backups and restores.

Restore Using DFDSS

```
//RSTM  JOB ,'BRUSER,CLASS=S,MSGCLASS=L,MSGLEVEL=(1,1)
//*
//**************************************************************
//***                                                        ***
//***  THIS JOB USES DFDSS TO DO A "LOGICAL" RESTORE BY      ***
//***  DSNAME.  THE "REPLACE" KEYWORD MUST BE USED TO WRITE  ***
//***   OVER THE DATA SET.                                   ***
//***                                                        ***
//***   IF FOR ANY REASON THE DATA SET HAS TO BE WRITTEN TO A ***
//***   DIFFERENT DASD VOLUME THEN DO THE FOLLOWING:         ***
//***  1.-UNCATALOG ORIGINAL DATA SET IF CATALOGED.         ***
//***  2.-ON THE OPEN SYSTEMS SIDE ALLOCATE THE SAME DATA SET ***
//***       ON ANOTHER VOLUME.                               ***
//***  3.-DO THE LOGICAL RESTORE.                            ***
//***                                                        ***
//***************************************************************
//***       RESTORE MEDIUM SIZE DSNAME FROM MF DASD          ***
//***                                                        ***
//***        DSNAME = HOST1.MEDIUM.MASTER                    ***
//***************************************************************
```

```
//*
//STEP1    EXEC PGM=ADRDSSU,REGION=5M,PARM='UTILMSG=YES'
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//INDD1  DD DSN=HOST1.DFDSS.BACKUP.MEDIUM,DISP=OLD *I/P DDNAME *
//OUTDD1   DD UNIT=3390,VOL=SER=UNIX11,DISP=SHR    * O/P DDNAME *
//SYSIN    DD *
  RESTORE                       /* COMMAND TO RESTORE      */  -
    DATASET( -                  /* DSN RESTORE OPERATION   */  -
     INCL(HOST1.MEDIUM.MASTER))  /* DSNAME TO BE RESTORED   */  -
    INDDNAME(INDD1)             /* I/P DDNAME              */  -
    OUTDDNAME(OUTDD1)           /* O/P DDNAME              */  -
    CANCELERROR                 /* CANCEL JOB IF ERRORS    */  -
    WRITECHECK                  /* VERIFY THE DATA         */  -
    REPLACE                     /* REPLACE WITH DATA FR SRC*/  -
    WAIT(0,0)                    /* NO WAIT FOR DATA SET    */
/*
//
```

Backup Using IEBGENER

```
//BKPM  JOB ,BRUSER,CLASS=S,MSGCLASS=L,MSGLEVEL=(1,1)
//*
//**************************************************************
//*** THIS JOB USES IBM UTILITY PROGRAM "IEBGENER" TO CREATE  ***
//***   A BACKUP OF THE DSNAME.                               ***
//**************************************************************
//*
//**************************************************************
//***              BACKUP MEDIUM DSN FROM MF DASD             ***
//***              DSN =  HOST1.MEDIUM.MASTER                 ***
//**************************************************************
//*
//STEP1    EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUT1   DD DSN=HOST1.MEDIUM.MASTER,DISP=OLD,  ** I/P DSNAME **
//         UNIT=3390,VOL=SER=UNIX10
//SYSUT2   DD DSN=HOST1.GENER.BACKUP.MEDIUM,     ** O/P DSNAME **
//         DISP=(NEW,CATLG,DELETE),
//         UNIT=3480,LABEL=(1,SL)
//SYSIN    DD DUMMY
/*
//
```

Restore Using IEBGENER

```
//RSTM JOB ,BRUSER,CLASS=S,MSGCLASS=L,MSGLEVEL=(1,1)
//*
//*********************************************************
//***                                                  ***
//*** USES IBM UTILITY PROGRAM “IEBGENER” TO RESTORE ***
//***   A BACKUP OF THE DSNAME TO DASD. ***
//***                                                  ***
//*** DSNAME MAY BE RESTORED TO THE SAME NAME OR A NEW   ***
//***   NAME.                                          ***
//***  TO RESTORE TO A NEW NAME THE NEW DATA SET “MUST”
//*** BE PRE-ALLOCATED ON THE “OPEN SYSTEMS” SIDE. ***
//***                                                  ***
//*********************************************************
//*
//**********************************************************
//***           RESTORE MEDIUM DSN FROM TAPE TO MF DASD ***
//***             DSN = HOST1.MEDIUM.MASTER ***
//***         NEW DSN = HOST1.NEW.MASTER ***
//*********************************************************
//*
//STEP1    EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=HOST1.GENER.BACKUP.MEDIUM,     ** I/P DSNAME **
//         DISP=OLD
//SYSUT2   DD DSN=HOST1.NEW.MASTER,              ** O/P DSNAME **
//         DISP=OLD,
//         UNIT=3390,VOL=SER=UNIX11
//SYSIN    DD DUMMY
/*
//
```

Backup Using FCOPY

```
//*
//**************************************************************
//***                                                        ***
//***  THIS JOB USES FCOPY TO DO A  “LOGICAL”  VOLUME BACKUP   ***
//***   IT WILL BACK UP ALL ALLOCATED SPACE.                 ***
//***                                                        ***
//**************************************************************
//***         BACKUP MEDIUM SIZE VOLUME FROM MF DASD         ***
//***                                                        ***
//***          VOLUME = MED001                               ***
//**************************************************************

* $$ JOB JNM=VOLCOPY,DISP=D,PRI=3                              *
* $$ NTFY=YES,
* $$ CLASS=0
* $$ LST DISP=H
// JOB VOLCOPY
// UPSI 00000000
// ASSGN SYS004,ECKD,VOL=MED001,SHR
// ASSGN SYS005,480
// TLBL UOUT,’HOST1.MEDIUM.MASTER’,95/365
// EXEC FCOPY
DUMP VOLUME IV=MED001 LIST
/*
/&
* $$ EOJ
```

Restore Using FCOPY

```
//*************************************************************
//***                                                      ***
//***   THIS JOB USES FCOPY TO DO A  “LOGICAL”  RESTORE.    ***
//***                                                      ***
//*************************************************************
//***        RESTORE MEDIUM SIZE VOLUME FROM MF DASD        ***
//***                                                      ***
//***          VOLUME = MED002                             ***
//*************************************************************
//*
* $$ JOB JNM=VOLREST,DISP=D,PRI=3                            *
* $$ NTFY=YES,
* $$ CLASS=0
* $$ LST DISP=H
// JOB VOLREST
// UPSI 00000000
// ASSGN SYS005,ECKD,VOL=MED002,SHR
// ASSGN SYS004,480
// TLBL UIN,’HOST1.MEDIUM.MASTER’,95/365
// EXEC FCOPY
RESTORE VOLUME OV=MED002 LIST
/*
/&
* $$ EOJ
```

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 the DataShare Facility. That is, mainframe, open systems, and backup / restore 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.

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.

VTOC Volume Table of Contents (mainframe directory).

Index