

BrightStor™ CA-Disk™ Backup and Restore for UNIX System Services

User Guide

2.0 SP04



Computer Associates™

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this documentation for their own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been destroyed.

To the extent permitted by applicable law, CA provides this documentation "as is" without warranty of any kind, including without limitation, any implied warranties of merchantability, fitness for a particular purpose or noninfringement. In no event will CA be liable to the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

© 2002 Computer Associates International, Inc. (CA)

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.



Contents

Chapter 1: General Information

Overview	1-1
Considerations	1-2
Pattern Matching	1-2
Learning to Use the Product	1-3
Basic Conventions and Requirements	1-3
Terminology	1-4
Control Statements	1-4
File Names and Patterns	1-5
Tailoring to User Needs	1-7
Date Formats	1-7
Year 2000 Considerations	1-9
Separator Pages	1-10
Messages	1-10
User Abend Codes	1-11
Storage (Memory) Requirements	1-11
JCL Requirements	1-11

Chapter 2: Basic System—DMSU Procedure

Data Storage Command Language	2-1
Graceful Shutdown	2-3
Condition Codes	2-4
UNIX File Permissions	2-5
Commands	2-6
DSCL Command Sequence	2-7
SCAN and FIND Rules	2-8
SELECT and EXCLUDE Rules	2-10
SET Command	2-11
SCAN UNIX	2-12
FIND Command	2-13

Processing Considerations	2-13
SELECT/EXCLUDE	2-15
TABLE= Considerations.....	2-16
Considerations	2-17

Chapter 3: Directives

HFSCREATE Command and Parameters	3-1
Examples of Command Use:	3-3

Chapter 4: Reports

UREPORT Command and Parameters	4-1
UATT – Permission/Owner/Size/Change Date Report	4-2
USPC – General Statistics of the UNIX File System	4-2

Chapter 5: Backup/Archive

Terminology	5-1
Midnight	5-2
Files Data Set Server	5-2
BACKUP Command and Parameters	5-2
BACKUP	5-3
ARCHIVE	5-3
Sample BACKUP Report	5-5

Chapter 6: Restore

Explicit Restore	6-1
Multiple Files Data Set for Restore	6-1
Restore Condition Codes	6-2
Files Data Set Updates During Restore/Recover	6-2
Use of Primary or Copy Archive Volumes	6-2
RESTORE Command	6-3
Execution	6-5
Sample Restore Report	6-6
FMS Recover	6-7
Condition Codes	6-7
Commands	6-7

DSCL SELECT CRITERIA Field Name Descriptions	6-9
RECOVER.....	6-12
Execution	6-13
Sample Recover Report	6-14

Chapter 7: Merge

Overview	7-1
Specifying What to Merge	7-2
Specifying Merge Output Options	7-4
Implementation Concerns.....	7-4
Operation Concerns	7-5
TAPEPULL Processing	7-6
Abend/Restart Considerations	7-7
Simulate Mode Considerations	7-8
Processing Logic	7-9
JCL – MERGE Command	7-10
Condition Codes	7-11
MERGE Command and Parameters	7-11
SYNTAX	7-12
Sample MERGE Report	7-15

Chapter 8: Index Maintenance

Maintenance Commands	8-1
PATHDELETE.....	8-1
VOLDELETE	8-4
PURGE	8-5
REBUILD	8-7
Reporting Commands	8-9
LISTU.....	8-10
LISTV Command	8-13
LISTV Flags	8-13
Sample LISTV Report	8-14
Making Additional Copies of Archive Tapes	8-15

Chapter 9: Using ISPF

General Considerations	9-2
General Panel Overview	9-3
Main Panel	9-3
Setting Configuration Options	9-5
File System/Path List Panel	9-6
File List Panel	9-10
Command Parameter Panels	9-12
Using the Panels	9-15
Setting Configuration Options	9-16
Explicit File Record Display	9-18
Explicit File Restore	9-20
Explicit Path Recovery	9-22
Displaying the Path List	9-24
Recover Paths	9-28
Restore from list	9-32

Chapter 10: Move/Copy

Considerations	10-1
MOVE/COPY Command and Parameters	10-2
COPY	10-3
MOVE	10-6

Appendix A: Installation

Overview	A-1
The SMP/E Process	A-2
Step 1. Library Names and Locations	A-2
Step 2. Download the Related Installation Materials Library	A-3
Step 3. Prepare the SMP/E Environment	A-4
Step 4. Allocate Data Sets	A-4
Step 5. Set DDDEF Entries	A-4
Step 6. Receive the Product	A-4
Step 7. Apply the Product	A-4
Step 8. Accept the Product	A-5
Step 9. Apply Maintenance	A-5
The Files Data Set	A-6
Create a Files Definition Member in Parmlib	A-6
Initialize the Files Data Set	A-7

The PROCLIB Data Set	A-7
Modify the following Cataloged Procedures	A-7
Running Tests	A-7
Considerations	A-8
Test 1	A-8
Test 2	A-9
Test 3	A-9
Test 4	A-9
Customizing ISPF Support	A-10
Customizing ISPF	A-11

Appendix B: Files Data Set

Subfiles of the Files Data Set	B-2
Restrictions	B-2
Subfile Entry Definition	B-2
Subfile Definition Parameters	B-3
Files Data Set Management	B-4
Formatting the Files Data Set	B-4
Determining the Allocation Size of the Files Data Set	B-4
Files Data Set Status Report	B-5
Field Descriptions—Status Report	B-6
Files Data Set Utilities	B-8
UNLOAD and RELOAD	B-8
UNLOAD	B-9
UNLOAD Command and Parameters	B-10
RELOAD	B-11
RELOAD Command and Parameters	B-12

Appendix C: Tape Management

Allocation Considerations	C-1
Ddnames Used	C-1
Producing Duplicate Archive Tapes Concurrently	C-2
Cataloging the Tape Archives	C-2
Assigning Tape Expiration Dates	C-2
Tapepool Considerations	C-3
From JCL	C-5
From a Tapepool	C-5
From the Operators	C-5

Customizing Tape Management Support	C-5
Method 1—Controlling Tapes Through the EDM	C-7
Method 2—Controlling Tapes by Expiration Date	C-9
Method 3—Controlling Tapes by Catalog Status	C-10
Method 4—Controlling Tapes Using TLMS/EDM	C-11
Interfacing Considerations With BrightStor CA-1	C-11

Appendix D: Tailoring Options

Suggested System Parameters	D-2
System Parameter for Lines Per Page	D-2
System Parameter for Separator Pages	D-2
User Exit for All Sysout Print Lines	D-2
System Parameters for Backup	D-2
Performance Tips	D-2
Files Data Set	D-2
Back Up Considerations	D-3
Specifying the Archive Medium	D-3
Using Multiple Types of Media	D-4
Naming the Archives	D-5
JCL Overrides and DSNNAME Usage	D-6
Assigning an Expiration Date to the Archive Data Set	D-7
Backing Up to Disk: Requirements and Recommendations	D-7
Archive Integrity and Security	D-12
Archive Tape Protection	D-12
Checks Against Overwriting Valid Tapes	D-13
Tape Status After Abends	D-14
User-Specified Condition Codes	D-15

Appendix E: Sysparms

SYSPARMS Member in PARMLIB	E-1
System Parameter Names Reserved for Users	E-2
Overriding Sysparms Instream	E-2
System Parameters	E-3
System Parameter Definitions	E-5

Appendix F: User Exits

User Exit Cross Reference Matrix	F-2
User Exit Descriptions	F-2
ARCDSNEX – Archive File Name Generation Exit	F-2
ARCEODEX – Archive Tape End Of File (CLOSE) Exit	F-3
DSCLSCEX – DSCL Screening Exit	F-4
SYSOUTEX – SYSOUT Exit	F-5
TAOPPOEX – Archive Tape Processing Exit (Post-Open)	F-6
TAOPPREX – Archive Tape Processing Exit (Pre-Open)	F-6
TAPCTLEX – Tape Management System Control Interface Exit	F-7
TAPEDLEX – Archive Volume Being Deleted from Archives	F-7
TMSCTLEX – Tape Management System Control Interface Exit	F-8
ADSTH025 – Direct Interface for BrightStor CA-1	F-9
ADSTH017 – Interface for TLMS	F-13
TPMGMTEX – Tape Management Interface Exit	F-13
UNLOADEX – Process Unloaded Files File Record	F-14
USERSEQ – ENQ/DEQ Exit	F-15

Appendix G: Parmlib

Purpose	G-1
Updating Member Data	G-2
Backup Considerations	G-2
PARMLIB Specifications	G-3
Examples	G-3
Special Rules for POOLDEFS	G-3
PARMLIB Syntax Validation Utility	G-4
User PARMLIB Members	G-4
DASD Pool List	G-4
Exclusion Table for Data Compression	G-6
Inclusion Table for Data Compression	G-6
Exemption Lists – Generalized	G-7
System Parameter List	G-7
User-Specified Condition Codes	G-8

Appendix H: JCL DD Statements

DD Statement Explanations	H-1
//ARCHIVEC	H-1
//ARCHIVES	H-2
//ARCHIVE0	H-2
//COMMANDS	H-2
//CMDPRINT	H-2
//COPYFILE	H-2
//DISPHOLD	H-2
//DISPWORK	H-3
//DSNEXPDT	H-3
//DSNIIN	H-3
//DSNIOUT	H-3
//DSNIWK01 ... WK03	H-3
//DSCBS	H-3
//FILEIN	H-3
//FILEOUT	H-4
//FILES	H-4
//FILEWK01 ... WK03	H-4
//FILPRINT	H-4
//INDEXREC	H-4
//MFILES	H-4
//MSGPRINT	H-4
//PARMLIB	H-4
//REPORTS	H-5
//SEQFILES	H-5
//SMFDATA	H-5
//SMSPRINT	H-5
//STEPCHAT	H-5
//SYSPARMS	H-5
//SYSPRINT	H-5
//TAPEPULL	H-5

Index

Glossary

General Information

IBM has implemented the UNIX operating system under their OS/390 umbrella. This operating system is every bit as unique and separate as MVS. But, because these operating systems must coexist on the same hardware platform, they must tolerate and not interfere with each other. The UNIX file system is one such area of toleration. IBM created Hierarchical File System (HFS) files to house the UNIX file system. To MVS, HFS files look like special VSAM Linear data sets. To UNIX, these files contain the UNIX file system that is a tree-based file system. This file system starts with a root "/" and expands to directories and files from there. There can be many HFS files to house the UNIX tree, however, the UNIX user sees the whole tree as one.

BrightStor CA-Disk Backup and Restore for UNIX System Services (BrightStor CA-Disk for USS) reports, backs up, restores, moves, and copies UNIX files contained within HFS files. A special UNIX file system scanner scans all or part of the tree structure for the purposes of reporting on and backing up directories and files. An index is maintained that tracks the directories and files backed up. The capability to do individual restores or multiple file restores has also been provided.

Overview

BrightStor CA-Disk for USS provides the following supported functions:

- Report
- Backup
- Archive
- Restore (explicit format)
- FMS Recover (implicit format)
- Index Maintenance
- Move
- Copy

Reporting, backup, archive, move, copy, and FMS recover use the DSCL command structure to perform their functions and are executed out of the DMSU and FMSU cataloged procedures. Restore, merge, and index maintenance use their own command structure and are executed out of the RESTOREU and IXMAINTU cataloged procedures respectively.

Considerations

This next section provides a list of considerations when using BrightStor CA-Disk for USS.

Files Data Set

Because of the differences between MVS and UNIX file types, a separate Files Data Set is required when backing up OS/390 UNIX files. For formatting instructions on creating a separate Files Data Set, see Formatting the Files Data Set in Appendix B “Files Data Set.”

UNIX File System

UNIX files are contained within one or more HFS files. The file systems to be backed up must all be mounted and available in order to get to all the files that need to be backed up.

Pattern Matching

BrightStor CA-Disk for USS uses the standard pattern matching characters as the MVS version except for the “/” (forward slash).

Note: Information on other pattern matching characters can be found in File Names and Patterns in this chapter.

Since the forward slash in UNIX is used to denote a directory structure, it could not be used in the traditional BrightStor CA-Disk for USS sense. BrightStor CA-Disk for USS uses the “/” to represent any variable character from that position to the end of the name. The portion of the name that precedes the slash is referred to as the PREFIX name. Therefore, the “\” (backslash) character was selected to replace it wherever a path name is specified.

For example, to select the files and directories in and under the directory “/bin,” specify the PATH parameter as follows:

```
PATH=/bin\
```

To select just the files within the “/bin” directory, specify the PATH parameter as follows:

```
PATH=/bin
```

Learning to Use the Product

This manual contains all of the information necessary to run the BrightStor CA-Disk for USS system after it has been installed. It is organized into sections pertaining to the functions of the system. Each section describes what purpose the function serves, how to use it, and considerations for implementing it. Since this is a reference manual, it is not easy to learn how to use the system by reading the BrightStor CA-Disk for USS *Users Guide* from beginning to end. If you are new to BrightStor CA-Disk for USS, use the following list of items as a guide for getting to know the system:

1. Review the Table of Contents of the manual to get an idea of the functions the system provides and what information is supplied about them.
2. Continue reading this entire section of the manual. This helps you learn how the system operates, what its restrictions are, and how to code BrightStor CA-Disk for USS command statements.
3. For a description of the BrightStor CA-Disk for USS PARMLIB data set, which is used by all BrightStor CA-Disk for USS functions, see Appendix G “Parmlib.” It is important to understand its purpose and how it is used.
4. At this point you are prepared to learn about specific functions of the system. To do this, select a function that you are interested in and read the section for it. We suggest that you select the REPORT function as the first one to learn because it is the easiest to use and provides a base of knowledge on how BrightStor CA-Disk for USS generally operates.
5. Be sure that when running your first set of jobs that you specify the SET MODE=SIMULATE command for all functions that provide it. After you have achieved the results you want with SIMULATE, remove the parameter (or comment it out), and run the function again, causing BrightStor CA-Disk for USS to actually perform the requested task.

Using these techniques, you can create simple cases, and then add to them. They help you to better understand the different options of each function and what they do. Keep track of your test output for future reference.

Basic Conventions and Requirements

Throughout the operation of BrightStor CA-Disk for USS, certain basic conventions and requirements are observed that help make processing consistent. These are summarized in the following pages.

Terminology

Several data storage management terms are commonly used from location to location, but the meanings given to the terms can vary widely. A glossary is included to clarify both the common terms as well as those unique to BrightStor CA-Disk for USS.

Control Statements

All commands consist of a character string of ten or fewer characters. While a command does not have to begin in any particular position, nothing can precede it in a statement.

Following a command, and preceded by one or more spaces, is a series of parameters that can consist of a combination of keyword and simple entries. Parameters can be specified in any order for a given command. For an example of this, see File Names and Patterns in this chapter.

Commands are specified after the `SYSIN DD *` statement for any function being run. They are then terminated with a `/*` after the last command entry. Commands can be specified in the JCL or through an input data set. If the command input is from a data set, its LRECL must be either 80 or 255. The input data set can be a sequential file or a PDS member.

If commands are 71 characters or less they can be specified within the JCL. If they are more than 71 characters, they can be continued on to the next line as described below or up to 246 characters can be specified on a single line through an input data set.

Command statements can be continued to additional lines in two ways:

- One technique permits termination of one statement line with a complete parameter and a trailing comma, with the remaining parameters starting in any position on the next line.
- An alternate continuation technique requires coding of the statement being continued through column 71 or 246 and placement of an X in column 72 or 247 to designate that the statement is being continued. The first non-blank character in the following statement is appended to the character in column 71 or 246 of the previous statement. A string cannot be continued to the next line. For example, a path name longer than what fits on a single 71-character line cannot be continued onto the next line. In these cases, the path name must be fed into *SYSIN* through a data set with an LRECL=255 as described above.

If you specify many parameters for a command, the first parameter must begin in the statement containing the command. A statement with only a command is processed as having no parameters. Parameter names can be abbreviated down to as few as three characters or as few as it takes to keep them from being confused with another parameter. Comments are permitted with the control statements and are designated by an asterisk in column one. They are printed out in the BrightStor CA-Disk for USS CMDPRINT exactly as they appear with the control stream.

File Names and Patterns

The PATH= parameter is probably the most frequently used of all parameters, and the one where the greatest degree of flexibility is usually required. BrightStor CA-Disk for USS allows either full file names or file name patterns to be entered in the PATH= parameter. A pattern name consists of the usual alphanumeric and national characters allowed in a file name, but also the following:

- A backslash (\)
- An exclamation point (!)
- A question mark (?)

The backslash (\) can be used to represent any variable character from that position to the end of the name. The portion of the name that precedes the slash is referred to as a PREFIX name. Following are some examples of the backslash (\) pattern in PATH=:

Example	Result
PATH=/A\	Selects all paths that begin with the letter "A."
PATH=/A/TEST\	Selects all paths that begin with the character string "A.TEST", with any following characters.

The exclamation point (!) can be used to represent any variable character up to the character string following the exclamation point; for example, the exclamation point defines the beginning of a character string (terminated by the next pattern character or the end of the data) that can be found anywhere within the name. Following are some examples of the exclamation point (!) pattern in PATH=:

Example	Result
PATH=!TEST	TEST selects all paths that contain "TEST" somewhere in the name.
PATH=/A?!DEPT21	Selects all paths that have a two-character first-level index that starts with an "A" and contain "DEPT21" somewhere in the remainder of the name.
PATH=!TEST!LOAD	Selects all paths that contain "TEST" somewhere in the name and "LOAD" somewhere following "TEST."

The question mark (?) can be used to represent any variable character within an index level or simple name. Multiple occurrences can be used within each level or simple name. Following is an example of the question mark (?) pattern in PATH=:

Example	Result
PATH=/A/TEST??	Selects all two-level paths with a first-level index of "A" and a simple name six characters in length, the first four of which must be "TEST"; for example, TEST01, TEST02, TEST99, TEST1A, TESTXX, and so on.

File Name Validation

BrightStor CA-Disk for USS commands that name a single specific file to be processed are referred to as "explicit" commands. Those that cause the UNIX file system to be scanned are called "implicit" commands. By definition, the file entered on an explicit command cannot contain any pattern characters. The specific name entered (original and perhaps new name) is validated by the UNIX operating system. Invalid characters cause the command to be rejected. Implicit commands do not have the file name/pattern entries validated in this manner, since they process names that already exist in the UNIX file system.

Tailoring to User Needs

Many system parameters exist that cause slight variations from the default program execution. If you want a particular variation, the appropriate sysparm is placed in the SYSPARMS member of the PARMLIB data set. For a cross-reference of sysparms applicable to each BrightStor CA-Disk for USS function, see Appendix E “Sysparms.” A description is also given of how sysparms can be dynamically overridden in those special cases that warrant it.

For larger scale variations, user exits are available. For a cross-reference of these, see Appendix F “User Exits.”

Date Formats

You can enter dates as input parameters in any of the following formats, unless otherwise noted in the documentation of particular BrightStor CA-Disk for USS functions. For related information, see Year 2000 Considerations in this chapter.

Gregorian Date Formats

Following is a list of Gregorian data formats:

- ddmonyy
- ddmonyyyy
- mm/dd/yy
- mm/dd/yyyy

where:

Format	Description
dd	A calendar day of a month from 01 to 31.
mon	A calendar month such as JAN, FEB, or MAR.
mm	A calendar month number from 01 to 12.
yy	A calendar year from 00 to 99 this year is stored as 1900 to 1999.
yyyy	A calendar year from 1900 to 2155.

Julian Date Formats

Following is a list of Julian data formats:

- yyddd
- yyyyddd
- yy.ddd
- yyyy.ddd
- yy/ddd
- yyyy/ddd

where:

Format	Description
ddd	A day of a year from 001 to 366
yy	A calendar year from 00 to 99 this year is stored as 1900 to 1999
yyyy	A calendar year from 1900 to 2155

For example, you can enter the date July 19, 1997, as:

Gregorian Dates	Julian Dates
19JUL97	97200
19JUL1997	1997200
07/19/97	97.200
07/19/1997	1997.200
	97/200
	1997/200

Also, as DSCL Select or Exclude criteria, you can specify the date in the form TODAY, TODAY-n, or TODAY+n, where “n” is a number of days in the past or future. For example:

- To select files that were last accessed yesterday, you can enter:
CRI=(UNIXLSAC, EQ, TODAY-1)

Year 2000 Considerations

BrightStor CA-Disk for USS is Year 2000 compliant. Year 2000 date testing was accomplished on a P/390 with BrightStor CA-Disk for USS Release 1.1.0 installed. All functions were tested while the system was IPL'd with the following dates:

- 1999.365 – end of century
- 2000.001 – beginning of century
- 2000.060 – February 29

BrightStor CA-Disk for USS follows all IBM conventions to identify dates within specific centuries in the following areas:

- Non-VSAM data sets use discontinuous binary, an xxyyy format where xx is from 0 255 and is the year when added to a base of 1900.
- VSAM and tapes use a separate century field.
- The Files Data Set uses the discontinuous binary for all dates.
- BrightStor CA-Disk for USS REPORTS use the mm/dd/yyyy.

Special Dates

BrightStor CA-Disk for USS treats Julian dates “99365”, “99366”, and “99999” as “never scratch” dates. That is, data sets whose ARCHVOLS or UNXINDEX records use these dates as expiration dates are not deleted automatically as expired by the BrightStor CA-Disk for USS function IXMAINT.

JCL, SYSPARMS, or Command parameters

Even after January 1, 2000, BrightStor CA-Disk for USS continues to honor these special “never expire” dates. If JCL specifies LABEL=EXPDT=99365, the data sets processed are kept until explicitly deleted even after December 31, 1999.

For example:

- If you explicitly specify EXPDT=99365 for an ARCHVOL or UNXINDEX record, then the data set is kept until explicitly deleted even after December 31, 1999:

```
SCAN UNIX
SELECT PATH=/\
BACKUP EXPDT=99365
//SYSPARMS DD *
DYNEXPDTE99365
```

The ARCHVOLS and UNXINDEX records created by this sample BACKUP job have their respective EXPDT values set to 99365. These records reside in the Files Data Set even after December 31, 1999.

- If you specify RETPD=nnnn and the system calculates an expiration date of the last day 99365, the actual expiration date is “bumped” one day to January 1, 2000, to prevent inadvertently assigning a “never expire” date:

```
//jobcard
//BACKUP EXEC DMSU
//ARCHIVE0 DD LABEL=RETPD=nnnn
SCAN UNIX
SELECT PATH=/\
BACKUP RETPD=nnnn
```

The ARCHVOLS and UNXINDEX records created by this sample BACKUP job have their respective RETPD values set to “TODAY+nnnn.” If “TODAY+nnnn” calculates to 99365, the actual expiration date is “bumped” to 2000.001.

- If you specify RETPD=nnnn and the calculation extends beyond 2155.365, then the EXPDT is set to 2155.365.

WARNING! RETPD values calculating to 98nnn and 99nnn can cause problems in certain situations. For example, TMS considers tapes valued at 99060 “scratch” after 60 days of non-use.

Separator Pages

The number of BrightStor CA-Disk for USS separator or header pages can be controlled by sysparm FLYCOUNT. Zero to nine pages can be produced. This sysparm can be overridden in your SYSPARMS member in PARMLIB. For details, see Overriding Sysparms Instream in Appendix E “Sysparms.”

Messages

BrightStor CA-Disk for USS messages all begin with a common format of MODULENM MSGN, where MODULENM indicates the name of the module issuing the message, and MSGN is the message number itself. This identifying information is then followed by explanatory text.

User Abend Codes

BrightStor CA-Disk for USS issues user abends when it is impossible to issue a diagnostic message, when additional information is needed to diagnose a problem, or when it is unsafe to continue due to data integrity. Following is a list of user abend codes and common reasons for them:

User Abend	Common Reasons
24	MSGPRINT DD statement is missing.
100	Message is self-explanatory; dump was attempted.
200	Message is self-explanatory; no dump was attempted.

Storage (Memory) Requirements

Buffers for BrightStor CA-Disk for USS I/O areas as well as memory tables for such items as member, alias, and notelist entries are dynamically built. In some cases, BrightStor CA-Disk for USS does not create variable length memory tables above the 16MB line. Because these items vary for each file and device type being processed, exact storage requirements cannot be given.

Regions sizes on supplied JCL procedures range from 1024K to 5120K to accommodate running BrightStor CA-Disk for USS with the maximum buffer space possible. The amount of region required below the 16MB line varies depending upon the size of your UNIX file system.

JCL Requirements

JCL procedures are provided for most BrightStor CA-Disk for USS functions, and DD statements for any volume or file that BrightStor CA-Disk for USS has been instructed to process are dynamically allocated as needed, freeing the user from any need to modify JCL. For a list of these procedures, see Modify the Following Cataloged Procedures in Appendix A "Installation."

Basic System—DMSU Procedure

The commands below are executed by the DMSU JCL procedure:

- ARCHIVE
- BACKUP
- UREPORT
- HFSCREATE
- MOVE
- COPY

In this release, Data Storage Control Language (DSCL) is used by most functions of BrightStor CA-Disk for USS. For more information, see DSCL Command Sequence in this chapter.

Data Storage Command Language

The objective of the BrightStor CA-Disk for USS DSCL is to provide a rich, comprehensive palette of commands for efficient, common processing.

Within a single execution you can report on backup, archive, move, or copy files. You can do this by scanning the entire UNIX file system, by specifying a particular path to begin the scan, or by the HFS data set (containers for the UNIX files).

You also have a rich set of criteria to select files for and exclude files from processing. And a rich set of action parameters lets you control exactly what is done to these files.

BrightStor CA-Disk for USS preprocesses your command stream to find any “overlapping requests,” that is, those that request processing of the same files. Such requests are grouped together to allow all needed actions to be performed during a single pass of the file system.

The functions merge, index maintenance, and restore use their own unique command sets. Functions using DSCL are invoked by executing the JCL procedure named DMSU. Following is a sample of the BrightStor CA-Disk for USS DSCL command language:

```

EDIT                               JCL.CNTL (DSCL) - 01.02                               Columns 00001 00072
Command ==>                          Scroll ==> CSR
***** ***** Top of Data *****
000100 //JOB CARD JOB (ACCT,##),CLASS=X,MSCLASS=T,NOTIFY=&USERID
000200 //DSCL EXEC DMSU,REGION=4096K
000300 //STEPLIB DD DISP=SHR,
000400 // ISN=SAMS.UNIX.LOADLIB
000500 //FILES DD DISP=SHR,
000600 // ISN=SAMS.UNIX.FILES
000700 //PARMLIB DD DISP=SHR,
000800 // ISN=SAMS.CHCD.PARMLIB
000900 // DD DISP=SHR,
001000 // ISN=SAMS.UNIX.PARMLIB
001100 //SYSIN DD *
001200 SET MODE=SIMULATE
001300 SCAN UNIX
001500 SELECT PATH=/JOURNAL\
001600 UREPORT UATT
001700 /*
***** ***** Bottom of Data *****

```

Notes:

- DMSU is a cataloged procedure in DISK120.UNIX.PROCLIB from the distribution tape.
- STEPLIB DD is the BrightStor CA-Disk for USS load library renamed from DISK120.UNIX.LOADLIB from the distribution tape.
- FILES DD is the user-defined Files Data Set.
- PARMLIB DD is the BrightStor CA-Disk for USS parameter library renamed from DISK120.UNIX.PARMLIB from the distribution tape.
- The “Action command” can either be UREPORT (see UREPORT Command and Parameters in the chapter “Reports”), BACKUP/ARCHIVE (see the chapter “Backup/Archive”), MOVE/COPY (see the chapter “Move/Copy”), or HFSCREATE (see the chapter “Directives”).
- The FMSU procedure, used for data set recovery, is similar to the DMSU procedure and its DSCL command structure.

To avoid unexpected execution of partially bad command strings, the DMSU procedure takes no action until all commands are successfully understood, and all sysparm overrides are validated.

To allow for sorted reports, the DMSU procedure has separate sort and print steps.

Graceful Shutdown

You can optionally specify a system parameter that allows the computer operator to terminate DSCL processing prematurely (a graceful shutdown).

Conversely, the only other way to force termination is to cancel the job. This is not recommended because files that are “in use” are left in precarious positions.

A message is sent to the operator at the beginning of the process and remains on the console for the duration of the job. If the operator replies END to this message, it indicates processing is to be stopped and DSCL terminates when the current file processing has finished.

After the operator replies to the messages, processing completes and all reports and statistics are generated as normal.

This option permits you to start DSCL and execute until the resources are needed for something else. Thus, DSCL can be executed in otherwise unused machine time.

Following is an example of this graceful shutdown:

```

SDSF OUTPUT DISPLAY ISPDLM1A JOB01201 DSID 2 LINE 6 COLUMNS 02- 133
COMMAND INPUT ==> SCROLL ==> CSR
15.40.03 JOB01201 SSSMF10 BILLING INFO : TR* - STERLING:TRAINING
15.40.04 JOB01201 IEF403I ISPDLM1A - STARTED - TIME=15.40.04
15.40.10 JOB01201 *94 REPLY 'END' TO TERMINATE ISPDLM1A PROCESSING
15.40.40 JOB01201 *IECS01A M 0F29,PRIVAT,SL,NOCOMP,ISPDLM1A,DMS.REPORTS,ISPDLM1.DMSN.C1998162.T154039
15.40.40 JOB01201 *TMS001 IECS01A M 0F29,PRIVAT,SL,NOCOMP,ISPDLM1A,DMS.REPORTS,ISPDLM1.DMSN.C1998162.T154039
15.40.52 JOB01201 IECTH59 0F29,207981,ISPDLM1A,ARCHIVE0,PERMANENT ,0001, .C1998162.T154039
15.40.55 JOB01201 IECTH51 TAPE 0N 0F29,207981,SL,NOCOMP,ISPDLM1A,DMS.REPORTS,ISPDLM1.DMSN.C1998162.T154039
15.40.56 JOB01201 ICH4081 USER(ISPDLM1) GROUP(STERLING) NAME(UNKNOWN)
DMSPROD.R00.LOADLIB CL(DATASET) UOL(DMSK01)
INSUFFICIENT ACCESS AUTHORITY
ACCESS INTENT(READ) ACCESS ALLOWED(NONE)
15.40.56 JOB01201 IEC150I 913-38,IFG0194E,ISPDLM1A,DMS,SYS00001,06B0,DMSK01,DMSPROD.R00.LOADLIB
15.41.10 JOB01201 R 94,END
15.41.15 JOB01201 IEF234E K 0F29,207981,PUT,ISPDLM1A,DMS
15.41.15 JOB01201 -
--TIMINGS (MINS.)--
15.41.15 JOB01201 --JOBNAME STEPNAME PROCSTEP RC EXCP CONN TCB SRB CLOCK SERV PG PAGE SWAP UIO SWAPS
15.41.15 JOB01201 --ISPDLM1A REPORTS DMS 04 1578 15174 .12 .00 1.1 81578 803 0 0 0 0
15.41.15 JOB01201 --ISPDLM1A REPORTS AMSSORT FLUSH 0 0 .00 .00 .0 0 803 0 0 0 0
15.41.15 JOB01201 --ISPDLM1A REPORTS AMSEXEC FLUSH 0 0 .00 .00 .0 0 803 0 0 0 0
15.41.18 JOB01201 --ISPDLM1A REPORTS SORT1 00 138 592 .00 .00 .0 4326 803 0 0 0 0
15.41.22 JOB01201 --ISPDLM1A REPORTS PRINT 00 293 1501 .01 .00 .0 8254 803 0 0 0 0
15.41.22 JOB01201 IEF404I ISPDLM1A - ENDED - TIME=15.41.22
15.41.22 JOB01201 --ISPDLM1A ENDED. NAME=MULLINGS TOTAL TCB CPU TIME= .13 TOTAL ELAPSED TIME= 1.3
15.41.22 JOB01201 $HASP395 ISPDLM1A ENDED
----- JES2 JOB STATISTICS -----
Aa B TCP/IP R 2 C 21 16:04 6/11/98

```

While reviewing the example of a graceful shutdown, note the following time stamps:

Time Stamp	Description
15.40.10	The outstanding message generated by using DSCLRPLY. This message remains on the console for the duration of the DSCL job.
15.41.10	The operator replied END. Doing so instructs DSCL to begin termination. The elapsed time DSCL takes to complete the termination is variable, depending on where it is in the file processing.
15.41.22	The PRINT step concludes normally, resulting in all reports and statistics being generated completely.

For more information, see Appendix E “Sysparms.”

Condition Codes

The DMSU procedure returns the following JCL condition codes:

Code	Description
0	All actions were processed successfully and informational messages issued. Check to see if any messages are issued and if further action is required. Review the generated reports to make sure that the files were processed as stated.
4	Some minor problems were encountered in processing, but it was possible to continue. The results are not exactly what you wanted, but there seemed to be no harm in continuing. Warning messages were issued. Review these messages and the generated reports to make sure that the files were processed as stated. Causes of this return code include the following: <ul style="list-style-type: none">■ The selected file was in use by another task.■ There was insufficient access authority.■ An I/O error occurred.■ Files were found for processing but no processing was done because they did not meet the stated selection criteria.■ The file was not selected because it is restricted for the specified action command.

Code	Description
8	<p>A major problem was encountered in processing, resulting in BrightStor CA-Disk for USS not even attempting some processing. Error messages were issued.</p> <p>Causes of this return code include the following:</p> <ul style="list-style-type: none"> ■ The volume was not mounted. ■ There was an inability to get the Format-4 DSCB for a volume. ■ A volume-oriented backup error occurred. ■ Path scan errors occurred. ■ No files could be found for processing.
12	<p>The entire function, or a large portion of it, could not be processed. Error messages were issued.</p> <p>Causes of this return code include the following:</p> <ul style="list-style-type: none"> ■ Task abends occurred. ■ Files were selected for processing but were bypassed by the action being attempted.
16	<p>Commands were in error. Error messages were issued and no action was taken.</p>

UNIX File Permissions

The UNIX file system determines who can do what to files based upon permission bits contained in the file attributes. Various functions within BrightStor CA-Disk for USS allow the ability to alter these attributes. Specification of these altered permissions take the form of a number provided to the parameter of the function/command being executed. The numbers and what they mean are defined below.

For permissions, specify a three-digit octal number. The first digit is access permission for the file owner. The second digit is access permission for any member of the group of the file. The third digit is access permission for anyone else. An octal digit is a number in the range of zero to seven. The permission associated with each value is:

Code	Description
0	None
1	Search or execute
2	Write

Code	Description
3	Write and search or execute
4	Read
5	Read and search or execute
6	Read and write
7	Read, write, and search or execute

Commands

The DMSU procedure is controlled by commands provided to the SYSIN DD statement. The commands are of five main types:

- Environment commands – those commands that set up the environment in which BrightStor CA-Disk for USS runs: SET.
- Candidate commands – those commands that find candidate data sets: SCAN and FIND.
- Selection commands – those commands that select or exclude data sets to be processed, based on your own specific criteria: SELECT and EXCLUDE.
- Action commands – commands that take action upon the selected resources: UREPORT, BACKUP, ARCHIVE.
- Directive commands – standalone commands that do not require the support of Candidate, Selection, or Action commands: HFSCREATE

The minimum information required for you to execute a DSCL action command is a SCAN or FIND command to name the path where processing begins, a SELECT command to select the files, and an action command. Directive commands may be executed on their own.

You can add other commands to gain additional selection or functionality.

DSCL Command Sequence

DSCL commands are listed below in their proper sequence. Each command is optional, except as described below:

Type	Name	Parameters
Environment	SET	MODE=, UID=,
Candidate	SCAN	UNIX, HFS, EXCHFS, SHOWFS
Candidate	FIND	PATH=, TABLE=, LEVELS=, HFS=, MPOINT, UNMOUNT
Selection	SELECT	PATH=, FILE=, TABLE=, CRITERIA=
Selection	EXCLUDE	PATH=, FILE=, TABLE=, CRITERIA=
Action	UREPORT	UATT, UFPC
Action	BACKUP	RETPD=, EXPDT=
Action	ARCHIVE	RETPD=, EXPDT=, DISP=
Action	COPY	TOHFS=, TOPATH=, DIRCREAT, SCRATCHFIL, MPOINT, UOWNER=, UID=, GOWNER=, GID=, PERMS, OVERWRITE=, PATHONLY, NOLOAD,
Action	MOVE	TOHFS=, TOPATH=, DIRCREAT, SCRATCHFIL, MPOINT, UOWNER=, UID=, GOWNER=, GID=, PERMS, OVERWRITE=, VERIFY
Directive	HFSCREATE	DSN=, SPACE=, TRKS, CYLS, VOLUMES=, MPOINT=, DATACLAS=, MGMTCLAS=, STORCLAS=

The SET command is optional. Although the SCAN and FIND commands are each optional individually, you must specify at least one SCAN or FIND command before any action command.

SELECT commands are mandatory but EXCLUDE commands are optional.

Although action commands are each optional individually, you must specify at least one action command for each SCAN/FIND group (for more information, see FIND Command in this chapter).

Because the purpose of each command is largely self-evident, general rules and guidelines for using these commands are described immediately following. Detailed explanations for most DSCL commands and parameters are presented later in this section. Detailed explanations for each of the action and directive commands and their parameters are presented in later sections in this manual.

SCAN and FIND Rules

You can specify any number of SCAN or FIND commands in a command stream. Multiple commands can also be intermixed for any given action command. BrightStor CA-Disk for USS observes the following rules in processing SCAN and FIND commands. The rules are complex enough that you can try them out with SET MODE=SIMULATE until you are familiar with them.

Note: SET MODE=SIMULATE is not necessary and is ignored while executing the UREPORT command, because no action is taking place.

SCAN and FIND Rule #1

DSCL processes the specified requests in terms of logical blocks termed "SCAN/FIND groups." A SCAN/FIND group begins with a SCAN or FIND command and ends when the next non-consecutive SCAN or FIND command is encountered, or at the end of the input command stream.

At least one ACTION command must be specified in every SCAN/FIND group. In its simplest form, a valid SCAN/FIND group looks like:

```
SCAN UNIX
SELECT PATH=/usr\
UREPORT UATT
```

which causes BrightStor CA-Disk for USS to scan all files in the /usr directory and produce a user attributes report.

Another simple SCAN/FIND group:

```
FIND PATH=/usr/ispdrk3\
BACKUP RETPD=30
```

begins processing at the /usr/ispdrk3 directory and backs up all files, including those in subdirectories. These are kept for 30 days.

While DSCL does not care about spacing, the indentation shown can help us in the examples that follow.

SCAN and FIND Rule #2

Consecutive SCAN commands, FIND commands, or both, apply to all subsequent selection and action commands until another SCAN or FIND command is encountered.

For example, if you have multiple naming conventions, you can report on each of your files with:

```
FIND PATH=/usr/ispdrk3\
FIND PATH=/usr/isprpn3\
UREPORT UATT
```

The REPORT command applies to each of the FIND commands. While it is possible to use both SCAN and FIND in combination, there is no benefit in doing so since the entire file system is scanned anyway.

SCAN and FIND Rule #3

SELECT commands, EXCLUDE commands, or both, apply to all action commands that follow until either a non-consecutive SELECT or EXCLUDE command is encountered, or until a SCAN or FIND command is encountered.

You can apply the preceding rule to one or more selection tests for multiple action commands. In the following example, the file system is scanned and all files that meet either SELECT test are both backed up and reported on:

```
SCAN UNIX
SELECT CRITERIA=(UNIXMODT, LE, TODAY), FILE=\
SELECT CRITERIA=(UNIXFTYP, EQ, REG), FILE=\
BACKUP
UREPORT UATT
```

In the following example, both directories are scanned and those files that have not been referenced this year are archived. In addition, those data sets that are larger than 1MB are reported on. Because the two SELECT commands are not consecutive, the first SELECT command has no effect on the ARCHIVE command. However, a data set can be selected by both action commands if it meets both selection criteria:

```
FIND PATH=/usr
FIND PATH=/tmp
SELECT CRITERIA=(UNIXFLSZ, GT, 1000000), FILE=\
UREPORT UATT
SELECT CRITERIA=(UNIXRFD, LT, 1999.001), FILE=\
ARCHIVE
```

You can also combine FIND commands with SELECT commands, EXCLUDE commands, or both:

```
FIND PATH=/usr
SELECT CRITERIA=(UNIXRFD, EQ, 1998.035), FILE=\
EXCLUDE CRITERIA=(UNIXUONR, EQ, <ISPDRK3>)
ARCHIVE
```

SCAN and FIND Rule #4

Any ACTION command applies to all SCAN commands, FIND commands, or both, in the current SCAN/FIND group.

This rule has already been demonstrated in previous examples. Whenever an action command is entered, the easiest way to determine which files it processes against is to look up in the command stream and find the first SCAN or FIND command that precedes it. The paths named on that command, plus any others named on preceding consecutive SCAN commands, FIND commands, or both, are affected by that ACTION command.

SELECT and EXCLUDE Rules

You can specify any number of SELECT and EXCLUDE commands in a command stream. Multiple commands can also be intermixed for any given action command. BrightStor CA-Disk for USS observes the following rules in processing SELECT and EXCLUDE commands. Use SET MODE= SIMULATE until you understand the complexities of the rules.

SELECT and EXCLUDE Rule #1

For each SELECT or EXCLUDE command, all parameters specified must be true for the statement to be true. This is equivalent to AND selection based on the parameters on any individual SELECT or EXCLUDE commands.

For example,

```
SELECT PATH=/usr, FILE=\\, CRITERIA=(UNIXUONR, EQ, <ISPDRK3>)
```

holds true only for files owned by ISPDRK3 in the /usr directory, not for any files in the directory, nor any files in subdirectories.

SELECT and EXCLUDE Rule #2

If only SELECT commands apply to an action command, at least one SELECT statement must be true for a file to be selected for BrightStor CA-Disk for USS processing. This is equivalent to ORing selection based on combining SELECT or EXCLUDE commands. This also means that when multiple SELECT commands apply to an action command, if any SELECT statement is true for a particular file, that file is selected for processing.

For example,

```
SCAN UNIX
SELECT FILE=accting\
SELECT FILE=payroll\
BACKUP
```

backs up files beginning with 'accting' or 'payroll' anywhere in the file system.

SELECT and EXCLUDE Rule #3

If only EXCLUDE commands are specified for a given action command and if any EXCLUDE statement is met, the file is bypassed. If no EXCLUDE commands test true, the file is processed.

For example,
 FIND PATH=/usr\
 EXCLUDE CRITERIA=(UNIXLSAC,GT,TODAY-30)
 ARCHIVE

archives any files in the /usr directory that have not been referenced in the past 30 days.

SELECT and EXCLUDE Rule #4

If both SELECT and EXCLUDE commands are specified for a given action command, the following rules apply:

- If an EXCLUDE command tests true, the data set is bypassed whether a SELECT statement tests true.
- If no EXCLUDE tests are met, at least one SELECT command must still test true for the data set to be processed.

SET Command

This optional command defines environment values that are in effect for the duration of this execution only. If used, you can put the SET command anywhere in the command stream before the first action command. For clarity, Computer Associates recommends that you specify this command before all other commands. As with all BrightStor CA-Disk for USS commands, the SET command can be commented out (nullified) by placing an asterisk "*" in column one of the line containing the command.

Following is a list of parameters that can be used with the SET command:

Parameter	Description
MODE=	Specify SIMULATE or SIM to have BrightStor CA-Disk for USS simulate the execution. To run in LIVE mode, either comment out the SET MODE= command from the input stream, or delete the line entirely. In general, in simulate mode, BrightStor CA-Disk for USS produces the normal messages and reports as if processing had taken place, but does not alter any files.
UID=	This parameter allows the setting of the effective user ID for the duration of the batch job. Its intent is to allow the user to invoke Superuser mode in order to have the authority to access all files. The user must be previously set up through their Security Administrator to allow them to invoke Superuser mode. If they are not, then this parameter has no effect and default security is in place for file access. Specify UID=0 to invoke Superuser mode.

Syntax

The syntax of the SET command is as follows:
 SET MODE=SIMULATE,UID=0

SCAN UNIX

This command instructs BrightStor CA-Disk for USS to find and scan the OS/390 UNIX file system. Doing so builds a list of candidate files for BrightStor CA-Disk for USS to select from. The SCAN command must precede any action command.

Syntax

The syntax of the SCAN UNIX command is as follows:
 SET MODE=UID=
 SCAN UNIX

Following is a list of parameters that can be used with the SCAN command:

Parameter	Description
SCAN HFS	This command indicates which HFS data sets are to be scanned for UNIX file selection. The difference between this command and the SCAN UNIX command is the SCAN UNIX command always scans the complete (MVS) data sets specified.
HFS=	Specify the name of the HFS data set to scan. Pattern characters may be used to select a group of HFS data sets.
EXCHFHS=	Specify up to ten names with or without patterns of HFS data sets not to scan. This is meant to be used in conjunction with the HFS= parameter to narrow down the list of HFS data sets to scan.
SHOWFS	Specify the parameter to print a line for every file system, mount point, and HFS name.

Syntax

The syntax of the SCAN HFS command is as follows:
 SET MODE=UID=
 SCAN HFS=,EXCHFHS=,SHOWFS

For example, for HFS data sets which start with the name of HFS.USER, backup all files contained within them. Following are the commands to show all file systems encountered:

```
SCAN HFS=HFS.USER\,SHOWFS
SELECT PATH=\
BACKUP
```

FIND Command

The FIND command specifies the path or HFS data set where processing begins. The function of the FIND command is to improve the performance of BrightStor CA-Disk for USS when only one or a small subset of files is to be processed. Rather than scanning the entire file system as happens with the SCAN command, BrightStor CA-Disk for USS starts scanning from the path or HFS data set specified in the command.

There are two distinct and separate FIND commands:

- FIND PATH=
- FIND HFS=

Each one is separate and has its own parameters. FIND PATH= is used for searching by UNIX path, and FIND HFS= is used for searching by MVS HFS data sets.

The SELECT and EXCLUDE commands can be used with the FIND command and they must follow the FIND command and precede the action commands for the specified path/file name or pattern. You can use this command in the same job stream as the SCAN command, although it duplicates effort if the file name specified on the FIND command is otherwise processed as a result of the SCAN command.

For an example of how the FIND and SCAN commands can accomplish the same objective, look at the following:

```
SCAN UNIX
SELECT PATH=/usr\
ARCHIVE
FIND PATH=/usr\
ARCHIVE
```

In this instance, the FIND command is probably a better choice because the overhead of scanning through all other paths is eliminated. However, a SCAN command with the SELECT command accomplishes the same objective.

Processing Considerations

While the SCAN command or FIND command is processing, there is an exposure window between the time the data sets have been selected for processing and when they are actually backed up or archived. It is possible that during this time the file can be deleted by some other event. If this happens, BrightStor CA-Disk for USS issues message 2664, which can be misleading in such cases.

Syntax

The syntax of the FIND command is as follows:

```
FIND PATH=, TABLE=, LEVELS=
FIND HFS=, MPOINT=, UNMOUNT
```

Following is a list of parameters that can be used with the FIND command:

Parameter	Description
PATH=	<p>This is where the path name to scan is placed. Specify a path or pattern of up to 255 characters in length. Up to ten path names can be specified. The only wildcard that is valid for this parameter is the backslash, which causes the default value of LEVELS= to be zero (see the explanation for the LEVELS= parameter below), and it must be the last character in the path name.</p>
TABLE=	<p>This parameter, required only when the PATH= parameter is not specified, is used to specify the name of a member in the parmlib data set containing the list (table) of fully qualified path names. Finally, this parameter is mutually exclusive with the PATH= parameter.</p> <p>Since SELECT commands, EXCLUDE commands, or both, can be combined with FIND TABLE=, paths and files are not just automatically selected because they appear in this list. In the following example, the path must be listed in the parmlib member named "UNXFILES" and must have been backed up within the last five days:</p> <pre>FIND TABLE=UNXFILES SELECT CRITERIA=(ARCDT, GE, TODAY-5)</pre> <p>Values specified in the TABLE= member have the same limitations as those for the PATH= parameter. For details, see the PATH= parameter listed in this table.</p> <p>There is no exact limit to the number of entries that can be entered into a TABLE= member. It all depends on the environment. The information for the table is compiled and stored in a variable length table in memory above the 16Mb line. The maximum number of table entries is directly related to the amount of memory available to DSCL.</p>
LEVELS=	<p>This optional parameter controls the number of directory levels that BrightStor CA-Disk for USS processes. A value of one causes BrightStor CA-Disk for USS to process only the directory specified in the PATH= parameter, and is the default when the pathname is not terminated with a backslash. A value of zero is interpreted to mean all levels, which is the default when the pathname is terminated with a backslash. Acceptable values are from 0 to 2147483647.</p>

Parameter	Description
HFS=	This parameter specifies an explicit MVS data set name of an existing HFS data set where files are scanned.
MPOINT=	The existence of this parameter indicates that if the HFS file is not already mounted, it is to be mounted at the mount point specified prior to processing. The directory specified as the mount point must already exist in the file system (this is a UNIX requirement).
UNMOUNT	This parameter indicates that the HFS file system is to be unmounted after processing is complete. For example, locate the HFS data set named HFS.USER1, mount it with a mount point of /user1, and then back up all the files within it. Unmount the HFS data set when you are done. <pre>FIND HFS=HFS.USER1,UNMOUNT,MPOINT=/user1 SELECT PATH=\ BACKUP</pre>

SELECT/EXCLUDE

The SELECT/EXCLUDE commands allow you to process only those files that satisfy the specified selection tests. These commands are required unless you have the sysparm SELECTRE specified with a value of N.

If a SELECT statement is specified, in order for a file to be selected, **all** tests specified on a single SELECT command must be met. If only one parameter is specified, only that single test must be met for selection to take place.

For example, if the PATH= parameter or the FILE= parameter, but not both, are specified on a single command, conceptually think, “process only these paths” or “process only these files.” If both the PATH= and FILE= parameters are specified, conceptually think “process only these files if defined in one of these paths.”

If multiple SELECT commands are specified for any given action command, this has the effect of a logical “OR” between SELECT commands. That is, if any SELECT command is true, the file is processed, regardless of whether other SELECT commands test true.

Syntax

The syntax of the SELECT/EXCLUDE commands is as follows:

```
SET MODE=
SCAN UNIX
SELECT PATH=,FILE=,CRITERIA=
EXCLUDE PATH=,FILE=,CRITERIA=
```

Following is a list of parameters that can be used with the SELECT/EXCLUDE commands:

Parameter	Description
PATH=	This is where the path name to scan is placed. Specify a path or pattern of up to 255 characters in length. Up to ten path names can be specified.
FILE=	If you want to limit the selection or exclusion to certain file names, you can do so by specifying this parameter.
TABLE=	<p>This parameter is used to specify the name of a member in the parmlib data set containing the list (table) of fully qualified path names. Finally, this parameter works the same as PATH=.</p> <p>Note that if both PATH= and TABLE= are specified, only paths that are in both lists are selected, excluded, or both, from processing.</p> <p>Values specified in the TABLE= member have the same limitations as those for the PATH= parameter, described in this table.</p> <p>There is not an exact limit to the number of entries that can be entered into a TABLE= member. It all depends on the environment. The information for the table is compiled and stored in a variable length table in memory above the 16Mb line. The maximum number of table entries is directly related to the amount of memory available to DSCL.</p>
CRITERIA	<p>Specify the selection criteria to be met before any BrightStor CA-Disk for USS action occurs. The criteria parameter is specified by a minimum of three subparameters as follows:</p> <p style="text-align: center;">(operand1, operator, operand2)</p>

TABLE= Considerations

The member name of the TABLE= parameter is specified with command parameters TABLE=. TABLE= is used by the SELECT/EXCLUDE (DSCL) commands and by the FIND (DSCL) command.

Place the pathnames and patterns that are to be included in processing in the member of parmlib named in the parameter. When BrightStor CA-Disk for USS is searching the file system for files to process, it ensures that the pathname is in this list before attempting to process it. If the pathname is not in this list, it is exempted from processing.

Note: By default, these tables are stored in the parmlib data set. However, you can store them in an alternative data set if you want. For details, see the TABLESDD parmlib parameter in System Parameter Definitions in Appendix E “Sysparms.”

Format

The format is 'DDDD.....' where:

- DDDD..... is a 1- to 1023-character pathname or pattern beginning in column one.

Considerations

An asterisk in the first column denotes a comment and is ignored for processing.

A line is continued by ending it with a comma, followed by a space.

Examples

The following entries process only the pathnames specified, and all their subdirectories. The first two entries are considered one path because of the continuation characters (comma, followed by a space).

```
/usr/development/sanbernardino/samsdisk/openedition,  
/ispdrk1\  
/usr/maintenance\  

```


Directives

Directive commands are special DSCL commands that do not necessarily require the normal SCAN/FIND, SELECT/EXCLUDE, UREPORT/BACKUP/ARCHIVE/MOVE/COPY combination of commands. These directives can be placed anywhere in a DSCL command stream and will be executed prior to any non-directive command. There is currently only one directive command: HFSCREATE.

The OS/390 UNIX File System is contained within containers called Hierarchical File System data sets (HFS) and with BrightStor CA-Disk for OS/390 UNIX ability to process and select UNIX files at an HFS data set level, it is possible to backup, archive, move, or copy files to new HFS data sets. Reasons for doing this may be to consolidate or expand HFS data sets or even to set up default HFS data sets when new users are added to the UNIX system. Normally, this would require a separate operation either online or batch job to get the new HFS data set created. The HFSCREATE command provides the ability to allocate an HFS data set from within the DSCL command stream.

HFSCREATE Command and Parameters

The HFSCREATE command is used to create HFS Data Sets. It may be executed without any other DSCL command. There is no limit to the number of times this command may be specified, however, as BrightStor CA-Disk for USS builds and validates the entire command structure in above-the-line memory before any command is executed, your system memory would be the only limitation.

Following is a list of parameters associated with the HFSCREATE command:

Parameter	Description
DSN=,	Specify the name of a new HFS data set here. This is a required parameter. Example: DSN=HFS . PAYROLL
SPACE=(pri,sec),	In pri, specify the initial amount of space to allocate for the new HFS data set. In sec, specify the incremental amount the new HFS file is to grow if/when the pri allocation is used up. This is a required parameter. Example: SPACE=(100 ,50)
{TRKS CYLS},	Works in conjunction with the space parameter. Specifies the type of allocation units to be used with the values in the SPACE parameter. This is a required parameter. Definitions are as follows: TRKS = Tracks CYLS = Cylinders Example: SPACE=(100 ,50) , TRKS
VOLUME=,	If the new HFS data set is not to be SMS managed, then specify the DASD volsers upon which to allocate the data set. Example: VOLUME=VOL001
MPOINT=,	Specify the HFS data set mount point here. When an HFS data set is mounted, a UNIX file system mount point must be provided. The mount point specified must be an existing directory name (UNIX requirement). This parameter is optional. If this parameter is not used, then the HFS data set will be allocated but not mounted. It will then have to be mounted manually with (where available) the MPOINT parameter within the DSCL command using the HFS data set. Example: MPOINT=/tmp/users/systems
MGMTCLAS=,	Specify the name of the SMS management class to use for allocating this data set. This parameter is optional. Example: MGMTCLAS=HFSSMC

Parameter	Description
STORCLAS=,	Specify the name of the SMS storage class to use for allocating this data set. This parameter is optional if storage class is automatically assigned by SMS, otherwise it is required. Example: STORCLAS=HFSSC
DATACLAS=,	Specify the name of the SMS data class to use for allocating this data set. This parameter is optional. Example: DATACLAS=HFSDC

Examples of Command Use:

Following are some examples of the HFSCREATE command:

1. Allocate an HFS data set with the name of HFS.CONOLID. It is to be SMS managed with a storage class name of = HFSSC, a management class name of = HFSSC, and a data class name of HFSDC. The amount of PRIMARY space to allocate is 1000 cylinders, with secondary space allocation of 250 cylinders.

```
//job.....
//EXEC DMSU
//SYSIN DD *
HFSCREATE DSN=HFS.CONOLID,
MPOINT=/tmp/ispjx11,
STORCLAS=HFSSC,
MGMTCLAS=HFSSC,
DATACLAS=HFSDC,
SPACE=(1000,250),
CYLS
/*
```

2. Allocate an HFS data set with the name of HFS.CONOLID. It is not to be SMS managed. It will be placed on DASD volsers VOL001 and VOL002. The amount of PRIMARY space to allocate is 1000 cylinders, with secondary space allocation of 250 cylinders.

```
//job.....
//EXEC DMSU
//SYSIN DD *
HFSCREATE DSN=HFS.CONOLID,
MPOINT=/tmp/ispjx11,
VOLUME=VOL001,
SPACE=(1000,250),
CYLS
/*
```


Although managing data storage requires many different tasks to be performed, the ability to generate meaningful reports probably remains as the most common. Become familiar with the reporting facilities that BrightStor CA-Disk for USS provides.

The DSCL commands supported for reporting on OS/390 UNIX files follows.

UREPORT Command and Parameters

The UREPORT command indicates that a report is to be generated for the files selected. Two types of reports are supported:

- UATT
- USPC

Note: While both report parameters are optional, one of them must be present on the UREPORT command for the command to process properly.

The following JCL statements and DSCL commands are used to produce both BrightStor CA-Disk for USS reports:

```
//jobname JOB (acct,info),etc.  
//REPORTS EXEC DMSU  
//SYSIN DD *  
SCAN UNIX  
SELECT PATH=\  
UREPORT UATT,USPC
```

UATT—Permission/Owner/Size/Change Date Report

The UATT report is a report that displays various attributes of UNIX directories and files. Following is an example of a UATT report:

```

-----
Menu Utilities Compilers Help
-----
BROUSE UNIXFL Line 0000000 Col 001 132
Command ==>
-----
***** Top of Data *****
1998.132 MAY 12, 1998 UNIX FILE SYSTEM PERMISSION/OWNER/SIZE/CHANGE DATE R E P O R T PAGE 1
SAMS:Disk - 05/390 UNIX 1.1.0

TUESDAY 2.48 PM

FILE NAME FILE TYPE -- PERMISSIONS -- OWNER GROUP FILE SIZE DIR CREATE/
(BYTES) FILE CHANGE DATE
-----
. DIR RWS R S R S OHVSKERN SVS1 11 28 DEC 16 1997
.. DIR RWS R S R S OHVSKERN SVS1 11 28 DEC 16 1997
... DIR R S R S R S OHVSKERN SVS1 16 03 MAY 1 1997
.SH_HISTOR DIR RW OHVSKERN SVS1 1889 9 12 DEC 17 1997
BIN DIR RWS R S R S OHVSKERN OHVSGRP 13 08 MAY 15 1997
DEV DIR RWS R S R S OHVSKERN OHVSGRP 16 13 APR 29 1997
SAMPLES DIR RWS R S R S OHVSKERN OHVSGRP 14 48 MAR 30 1997
THP DIR RWS RWS RWS OHVSKERN OHVSGRP 14 55 FEB 27 1998
U DIR RWS R S R S OHVSKERN OHVSGRP 20 41 APR 29 1997
USER DIR RWS R S R S OHVSKERN SVS1 11 23 FEB 26 1998
USR DIR RWS R S R S OHVSKERN OHVSGRP 16 16 MAR 27 1997
BONUSPAK2 DIR RWS R S R S OHVSKERN SVS1 19 25 APR 29 1997
BP2ARCHIVE DIR RWS R S R S OHVSKERN SVS1 19 25 APR 29 1997
-----
TOTAL NUMBER OF FILES: 13
TOTAL ACCUMULATED FILE SIZE: 1889
-----
Aa B TCPIP R 4 C 15 10:00 8/11/98
    
```

USPC—General Statistics of the UNIX File System

The USPC report is a report that displays the general status of the UNIX file system. Following is an example of a USPC report:

```

-----
Menu Utilities Compilers Help
-----
BROUSE UNIXGN Line 0000000 Col 001 132
Command ==>
-----
***** Top of Data *****
1998.132 MAY 12, 1998 UNIX FILE SYSTEM GENERAL STATISTICS PAGE 1
SAMS:Disk - 05/390 UNIX 1.1.0

TUESDAY 1.54 PM

REGULAR FIFO SYMBOLIC CHARACTER BLOCK SPECIAL UNKNOWN SIZE
DIRECTORIES FILES FILES LINKS FILES SPECIAL FILES FILES (BYTES)
-----
1382 6229 160 774 324062245
-----
***** Bottom of Data *****
-----
Aa B TCPIP R 4 C 15 10:13 8/11/98
    
```

Backup/Archive

Probably the most important data storage task today is the ability to issue and maintain proper backups of user data. Under your control, the backups you create with BrightStor CA-Disk for USS can reside on:

- Tape – the default. For an explanation, see ARC0TYPE.
- Cartridge – by overriding sysparm ARC0TYPE with a value of CART
- Disk – by overriding sysparm ARC0TYPE with a value of DISK

Note: To override default system parameter values, create a member in PARMLIB called SYSPARMS. For details, see SYSPARMS Member in PARMLIB in Appendix E “Sysparms.”

Terminology

Following is a list of terminology associated with the back up and archive procedures:

Back up – Backing up (copying) data from your UNIX file system to one or more of the supported media types and indexed in the BrightStor CA-Disk for USS Files Data Set, leaving the original file in place. For an example of this, see Backup Command Example in this chapter.

Incremental Back Up – The process of backing up only those files that have been modified since the previous backup. This type of back up runs faster than a full volume back up, and does not needlessly back up files that have not changed.

Full Back Up – The process of backing up all files, regardless of whether they have been changed. To issue a full backup, do not include any criteria statements. For example:

```
SCAN UNIX
SELECT PATH=/\
BACKUP
```

Archive—Backing up (copying) data from your UNIX file system to one or more of the supported media types and indexed in the BrightStor CA-Disk for USS Files Data Set and deleting the original file. For an example of this, see Backup Command Example in this chapter.

Midnight “Rapid Aging” Problems

When BrightStor CA-Disk for USS jobs that examine date fields are run very shortly before midnight, you must take into account the “rapid aging” that occurs at the stroke of midnight. Files that were created or used minutes ago suddenly appear as if the action took place a day ago.

When batch jobs create files, the create date used by the operating system is taken from the date the job started, not the date that each specific step of the job started. Files actually created in steps running after midnight, but part of a multistep job that began before midnight, have create dates of the previous day and immediately appear to be one day old.

Any selection tests based upon date values should take these two items into account, or many more files can be selected than were intended.

Files Data Set Server

When BrightStor CA-Disk for OS/390 UNIX backs up a file, the file is indexed in the Files Data Set. This record is inserted into the UNXINDEX subfile. Indexing is processed through a subtask so that the backup task is not affected by the Files Data Set I/O (protecting performance). This server process happens only during back up. The server is not active during archive processing or when archive is done in the same command stream as back up.

If the backup task abends in the middle of processing, there is a risk of a backed up file not being indexed. A purge and rebuild of the volume mounted at the time of the abend ensures that all files are properly indexed.

BACKUP Command and Parameters

The DSCL commands supported for backing up OS/390 UNIX files are as follows:

- BACKUP
- ARCHIVE

BACKUP

This command has the same meaning in the UNIX environment as it does in the MVS environment. That is, it backs up the selected files. The parameters EXPDT and RETPD and have the following meaning:

Parameter	Description
RETPD=	By default, files are assigned a retention period equal to the value of sysparm RETRETPD. To override the default, specify a one- to five-digit number indicating the number of days the selected files are to be retained in the BrightStor CA-Disk for USS indexes. The date calculated from this retention period cannot exceed the year 2155.
EXPDT=	By default, files are assigned a retention period equal to the value of sysparm RETRETPD. To override the default, specify a date constant in any acceptable BrightStor CA- format. For a list of valid dates, see Date Formats in the chapter "General Information."

Syntax

The syntax of the BACKUP command is as follows:

```
SET MODE=
SCAN UNIX
SELECT PATH=, FILE=, CRITERIA=
EXCLUDE PATH=, FILE=, CRITERIA=
BACKUP RETPD=, EXPDT=
```

ARCHIVE

This command performs the same action as the BACKUP command, with the optional action of deleting the original file. DISP=DELETE is the default. The following parameters are available:

Parameter	Description
RETPD=	By default, files are assigned a retention period equal to the value of sysparm RETRETPD. To override the default, specify a one- to five-digit number indicating the number of days the selected files are to be retained in the BrightStor CA-Disk for USS indexes. The date calculated from this retention period cannot exceed the year 2155.
EXPDT=	By default, files are assigned a retention period equal to the value of sysparm RETRETPD. To override the default, specify a date constant in any acceptable BrightStor CA-Disk for USS format.

Parameter	Description
DISP	This optional parameter indicates the disposition processing of the original file after it has been successfully backed up. The values are: <ul style="list-style-type: none">■ DELETE—deletes the original file after backup. This is the default value.■ KEEP—leaves the file on disk after backup. The ARCHIVE with DISP=KEEP is the same as BACKUP.

Syntax

The syntax of the ARCHIVE command is as follows:

```
SET MODE=  
SCAN UNIX  
SELECT PATH=, FILE=, CRITERIA=  
EXCLUDE PATH=, FILE=, CRITERIA=  
ARCHIVE RETPD=, EXPDT=, DISP=
```

Execution

Back up and archive are accomplished by supplying the above commands to the DMSU procedure. The DMSU procedure is an MVS procedure that is run in the MVS environment.

BACKUP Command Example

Scan the UNIX file system and back up all files that are owned by the SYSTEM Group. Do this in simulate mode first in order to verify command syntax. Keep these files for 30 days, which is the sysparm RETRETPD default.

```
//job.....  
//UNIXBKUP EXEC DMSU  
//SYSIN DD *  
SET MODE=SIMULATE  
SCAN UNIX  
SELECT PATH=/\, CRITERIA=(UNIXGONR, EQ, <SYSTEM>)  
BACKUP  
/*
```

ARCHIVE Command Example

Scan the UNIX file system and back up all files in the /usr/ispdrk3 directory, and then delete them after successful backup. Do this in simulate mode first in order to verify command syntax. Keep the archived files for three years.

```
//job.....
//UNIXBKUP EXEC DMSU
//SYSIN DD *
SET MODE=SIMULATE
SCAN UNIX
SELECT PATH=/usr/ispdrk3\
ARCHIVE RETPD=1095
/*
```

Sample BACKUP Report

The following report is produced each time you execute a DSCL BACKUP command using the DMSU catalog procedure:

```

-----
Command ==> ISPSXC3J
1998.217 AUG 05, 1998 0 5 / 3 9 0 U N I X B A C K U P
WEDNESDAY 10.18 AM SAMS:Disk - OS/390 UNIX 1.1.0 PAGE 1
-----
DIRECTOR\FILE NAME FILE TYPE CREATION TIME\DATE LAST MOD TIME\DATE DISPOSITION FILE SIZE
-----
DIR: /USER/ISPLAL2
* .SH_HISTORY FIL 9:37 FEB 18,1998 9:40 FEB 18,1998 BACKUP 3.38K
  CCOPY.C FIL 15:12 JAN 20,1998 15:15 JAN 20,1998 BACKUP 0
  DC1.L FIL 14:07 JAN 16,1998 14:07 JAN 16,1998 BACKUP 147
  DC1.V FIL 14:07 JAN 16,1998 14:07 JAN 16,1998 BACKUP 373
  DC3.L FIL 14:07 JAN 16,1998 14:07 JAN 16,1998 BACKUP 861
  DC3.V FIL 14:07 JAN 16,1998 14:07 JAN 16,1998 BACKUP 2.67K
* TEST.SCR FIL 14:06 JAN 26,1998 14:10 JAN 26,1998 BACKUP 77
  TESTDBH_DIR FIL 13:00 JAN 21,1998 15:20 JAN 21,1998 BACKUP 4.00K
  TESTDBH_PAG FIL 13:00 JAN 21,1998 15:19 JAN 21,1998 BACKUP 40.25
* TESTSCR FIL 9:23 JAN 29,1998 10:06 JAN 29,1998 BACKUP 314
  ZCOPY.C FIL 11:19 JAN 21,1998 11:22 JAN 21,1998 BACKUP 895
  BKUPPAK2.Z FIL 14:32 JAN 26,1998 14:33 JAN 26,1998 BACKUP 20.00
-----
TOTAL NUMBER OF FILES: 11
TOTAL ACCUMULATED FILE SIZE: 72007
* AN ASTERISK BEFORE DATA SET NAME INDICATES AN ERROR OCCURRED WHILE PROCESSING THE DATA SET
-----
A B TCP/IP R 4 C 15 11:33 8/05/98

```


Restoring UNIX files that have been backed up by BrightStor CA-Disk for USS can be accomplished in either one of two ways:

- Explicit Restore – using the RESTORE command
- Implicit Restore – using the FMS RECOVER command

Explicit Restore

The restore function begins by reading all of the restore commands and then sequencing them by file number within each archive tape (or archive file on disk), which is the optimum performance arrangement. All files being restored from the archives on disk are done before those archived to tape.

Multiple Files Data Set for Restore

Multiple Files Data Sets are supported for restore and DSCL recover processing. If your installation maintains multiple Files Data Sets and you want to make them all available to these functions, you should use the MFILES DD statement instead of the FILES DD statement. Add each Files Data Set name to the MFILES concatenation (up to 256 Files Data Sets are supported). The order of the concatenation determines the order of search for the data set. The first Files Data Set that contains an index record for the archived file used, even if another Files Data Set contains a more recent version of the file. Therefore, it is very important that Files Data Sets be specified in the proper order.

Note: The FILES DD statement is ignored if there is also an MFILES DD statement in the procedure. Concatenated Files Data Sets are not supported with the FILES DD statement.

Restore Condition Codes

The following return codes are set at the completion of the restore job step:

Code	Meaning
0	All files requested to be restored were restored successfully
2	Disposition processing (requested/default catalog actions) failed for one or more files
16	One or more attempted restores failed

Files Data Set Updates During Restore/Recover

When a restore/recover occurs, the Files Data Set archive record for each file is updated. This record is called the UNXINDEX record. Three fields are updated in the UNXINDEX record for each restore. The first two are the date and time of the restore. The third field is a count of the number of restores that have been done from that UNXINDEX record. The restore date is used for two functions:

- The IXMAINT function uses this date to determine if a UNXINDEX record should be removed from the Files Data Set
- The RECOVER function uses this date in conjunction with its RESDT criteria parameter

The restore time and count are currently for informational purposes only.

Use of Primary or Copy Archive Volumes

If a primary archive volume becomes unusable, the RESTORE/RECOVER functions can use the duplicate copy by disabling the primary volume. For more information, see the chapter "Index Maintenance."

Default processing of the RESTORE/RECOVER functions calls for the primary archive volume (unless it has been disabled). If it is necessary to always use the duplicate volumes, such as in a disaster recovery situation, specify sysparm RESCOPYT with a value of Y.

RESTORE Command

This is the primary command and indicates to BrightStor CA-Disk for USS that you want to restore a UNIX file. Only one file per command is supported and the complete path name must be specified (required commands followed by optional commands). PATH=, is required for the RESTORE command. The syntax is as follows:

```
RESTORE PATH=, DIRCREAT, GID=, GROUPID=, NEWPATH=, PATHATR=, PTHPERMS=,
SCRATCH, SUPRUSER, TIMEDATE=, VERSION=, SIMULATE
```

Specify the file name and its path to be restored. BrightStor CA-Disk for USS uses the complete path and file name to locate the appropriate file to restore.

Note: No pattern matching characters can be supplied.

Following is a list of parameters that can be used with the RESTORE command:

Parameter	Description
DIRCREAT	If the directory into which the file to be restored does not exist, use this parameter to indicate that the directory is to be created.
GID=,	Specify 1 to 12 numbers for GID. The RACF GROUPID will be verified and this GID will be used to update the attributes of this file. This field is optional and is mutually exclusive with GROUPID. Used with PATHATR.
GROUPID=,	Specify 1 to 8 characters for GROUPID. The corresponding RACF GID will be used to update the attributes of this file. This field is optional and is mutually exclusive with GID. Used with PATHATR.

Parameter	Description
NEWPATH=,	<p>If you want the file restored to a different path, a different name, or both, specify the new path and file name here. Also, see the description for DIRCREAT in this table.</p> <p>If a partial pathname followed by a wildcard is specified for the PATH parameter in the selection criteria, then the value specified for NEWPATH will replace only the portion of the original pathname that was specified.</p> <p>For Example:</p> <p>If the original path for the file is: <code>/a/b/c/d/e/file</code></p> <p>with a command specified: <code>SCAN UNIX select PATH=a/b/c/d/e,FILE=file RECOVER NEWPATH=/a/z</code></p> <p>then the file will be restored to: <code>/a/z/file</code></p> <p>However, if the commands are coded: <code>SCAN UNIX SELECT PATH=/a/b/c/\,FILE=file RECOVER NEWPATH=/a/z</code></p> <p>then the file will be restored to: <code>/a/z/d/e/file</code></p>
PATHATR=,	<p>This command parameter is used in conjunction with the following attribute override command parameters: PTHPERMS, GROUPLD, GID, USERID, and UID. Specify "YES" to always override the attributes for the restored file with the override command parameters. Specify "NO" (the default) to only override the attributes for the restored file when the restored path does not exist on disk and there is no path record.</p>
PTHPERMS=,	<p>Specify the Path Permissions for the restored Path. This three digit numeric field is optional. Three digits must be entered and only 0 through 7 are allowed. Used with PATHATR.</p>
SCRATCH,	<p>Use this parameter to tell BrightStor CA-Disk for USS to overwrite a file by the same name if it exists in the same location that the backed up file is to be restored. By default, an existing file is not overwritten.</p>
SIMULATE,	<p>Specify this parameter to cause the restore of a file to be simulated (not physically restored). This parameter is optional. If more than one file is to be simulated, this parameter must be specified for every command that is to be simulated. Although not recommended, simulated restores can be intermixed with live restores.</p>

Parameter	Description
SUPRUSER	Restore processing can require Superuser authority (UID 0) to perform certain functions. These functions can be to update file attributes, delete a file, or create special files. If the user is authorized to invoke the UNIX Superuser authority (UID 0), then specification of this parameter invokes it and thus allows the actions to happen. If the user is not authorized to invoke Superuser authority, specification of this parameter is ignored.
TIMEDATE=,	<p>This parameter can be used to identify a specific version of the file within the archives, rather than the relative version as supplied on the VERSION= parameter. Using it is appropriate if a possibility exists that another version of the same file is placed in the archives before the restore can be completed.</p> <p>The format for this parameter is “hhmmdate” where “hhmm” is the 24-hour time and “date” is any recognized BrightStor CA-Disk for USS date format. Correct values can be obtained by listing the BrightStor CA-Disk for USS index entries for the data set through batch or TSO.</p>
UID	Specify 1 to 12 numbers for UID. The RACF USERID will be verified and this UID will be used to update the attributes of this file. This field is optional and is mutually exclusive with USERID. Used with PATHATR.
USERID=,	Specify 1 to 8 characters for USERID. The corresponding RACF UID will be used to update the attributes of this file. This field is optional and is mutually exclusive with UID. Used with PATHATR.
VERSION=,	To restore a version other than the most recent file contained in the BrightStor CA-Disk for USS archives, specify from -1 to -99 to get the successively older version of the file.

Execution

Explicit restore accomplished by supplying the above commands to the RESTOREU procedure. The RESTOREU procedure is an MVS procedure that is run in the MVS environment.

Examples

Restore the latest version of a file named myfile.txt that was contained within the directory path name of /user/ispjxl1. Restore this file to its original file position and delete the copy of the file that is currently in that location.

```
//job.....
//RESTORE EXEC RESTOREU
//SYSIN DD *
RESTORE PATH=/user/ispjxl1/myfile.txt,SCRATCH
/*
```

Restore the -1 version of a file named myfile.txt that was contained within the directory path name of /user/ispjxl1. Restore this file to a new path named /user/ispjxl1/delta. If the new path does not exist, then create it. If the path does exist and a copy of the file already exists in that location, then delete it.

```
//job.....
//RESTORE EXEC RESTOREU
//SYSIN DD *
RESTORE PATH=/user/ispjxl1/myfile.txt,
NEWPATH=/user/ispjxl1/delta/myfile.txt,
SCRATCH,DIRCREAT,VERSION=-1
/*
```

Sample Restore Report

The following is a sample report generated from the RESTORE command:

```

File Edit Transfer Options Connection Macro Window Help
-----
File Edit Confirm Menu Utilities Compilers Test Help
EDIT      ISPJKL5J                               Columns 00001 00124
Command ==>
***** Top of Data *****
000100          JOB=ISPJKL5J                      FILES DSN=ISPJKL5.UNIX02.FILES
000200          STEP=S1RECVR                      PARMLIB DSN=ISPJKL5.LK136779.PARMLIB
000300                                               DSN=DISK920.QUAL.PARMLIB
000400                                               DSN=DISK920.UPDT.PARMLIB
000500                                               DSN=DISK920.BASE.PARMLIB
000600          1998.222 AUG 10, 1998                MONDAY    3.44 PM
000700
000800 1998.222 AUG 10, 1998                RESTORED UNIX FILES                                PAGE 1
001000 MONDAY    3.44 PM                                SAMS:Disk - 08/390 UNIX 1.1.0
001200
001300 PATH / NEW PATH                                FILE ARCHIVED  CREATION  FILE
001400                                               TYPE TIME DATE  TIME DATE  SIZE
-----
001600 /USER/ISPJKL5/DIROUT.TXT                       FIL 1140 1998216 1155 1998142 12.26K
001700 /USER/ISPJKL5/JOHNOUT.TXT
001800
001900                                               TOTAL          TOTAL
002000                                               FILES          BYTES
002100 RESTORED...                                -----
002200                                               1              12.26K
***** Bottom of Data *****
As B TCP/IP R 18 C Z 9:15 8/11/98

```

FMS Recover

FMS Recover uses the DSCL command syntax to achieve the process of implicit UNIX file recovery. It has a rich set of parameters that allow for granular tailoring of the file recovery process.

The DSCL commands to use for FMS Recovery are as follows:

- SET MODE=SIMULATE
- SCAN UNXINDEX
- SELECT PATH=,FILE=,CRITERIA=
- EXCLUDE PATH=,FILE=,CRITERIA=
- RECOVER

Condition Codes

The FMSU PROC returns the following JCL condition codes:

Code	Meaning
0	Successful completion.
4	No files selected or 1 or more files not restored.
8	Files resource could not be acquired (for example, security or enqueues).
12	Unrecoverable error encountered in files.
16	Command error.

Commands

Following is a list of DSCL commands for use with FMS Recovery and their descriptions:

SET MODE=SIMULATE

Use the SET command to define environmental values that are in effect for the duration of this execution only. If used, you can put the SET command anywhere in the command stream before the first action command. For clarity, we recommend that you specify this command before all other commands.

As with all BrightStor CA-Disk for USS commands, the SET command can be commented out (nullified) by placing an asterisk "*" in column 1 of the line containing the command.

SCAN UNINDEX

This command instructs BrightStor CA-Disk for USS to scan the indices that hold a record of all the UNIX files backed up.

SELECT/EXCLUDE

For a description of how the SELECT and EXCLUDE commands operate, see the chapter "Reports." However, new parameters were added to support the UNIX file system. Following is a list of parameters that can be used with the SELECT and EXCLUDE commands:

Parameter	Description
PATH=	This is where the path name or pattern of up to 255 characters to scan is placed. Up to ten path names can be specified. Example: PATH=/dir1/dir2/my????file
FILE=	If you want to limit the selection/exclusion of file names, specify this parameter with up to ten names. Names can be up to 255 characters in length. Pattern matching is allowed.
CRITERIA=	Specify the selection criteria to be met before any BrightStor CA-Disk for USS action occurs. The criteria parameter is specified by a minimum of three subparameters as follows: (operand1, operator, operand2) where the operands can consist of an expression as defined below. The maximum number of subparameters allowed is controlled by sysparm VSCRIMAX.

For logical operators, you can specify any of the following FMS criteria operators:

Operator	Meaning	Operator	Meaning
EQ	equal to	LE	less than or equal to
NE	not equal to	GT	greater than
LT	less than	GE	greater than or equal to

For operands, you can specify any of the following:

- Decimal number constants, such as 10 or 1000.
- Date constants in any acceptable BrightStor CA-Disk for USS format, such as 96200, 96.200, 96/200, 19JUL96, 07/19/96, 1996200, 1996.200, 19JUL1996, 1996/200, or 07/19/1996. If BrightStor CA-Disk for USS has trouble differentiating between a digit Julian date constant, such as 96200, and a decimal number constant, it issues an appropriate message and asks you to use another format.
- Expressions that use addition (+) or subtraction (-) to combine a constant value with a field name or other constant value. For example, the expression TODAY-30 represents the date 30 days ago.
- Any of the DSCL select criteria field names listed in the following table:

Field Name	Definition	Field Name	Definition
TODAY	Today's date	UNIXFLSZ	File size in bytes
UNIXARDT	Backup date	UNIXBLSZ	Size of the blocks
UNIXARTM	Backup time	UNIXBLAL	Number of blocks used
UNIXDSKY	Archive key	UNIXFTYP	File type
UNIXRSDT	Restore date in UNXINDEX record	UNIXLSAC	Last access date
UNIXRSTM	Restore time in UNXINDEX record	UNIXMODT	Last modification date
UNIXVRSN	Version number of the UNXINDEX record	UNIXSTCH	Last change date
UNIXUON R	User owner of the file	UNIXCRDT	Creation date
UNIXGON R	Group owner of the file	UNIXRFDT	Last reference date

DSCL SELECT CRITERIA Field Name Descriptions

Following is a list of DSCL SELECT CRITERIA field name descriptions:

TODAY—Today's date (date format)

This value is derived from the system clock. Comparison dates used in conjunction with this criterion can be entered in any acceptable BrightStor CA-Disk for USS format. For example:

```
CRI=(UNIXARDT, EQ, TODAY)
```

UNIXARDT—Backup date (date format)

This is the date the file was backed up. For example:
CRI=(UNIXARDT, EQ, 1998.035)

UNIXARTM—Backup time (time format)

This is the time the file was backed up. A 24-hour time clock is used. For example:
CRI=(UNIXARTM, EQ, 0830)

UNIXDSKY—Archive Key (Character Literal)

This is the key of the ARCHVOL record holding the backed up UNIX files. For example:
CRI=(UNIXDSKY, EQ, <@99997>)

UNIXRSDT—Last RESTORE Date (date format)

This is the date the file was last restored. For example:
CRI=(UNIXRSDT, EQ, 1998.035)

UNIXRSTM—Last RESTORE Time (time format)

This is the time the file was last restored. A 24-hour time clock is used. For example:
CRI=(UNIXRSTM, EQ, 0830)

UNIXVRSN—Version # of the UNXINDEX record

Use this parameter to select the version of the files to restore. Specify as negative or 0 - the most current version. For example:
CRI=(UNIXVRSN, EQ, -1)

UNIXUONR—User owner (Character Literal)

This is the user owner of the file. For example:
CRI=(UNIXUONR, EQ, <ISPJXL1>)

UNIXGONR—Group owner (Character Literal)

This is the group owner of the file. For example:
CRI=(UNIXGONR, EQ, <SYSTEM>)

UNIXFLSZ—File size in bytes (numeric)

This is the size of the file in bytes. A value of up to 4,294,967,295 is supported. For example:

```
CRI=(UNIXFLSZ,GT,1000000)
```

UNIXBLSZ—Size of the block of the file in bytes (numeric)

This is the size of the blocks of the file in bytes. A value of up to 4,294,967,295 is supported. For example:

```
CRI=(UNIXBLSZ,GT,4096)
```

UNIXBLAL—Number of blocks used (numeric)

This is the number of blocks the file is using. A value of up to 4,294,967,295 is supported. For example:

```
CRI=(UNIXBLAL,GT,1000000)
```

UNIXFTYP—File type (character literal)

This is the type of file. For example:

```
CRI=(UNIXFTYP,EQ,any valid value)
```

UNIXLSAC—Last access date (date format)

This is the date the file was last accessed. For example:

```
CRI=(UNIXLSAC,EQ,1998.035)
```

UNIXMODT—Last modification date (date format)

This is the date the file was last modified. For example:

```
CRI=(UNIXMODT,EQ,1998.035)
```

UNIXSTCH—Last change date (date format)

This is the date when the file status was changed. For example:

```
CRI=(UNIXSTCH,EQ,1998.035)
```

UNIXCRDT—Creation date (date format)

This is the creation date of the file. For example:

```
CRI=(UNIXCRDT,EQ,1998.035)
```

UNIXFRDT—Last reference date (date format)

This is the date the file was last referenced.

```
CRI=(UNIXFRDT, EQ, 1998.035)
```

RECOVER

This is the action command and indicates to BrightStor CA-Disk for USS that multiple UNIX files are to be restored. The syntax is as follows:

```
RECOVER NEWPATH=, DIRCREAT, GID=, GROUPID=, PATHATR=, PTHPERMS=, SCRATCH,  
SUPRUSER, UID, USERID
```

Following is a list of parameters than can be used with the RECOVER command:

Parameter	Description
DIRCREAT	If the directory into which the file to be restored does not exist, this parameter indicates that the directory is to be created.
GID=,	Specify 1 to 12 numbers for GID. The RACF GROUPID will be verified and this GID will be used to update the attributes of this file. This field is optional and is mutually exclusive with GROUPID. Used with PATHATR.
GROUPID=,	Specify 1 to 8 characters for GROUPID. The corresponding RACF GID will be used to update the attributes of this file. This field is optional and is mutually exclusive with GID. Used with PATHATR.
NEWPATH=	If you want to have the files restored to a different path, specify the new path here. This value cannot include a pattern character; it must be an explicit path name. Also see the description for SUPRUSER in this table.
PATHATR=,	This command parameter is used in conjunction with the following attribute override command parameters: PTHPERMS, GROUPID, GID, USERID, and UID. Specify "YES" to always override the attributes for the restored file with the override command parameters. Specify "NO" (the default) to only override the attributes for the restored file when the restored path does not exist on disk and there is no path record.
PTHPERMS=,	Specify the Path Permissions for the restored Path. This 3-digit numeric field is optional. Three digits must be entered and only 0 through 7 are allowed. Used with PATHATR.
SCRATCH,	Using this parameter instructs BrightStor CA-Disk for USS to overwrite a file by the same name if it exists in the same location that the backed up file is to be restored. By default, an existing file is not overwritten.

Parameter	Description
SIMULATE,	Specify this parameter to cause the restore of a file to be simulated (not physically restored). This parameter is optional. If more than one file is to be simulated, this parameter must be specified for every command that is to be simulated. Although not recommended, simulated restores can be intermixed with live restores.
SUPRUSER	Restore processing can require Superuser authority (UID 0) to perform certain functions. These functions can be to update file attributes, delete a file or create special files. If the user is authorized to invoke the UNIX Superuser authority (UID 0), then specification of this parameter invokes it and thus allows the actions to happen. If the user is not authorized to invoke Superuser authority, specification of this parameter is ignored.
UID	Specify 1 to 12 numbers for UID. The RACF USERID will be verified and this UID will be used to update the attributes of this file. This field is optional and is mutually exclusive with USERID. Used with PATHATR.
USERID=,	Specify 1 to 8 characters for USERID. The corresponding RACF UID will be used to update the attributes of this file. This field is optional and is mutually exclusive with UID. Used with PATHATR.

Execution

FMS Recover is accomplished by supplying the above commands to the FMSU procedure. The FMSU procedure is an MVS procedure that is run in the MVS environment.

Examples:

Scan the BrightStor CA-Disk for USS Files Data Set and recover the entire UNIX file system. Delete files that can already be there and create directories if they do not exist. Run in simulate mode first to make sure that the results you want are achieved.

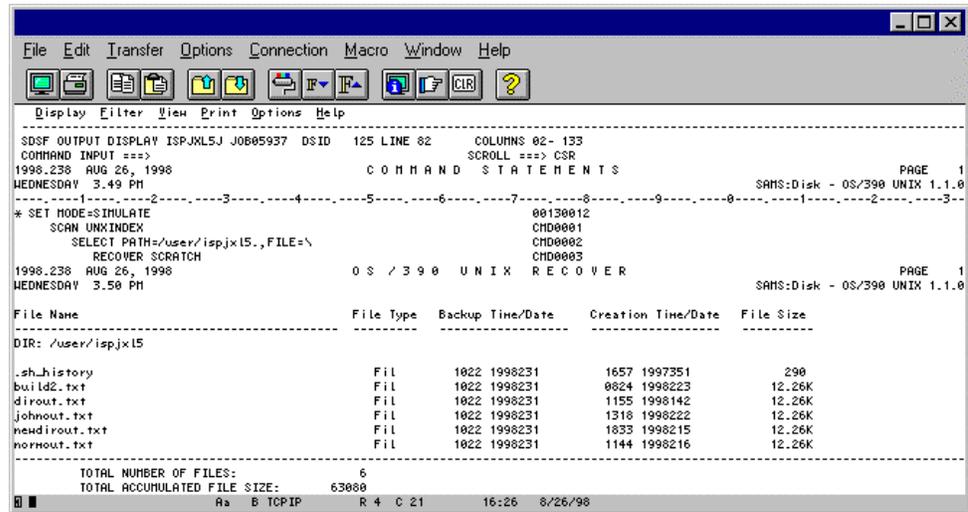
```
//job.....
//UNIXFMS EXEC FMSU
//SYSIN DD *
SET MODE=SIMULATE
SCAN UNXINDEX
SELECT PATH=/\
RECOVER SCRATCH,DIRCREAT
/*
```

Scan the BrightStor CA-Disk for USS Files Data Set. Recover all the files that are Symbolic Links. Run in simulate mode first to make sure that the results you want are achieved.

```
//job.....
//UNIXFMS EXEC FMSU
//SYSIN DD *
SET MODE=SIMULATE
SCAN UNXINDEX
SELECT PATH=/\,CRITERIA=(UNIXFTYP,EQ,SYM)
RECOVER
/*
```

Sample Recover Report

The following is a sample report generated from the RECOVER command:



This function provides a convenient means to manage the index entries for archived files and the volumes that contain them.

Commands Available	JCL Procedure Executed
MERGE	MERGEU

The MERGE function helps you manage the BrightStor CA-Disk for USS archive files that are on various disk, tape, and tape cartridges.

Overview

When files are backed up or archived, each is assigned a retention period (expiration date). By definition, then, 100 percent of the files are unexpired on the day they are first archived. To make the most efficient use of both disk and tape, BrightStor CA-Disk for USS (by default) packs all of these archived files into a single file (if archiving to tape), or a single data set allocated on disk.

As each day passes, however, some of these files can expire, creating “dead space” in the archives. Given enough time, all of the files will expire and the tape can be recycled or the archive data set on disk can be deleted to reclaim the space. The IXMAINT function accomplishes this. For more information, see the chapter “Index Maintenance.”

Each archive volume (tape or disk data set) starts out as packed full of 100 percent unexpired files. As the days pass, this eventually drops to zero percent; that is, all files are expired. Important questions to ask yourself are:

- How fast does this take place?
- Can you afford to wait until it drops all the way to zero percent?

In many cases, tapes and disk space are wasted if you do. The archives need to be consolidated to reclaim the space.

This is the function that merge performs. Each merge run consolidates the unexpired files from one or more tape volumes, or archive data sets on disk, onto new archive volumes, thus freeing up tape volumes, disk space, or both. This reduces the space required to store the BrightStor CA-Disk for USS archives, as well as eliminates disk archive data sets with a large percentage of expired files. After the unexpired files are successfully merged forward, the input data sets on disk are deleted, and input tapes are returned to scratch status – disk space is reclaimed and tapes are made available for reuse. All expired BrightStor CA-Disk for USS index entries are dropped from the archives and unexpired entries are updated to reflect their locations on the new tape or disk archives.

Specifying What to Merge

Since it is very common to have many archive volumes, it is not practical to process all of them in a single merge execution. The time and machine resources required prohibit execution. To help you control the merge process, command parameters can be specified to control which archive files are to be merged. By running the merge in simulate mode and asking for a tape pull list, you can determine prior to the live merge run precisely what the input to the merge process will be. Following is a list of these command parameters:

Parameter	Description
TYPES=	The most commonly used command parameter. This parameter allows you to limit the scan of archive data sets to those residing on specified device types. For example, you can specify that a merge of only the disk archive data sets be run daily, since it processes without operator intervention and makes efficient use of disk space.
PERCENT	This parameter allows you to specify the threshold of active, or unexpired, space that is to be maintained in an archive data set. When the value falls below the percent value specified, the volume is selected for merge processing. This allows archive tapes, archive disk data sets, or both, that are adequately used to be skipped.
MAXVOLS	This parameter can be used to limit the merge process to a maximum number of archive volumes. Processing terminates when the maximum volume limit is reached. A warning message is issued to indicate that some volumes are bypassed. The next execution of merge processes the bypassed volumes.

Parameter	Description
MERGE	This parameter can be further limited to processing only archive volumes whose names match those specified in the TDSNAMES= parameter. The NOMERGEIN parameter excludes archive volumes created in previous Merge or Xcopy runs from being selected as input to merge. INCLUDE and EXCLUDE parameters further refine the input by permitting specific archive volumes to be listed and acted upon.
DAYSSOLD	This parameter limits the input based upon the “age” of the archive volumes; that is, how long ago were they created.
FDAYSSOLD	This parameter limits the input based upon the “age” of the archive volumes; that is, how long ago were they created.
CREDIT	This parameter limits the input based upon the “age” of the archive volumes; that is, how long ago were they created.
FCREDIT	This parameter limits the input based upon the “age” of the archive volumes; that is, how long ago were they created.
DATE=	This parameter provides additional flexibility based upon the archive expiration dates. It instructs merge to pretend that today is the date specified, rather than the current date, which obviously affects whether merge thinks that a data set or volume has expired
EXPDAYS	This parameter provides additional flexibility based upon the archive expiration dates. It excludes volumes that are about to expire (why waste processing resources if they expire and be deleted “tomorrow?”).
MERGE	This parameter, by default, bypasses multi-volume TAPE files with message 4005; however, it does include multi-volume DISK files.
INCLUDE	Use this parameter if you want to process multi-volume TAPE files. For a description, see INCLUDE= in Syntax in this chapter.

For more information on each of these parameters, see CREDIT= in Syntax in this chapter.

Specifying Merge Output Options

Great flexibility in controlling the creation of new archive volumes is provided through command parameters and system parameters.

By using the LIMITS parameter, you can specify that archive data sets are to be written to up to five new archive volumes, each containing only files within a given range of expiration dates.

You can also control the device types to which the new archives are written through the dynamic allocation sysparms MERPnTYP and MERPnNAM. The most suitable archive medium for each range can also be specified. For example, one range can be written to disk archive data sets, another to tape, and yet another to tape cartridge.

Merge also allows you to produce duplicate copies concurrently with the creation of the new archive volumes. This is accomplished through dynamic allocation by specifying sysparms MERCnTYP and MERCnNAM. This process is sometimes referred to as true duplexing, and is the most efficient way to produce duplicate copies. The copies can also go to different device types, much like the primary copies.

You can, however, create additional copies by executing an extra job step following the primary merge. If true duplexing is done, this extra step creates the triplicate copy. If true duplexing is not done, this extra step creates a duplicate copy.

Implementation Concerns

Running a merge can be a time-consuming process, especially when a large number of tapes are involved. You can avoid running it for some tape volumes by maximizing tape usage at archive time and ensuring that expiration dates for data sets written to each tape are not spread over a large period of time. This allows tapes to be removed from the archives in a timely manner when all files on each tape have expired. Use the index maintenance function PATHDELETE (or possibly VOLDELETE) to remove expired files and volumes from the archive index.

The merge process performs functions similar to the index maintenance PATHDELETE (and VOLDELETE) command, in that they both delete expired files from the archives. However, the index maintenance functions perform at a much more efficient rate. To improve merge performance, run the PATHDELETE (or VOLDELETE) index maintenance command prior to running merge.

Note: Sysparms ARCHPACK and ARCDKBZ are used by merge in the same manner in which archive processing uses them. The packing option should probably never be turned off (ARCHPACK specified with a value of N) if merge is creating tapes, since it merges forward only one data set per tape, creating an enormous number of new archive tapes, and defeating the purpose of merge. ARCDKBZ should be set to an efficient value common to all the output devices you are using.

Operation Concerns

Merges run on a scheduled basis, based on the needs of your installation. A disk-to-disk merge can be run on a daily basis because it requires no operator intervention and can quickly free up DASD space. For tapes, merge is usually run on a weekly or monthly basis, depending on the number of archive volumes created and the amount of space that can be recovered from them. Not all BrightStor CA-Disk for USS archive volumes need to be run through merge. Process only those volumes that are wasting a large amount of space, or those archive data sets on disk that can now be moved to tape for the remainder of their retention period.

During a merge run, input 3480 cartridges use the DD statement //ARCHIVER, while input disk drives are referenced using the DD statement //ARCHIVED.

Depending on what is being merged, one tape, one 3480 drive, or both, is needed for input. Since the ARCHVOLS records contain device type information and data set names, BrightStor CA-Disk for USS can dynamically allocate each of these input types as appropriate. JCL can be provided for tape and 3480s.

If the output is to disk, BrightStor CA-Disk for USS always allocates the disk archive data set dynamically. To allocate all primary output devices dynamically, the data set names and the device types must be supplied in sysparms MERPnNAM and MERPnTYP. Concurrent duplicate output copies can also be created and dynamically allocated by supplying proper values for sysparms MERCnNAM and MERCnTYP. If the concurrent copies are not wanted, sysparm MERCnTYP must be set to NULL. For more information about these sysparms, see their sysparm descriptions in Syntax in this chapter.

Tape or 3480 devices can be allocated through JCL by supplying //ARCHIVE1, //ARCHIVE2, DD statements for the primary output, and //ARCHIVC1, //ARCHIVC2, for the copy devices.

You can optionally specify a system parameter that allows the computer operator to terminate merge prematurely (a graceful shutdown). A message is issued to the operator at the beginning of the merge process. If the operator replies to the message to indicate that processing is to be stopped, merge terminates when the current input volume has completed processing. Processing completes normally, and all reports and statistics are generated as normal. This option permits you to start merge and run until the resources are needed for something else. Thus merge can be run in otherwise unused machine time. Specify this option through sysparm MERREPLY.

TAPEPULL Processing

Some installations require that all tapes be pulled from the tape library and be provided with the job that uses them before it goes into execution. A facility to generate such a list for merge processing is provided. This facility is invoked through the TAPEPULL parameter.

Two steps are required:

1. Run MERGE with the SIMULATE and TAPEPULL=GEN parameters. Note that if you fail to specify SIMULATE, BrightStor CA-Disk for USS forces a simulation run for you. This accomplishes two things:
 - It produces the list of tape volumes that are used as input for the LIVE merge run, enabling operators to pull them from the library.
 - It saves the list in a disk data set, such that a subsequent LIVE run can use them as input.

Note: The tape pull list specifies only tape volumes to be used for input for the merge run. If output is directed to tape, appropriate tape volumes are called for during the live run, as they are needed.

There is a significant distinction between the printed tape pull list and the list written to disk. The printed list does not contain any keys of disk archive data sets that are selected for processing. The list written to disk, which can be used by merge in a subsequent live run, contains these keys. It is assumed that the printed list is used to actually allow tape volumes to be pulled and readied prior to the live run.

The TAPEPULL list written to disk only includes the first volume of multi-volume files. The TAPEPULL printed list includes all volumes for operations use.

2. Run MERGE without the SIMULATE parameter and with TAPEPULL=USE specified.

Note that when running merge using the TAPEPULL option, the volume selection is done during step 1 when the list is being generated. Therefore, all selection criteria desired should be specified on the command at this time.

Note: All parameters specified on TAPEPULL=GEN **must also be specified on** TAPEPULL=USE. The only additional parameters that can be added to the command are COPY and EXCLUDE.

When using the TAPEPULL function, an additional JCL statement must be supplied in the merge JCL procedure. It can be supplied as a JCL override or be included directly in the JCL procedure. The DD statement "TAPEPULL" must point to a sequential data set with the attributes RECFM=FB, LRECL=80, BLKSIZE=4000. Allocate the data set prior to the merge run.

MERGE, by default, bypasses multi-volume TAPE files with message 4005. It includes multi-volume DISK files. If you want to also process multi-volume TAPE files, use the INCLUDE command parameter. For a description, see INCLUDE= in Syntax in this chapter.

Use the following DD statement as an example. It must be included for both the TAPEPULL=GEN and TAPEPULL=USE options.

```
//TAPEPULL DD DSN=SAMS.DISK.MERGE.TAPEPULL,DISP=OLD
```

Abend/Restart Considerations

During merge processing, the DSNINDEX is updated to reflect the new volume of residence as each file is successfully copied to its new volume. If the merge step terminates abnormally, the ARCHVOLS file record for the current output has incomplete statistical fields. These are normally updated during archive CLOSE processing. The absence of this information does not prevent the restoration of a file from the output, nor does it cause any problems during processing of the output in a subsequent merge.

You can correct the incomplete statistical information in the ARCHVOLS record by purging it from the Files Data Set (PURGE command), and then using the REBUILD command to reconstruct the ARCHVOLS and UNXINDEX records from the archive data set, whether on tape or disk. If the archive data set resides on disk, use the default of SCRATCH=NO during purge processing so that REBUILD can take place.

Merge processing can be restarted by submitting the same JCL as used in the failing job. If the same selection tests are specified, processing continues with the data set being processed when the failure took place.

BrightStor CA-Disk for USS considers any tape it has written to be a good tape, even though an abend occurred during processing. If you are using a tape management system in your installation, make sure that it does not treat such tapes as scratch tapes. This is not an issue if you use the EDM option of your tape management system. Some tape management systems (TLMS for example), can default to giving "abend tapes" a one-day retention period. If this is the case, either change your tape management system to not do this for BrightStor CA-Disk for USS tapes, or take clerical means to properly save the BrightStor CA-Disk for USS tapes after an abend.

Simulate Mode Considerations

By using the SIMULATE parameter on a MERGE job, BrightStor CA-Disk for USS instructs MERGE to execute without actually mounting any ARCHVOLS or moving any UNXINDEX records. The result is a report indicating which UNXINDEX and ARCHVOL records are selected in a live run.

Because output devices are not allocated during simulations, BrightStor CA-Disk for USS cannot accurately predict the number of output ARCHVOLS in a live run. If you only want to know which UNXINDEX record, ARCHVOL record, or both, are selected for input by MERGE, then this is not a concern. However, take certain steps if your intent is to determine the approximate number of ARCHVOLS that are required for output.

MERGE can simulate specific device types to predict volume capacity, and when a volume switch would occur. This is accomplished by specifying the appropriate DYNnUNIT sysparm with the BrightStor CA-Disk for USS value for the device type to be simulated as indicated below:

The MERP1TYP (and MERC1TYP, if used) sysparm should specify one of the following Media Option types:

Media Option	Sysparm in Effect
3480	DYNCUNIT
DYN1	DYN1UNIT
DYN2	DYN2UNIT
DYN3	DYN3UNIT

Specify the DYNnUNIT sysparm, related to the above Media Options, with one of the cartridge device types.

Note: The simulated devices represented in this table only work when sysparm CARTCALC is specified with a value of Y. If CARTCALC is set to N, all 3480/3490 devices use CARTFEET.

Processing Logic

The input to a merge process, simulated or live, is the unloaded UNXINDEX file sorted by file number within each archive data set (by archive key). All of the various input selection tests are applied to determine which archive volumes (tapes or disk archive data sets) are to be processed.

The ARCHVOLS record corresponding to each of the selected volumes is then examined (a flag bit is tested) to ensure that the volume has been properly closed. If it has not, it could be in current use, in which case it should not be processed by merge. This condition can occur if archive and merge jobs are running at the same time. To maintain data integrity under these conditions, even if the jobs are running concurrently over midnight, BrightStor CA-Disk for USS does not merge an “open volume” unless its creation date is less than today's date minus one (that is, CREDIT < TODAY-1).

The UNXINDEX records are sorted by file number (position) within each archive volume. This permits all files being merged to be located by “skipping forward,” whether this is a tape spin or a seek of the disk heads. Each unexpired file is merged forward to a new archive volume and its index entry updated to reflect its new location (live mode only). Its expiration date in conjunction with the LIMITS= parameter determines which of the new output devices is used. If its expiration date is greater than the highest limit specified, it is merged to the archive data set associated with the highest limit.

Index entries for expired files are deleted. When all index entries for a volume have been either deleted or updated to reflect their new locations on a new archive volume, the old ARCHVOLS record is deleted as well. Thus, when any of the moved files are to be restored, BrightStor CA-Disk for USS calls for the new archvol.

The expiration date assigned to each archive volume created by merge is taken from the sysparm value for DYNEXPDT, or from the value you have specified in overriding JCL. The default value of DYNEXPDT is the “never expire” Julian date of 99365. If you specify a different non-zero value, it is used. In an EDM environment, DYNEXPDT leaves its default value that allows DSNINDEX and ARCHVOL records to expire correctly.

Merge processing relies on the specification of sysparms to determine the average length of archive tape volumes being used. The information is needed to determine the amount of space used on each ARCHVOL.

Note: Candidates to be merged forward are determined solely by the expiration date of the file index entry, and not by the expiration date of the volume index entry. This means that if the EXPDT or RETPD parameters were “hard-coded” on the //ARCHIVE0 DD statement during an archive run, it is possible to have an unexpired file on an expired archive volume. The unexpired file is still called for in merge processing to ensure the integrity of merge. No unexpired file is dropped. If the expired tape has been written over or the archive data set on disk has been deleted, errors result when trying to locate the data set to be merged. If “hard-coded” JCL expiration dates are being used, run the VOLDELETE command of IXMAINT before running merge to prevent the above problem from occurring.

Merge processing creates report records for all volumes and data sets processed, indicating the actions taken. They are created concurrently with the MERGE and INDEX UPDATE step, but must be resorted into the proper sequence for the report writer.

JCL—MERGE Command

The MERGE command shown with the sample JCL below selects as input to the merge process those archive volumes that are less than 70 percent used. “Archive volumes” refers to both archive tape volumes and disk archive data sets. Less than 70 percent used means that the unexpired files occupy less than 70 percent of either the length of the tape, or the disk space allocated to the disk archive data set. The unexpired files from these selected volumes are moved to one of three new output units, based on expiration date. Those moved to either of the first two expiration groups have a duplicate copy created concurrently. Those moved to the third (highest) expiration group do not have a duplicate created.

Following is sample JCL for the MERGE Command:

```

File Edit Confirm Menu Utilities Compiler Test Help
-----
EDIT                               .CNTL(MERGE) - 01.00           Columns 00001 00072
Command ==>                        Scroll ==> CSR
***** ***** Top of Data *****
000100 //JOB(XXXX,XXX), 'NNNNNNN', CLASS=X, MSGCLASS=T, NOTIFY=&USERID
000200 //TAPEMERG EXEC MERGE
000300 //MERGE.SYSPARM DD *
000400 MERP1NANDMS.MERGPR11
000500 MERP1TYPDISK
000600 MERP1NANDMS.MERGCOP1
000700 MERP1TYP1TAPE
000800 *
000900 MERP2NANDMS.MERGPR12
001000 MERP2TYP1TAPE
001100 MERP2NANDMS.MERGCOP2
001200 MERP2TYP1TAPE
001300 *
001400 MERP3NANDMS.MERGPR13
001500 MERP3TYP3480
001600 MERP3TYPNULL
001700 //MERGE.SYSPARM DD *
001800 MERGE PERCENT=70,LIMITS=(90,180,365)
***** ***** Bottom of Data *****
Aa A TCPIP R 4 C 15 14:40 3/02/98

```

Condition Codes

The following condition codes are returned from the stepname MERGE during the MERGE process:

Condition Code	Descriptions
0	Successful execution.
4	Warning messages issued.
8	An error terminated processing.
16	A critical error caused merge termination.

MERGE Command and Parameters

Following is a description of the MERGE command and its associated parameters. Only one MERGE command can be supplied per execution.

SYNTAX

The syntax of the MERGE command is as follows:

```
MERGE
SIMULATE, TYPES=, LIMITS=, PERCENT=, DATE=, EXPDAYS=, TAPEPULL=, MAXVOLS=, TDSNAMES=,
NOMERGEIN, INCLUDE=, EXCLUDE=, FCREDIT=, FDAYSOLD=, CREDIT=, DAYSOLD=, LISTALL, VERIFY
```

Following is a list of parameters that can be used with the MERGE command:

Parameter	Description
SIMULATE	This parameter causes MERGE to execute without mounting any tape volumes or moving any archived data sets. It processes the DSNINDEX and ARCHVOLS files and produces reports indicating what would have occurred in a live run.
TYPES=	This optional parameter can be specified to limit processing to those archive data sets residing on specified device types. You can specify TAPE, DISK, 3480, DYN1, DYN2, DYN3, or any combination. If this parameter is omitted, processing defaults to all device types.
LIMITS=	This optional parameter governs the distribution of data sets on the new output volumes. It specifies a list of one to five limit values of either Julian dates or retention periods in ascending sequence. The retention periods are 1- to 5-digit numbers and the Julian dates are 7-digit dates (yyyyddd). Each value represents a boundary for the data sets that are to be merged to that output group. When the expiration date of a data set is greater than a boundary limit, it is merged to the next output group, with the exception of the last boundary limit. If the expiration date of a data set exceeds even the last boundary value, it is still merged to the last output group. A maximum of five entries is supported.
PERCENT=	This optional parameter indicates the minimum percent of used space to be maintained on each archive volume. When the unexpired data sets on a volume account for less than this percentage of the total volume, the volume is selected for merge processing. <ul style="list-style-type: none"> ■ PERCENT=0—Select no archive volumes based on the PERCENT parameter. Use the INCLUDE parameter for input volume specification. ■ PERCENT=nn—Select the volume when its percent used drops below nn. ■ PERCENT=100—This is the default. Process all archive volumes regardless of the space used on them.

Parameter	Description
DATE=	This optional parameter specifies that a date in an accepted BrightStor CA-Disk for USS format be used instead of the current date in determining which archived data sets have expired.
EXPDAYS=nnnn	<p>This parameter excludes archive volumes from the merge processing if either the archive volume will expire within nnnn days or all of the data sets on the archive volume will expire within nnnn days, where nnnn is a number of days from the current date or the date specified on the DATE parameter.</p> <p>The current date plus this number of days is compared to the archive volume expiration date and the highest expiration date of all the data sets on the volume. If the date is greater than either of these expiration dates, the volume is excluded from MERGE processing.</p>
TAPEPULL=	<p>This is an optional parameter. For a complete description of the TAPEPULL support, including the need for the // TAPEPULL DD statement, see Operation Concerns and TAPEPULL Processing in this chapter.</p> <p>Specify TAPEPULL=GEN to generate a hardcopy list of the archive tape volumes selected for merging. This list is also saved in a disk data set and used with the TAPEPULL=USE option as described below. Normally the SIMULATE parameter is supplied along with this parameter.</p> <p>Specify TAPEPULL=USE to limit processing to those input volumes listed when the TAPEPULL=GEN parameter was specified on a previous execution. The only valid additional, optional parameters when processing in this mode are COPY, LIMIT, and EXCLUDE.</p>
MAXVOLS=xxx	This optional parameter specifies a maximum number of input volumes to be processed for this execution of MERGE. The default value for this parameter is 150.
TDSNAMES=dsnamelist	This optional parameter specifies 1 to 20 archive volume data set names or patterns for which processing is to be limited.
NOMERGEIN	This optional parameter causes archive volumes created during previous merge runs to be excluded, unless also in an INCLUDE list.

Parameter	Description
INCLUDE=	<p>A list of one or more archive volumes or patterns (or keys to disk archive data sets) to be included in the MERGE input without regard to their eligibility for other reasons. Their presence in this list overrides even the EXCLUDE and NOMERGEIN parameters.</p> <p>The volumes specified in this parameter are processed, as well as those selected as a result of the PERCENT parameter. Specify PERCENT=0 to cause only these volumes to be processed. A maximum of 100 entries is supported.</p>
EXCLUDE=	<p>A list of one or more archive volumes or volume patterns (or keys to disk archive data sets) to be excluded from MERGE processing, unless in an INCLUDE list. A maximum of 50 entries is supported.</p>
FCREDIT=	<p>This optional parameter specifies the scan of ARCHVOLS records to those whose creation date is greater than or equal to (“from or since this creation date”) the date specified through this value. The date value must be in an accepted BrightStor CA-Disk for USS format. This parameter and the FDAYSOLD= parameter are mutually exclusive. Do not specify both on the same MERGE command or the command is rejected.</p>
FDAYSOLD=	<p>This optional parameter limits the scan of ARCHVOLS records to those whose creation date is greater than or equal to the current date (or an alternate date specified in the DATE= parameter) minus the number of days specified on this parameter. The maximum value for this parameter is 9999. This parameter and the FCREDIT= parameter are mutually exclusive. Do not specify both on the same MERGE command or the command is rejected.</p>
CREDIT=	<p>This optional parameter limits the scan of ARCHVOLS records to those whose creation date is less than or equal to the date specified through this value. The date value must be in an accepted BrightStor CA-Disk for USS format. This parameter and the DAYSOLD= parameter are mutually exclusive. Do not specify both on the same MERGE command or the command is rejected.</p>

Parameter	Description
DAYSOLD=	This optional parameter limits the scan of ARCHVOLs records to those whose creation date is less than or equal to the current date (or an alternate date specified in the DATE= parm) minus the number of days specified on this parameter. The maximum value for this parameter is 9999. This parameter and the CREDIT= parameter are mutually exclusive. Do not specify both on the same MERGE command or the command is rejected.
LISTALL	Normally report records and messages generated by this function are produced only for the volumes actually processed. If you want to print statistics and messages for volumes not processed, specify this parameter on the MERGE command.
VERIFY	Normally the only update to the UNXINDEX records is the new ARCHVOL key. VERIFY causes the archive media record to be compared to the UNXINDEX record. If the UNXINDEX record has been corrupted, it will be fixed. No updates are made to the archive media record.

Sample MERGE Report

Following is a sample MERGE report:

```

File Edit Transfer Options Connection Macro Window Help
-----
EVS- U- U- DISPLAY 19-JULY 19 08:16:59 1510 116 LINES 1  VOLUME= 02- 153
TOP-HINT TYP T ===  START I ===: CSR
1997.306 OCT 27, 1997  MERGE DATA SET STATUS  PAGE
MEMORY 1.43 PM  GPHS.DISK 0.C.0

-----
DATA SET NAME  ARCHDATE  KEY  FILE  BLOCK  BYTES  COUNT  EXPDATE  ACTION  TAKEN  KEY  FILE  BLOCK  CO
-----
02U01.L1.SK400.LUBR...C01  1997.306 097507  1  1  496  19 1997.306  VERGEJ  200697  7  43
02U02.TASK97A.T100A.C01  1997.306 097507  2  20  507  19 1997.306  VERFEJ  200697  7  20
::EPK021.T02T.D020HFUT  1997.206 097509  1  1  3  1 1997.306  VERGED  200697  7  43
:EPK021.T02T.D020HFUT  1997.206 097508  1  1  3  1 1997.306  VERGED  200697  3  35
:EPK021.T02T.D020HFUT  1997.206 097509  2  2  1  1 1997.306  VERGEJ  200697  8  44
:EPK021.T02T.D020HFUT  1997.206 097508  2  2  1  1 1997.306  VERFEJ  200697  4  40
:EPK021.T02T.D020HFUT  1997.206 097509  3  3  4  1 1997.306  VERGED  200697  9  40
:EPK021.T02T.D020HFUT  1997.206 097508  3  3  4  1 1997.306  VERGED  200697  5  41
:EPK021.T02T.D020HFUT  1997.206 097509  4  4  4  1 1997.306  VERGED  200697  10  46
:EPK021.T02T.D020HFUT  1997.206 097508  4  4  4  1 1997.306  VERFEJ  200697  6  40
1997.306 OCT 27, 1997  MERGE VOL J N C S T A T U S  PAGE
MEMORY 1.43 PM  GPHS.DISK 0.C.0

-----
ARCHIVE VOL.  DATA BLOCK  CRIT=40  NO  ACTION  DATA  UNEXPTED  NO
KEY  UNIT ID IN SET NAME  BYTES  COUNT  BYTES  FILE  FILE  BYTES  FILE
-----
027677  1  :EPK021.PITFEJ.DMS0.D1957350.T12484  1997.306  2  38  000  VERFEJ  2  38  1007
027630  1  :EPK021.C.MGEJ.T.DMCC.D1797290.T134700  1997.311  4  4  10  VERGED  4  4  11
-----

```


Index Maintenance

Index maintenance is used to report on and maintain the file that holds an index of all the files and paths backed up. This file is called the BrightStor CA-Disk for USS Files Data Set. The commands are divided into two groups, maintenance commands and reporting commands.

Maintenance Commands

Following is a list of maintenance commands:

Command	Description
PATHDELETE	Used to delete expired UNIX file records at the PATH level.
VOLDELETE	Used to delete expired UNIX file records at an ARCHVOL level.
PURGE	Used to delete UNIX file records whether they are expired.
REBUILD	Used to recreate UNXINDEX and ARCHVOL records that have either expired prematurely, or were intentionally purged from the Files Data Set.

PATHDELETE

The Files Data Set stores all information about backup data until PATHDELETE runs and releases expired or unwanted backup data from the UNXINDEX and ARCHVOLS subfiles. The entire Files Data Set is scanned for expired data. If all data sets for an archive volume are deleted through processing, the ARCHVOLS record is also deleted and the volume is released from the archives.

Note: The recommended method of operation is to execute PATHDELETE daily during off hours. If this is not possible, execute it after a week, at the very minimum.

Following is a list of parameters that can be used with the PATHDELETE command:

Parameter	Description
SIMULATE	This optional parameter indicates that the execution is to be simulated. Messages are issued as if processing has taken place, but no updates to the archive index are performed.
PATHS=	This optional parameter indicates that processing is to be limited to the path and file names and patterns specified in the supplied list. Patterns are supported and one to ten entries can be supplied.
FILE=	If you want to limit the selection of file names, specify this parameter with up to ten names. Pattern matching is allowed in the name.
VOLUMES=	This optional parameter indicates that processing is to be limited to the ARCHVOLS specified in the supplied list. For archive files on disk (rather than tape), this list consists of the BrightStor CA-Disk for USS generated keys rather than the actual volumes. From 1 to 40 ARCHVOLS, prefixes, or patterns can be supplied. Note: Sysparm IXMCLEANy can increase the number of volumes selected for processing. The purpose of this sysparm is to clean up broken chains between ARCHVOLS and UNXINDEX records. Even if you restrict the number of ARCHVOLS to be selected by specifying VOLUMES= on your command, IXMCLEAN is added to this list any unchained ARCHVOL.
COPIES=	This optional parameter can be defined 1-9999 to indicate that this number of versions of each file is to be maintained in the archives. For example, if COPIES= 3 was specified, all but the three most current unexpired versions of a file are deleted from the archives, regardless of their expiration dates. If multiple copies of a file are archived on the same date, they are considered to be one version of the file.
DATE=YYDDD	This parameter can specify a date in any accepted BrightStor CA-Disk for USS format to be used as the current date whenever date comparisons are made.
LIST=	Normally a message is printed for each file deleted from the archives. If you want to suppress the messages, use this parameter.
LIST=NONE	This parameter suppresses all messages. ALL is the default.

Parameter	Description
SCRATCH=	This keyword parameter tells BrightStor CA-Disk for USS whether to scratch the archive file on disk when all UNXINDEX records that point to it have been deleted and the ARCHVOLS record is being deleted as a result.
SCRATCH=YES	This is the default. SCRATCH=NO keeps the archive file on disk even though there are no more UNXINDEX records and the ARCHVOLS record is being deleted.

Execution

PATHDELETE is accomplished by supplying the above commands to the IXMAINTU procedure. The IXMAINTU procedure is an MVS procedure that runs in the MVS environment.

Example

Delete all expired backed up UNIX files.

```
//job.....
//MAINT EXEC IXMAINTU
//SYSIN DD *
PATHDELETE
/*
```

Condition Code Settings

The following list contains possible condition codes that can be issued while executing the IXMAINTU procedure:

Condition Code	Description
0	Successful completion.
4	Informational warning messages have been issued.
8	Critical errors were encountered; processing was terminated.

VOLDELETE

The ARCHVOL expiration date is established based on the value set for sysparm DYNEXPDTE99365. The recommended value for this sysparm is E99365 (never scratch), its default value. Using this recommendation ensures that an ARCHVOL record exists in the archives, as long as it contains one or more unexpired UNXINDEX records.

The VOLDELETE command is used to delete ARCHVOLS that were created outside of this recommendation. It causes all expired archive volumes (and its contents), either on tape or on disk, to be deleted from the archives. An archive volume (ARCHVOLS record) is considered expired if its expiration date is less than or equal to the current date. Any ARCHVOLS record not expired is not processed.

Following is a list of parameters associated with the VOLDELETE command:

Parameter	Description
SIMULATE	This optional parameter indicates that the execution is to be simulated. Messages are issued as if processing has taken place, but no updates to the archive index are performed.
VOLUMES=	This optional parameter indicates that processing is to be limited to the ARCHVOLS specified in the supplied list. For archive files on disk (rather than tape), this list consists of the generated keys rather than the actual volume serial. From 1 to 40 ARCHVOLS, prefixes, or patterns can be supplied.
EXPFILS	This optional parameter indicates that in addition to expired ARCHVOLS, all expired files are to be deleted, regardless of their presence on an unexpired ARCHVOL. Specifying this parameter is equivalent to running PATHDELETE VOL=/,PATH=/ before running VOLDELETE. That is, all expired UNXINDEX records encountered during processing are deleted regardless of what ARCHVOLS record they point to. In addition, if all UNXINDEX records pointing to a given ARCHVOLS record are deleted, the ARCHVOLS records are also deleted. This can be confusing in that if this occurs, it is possible that an ARCHVOLS record might be deleted that was not specified on the VOLDELETE command of this job.
DATE=	This parameter can specify a date in any accepted BrightStor CA-Disk for USS format to be used as the current date (instead of the real current date) whenever date comparisons are made.

Parameter	Description
LIST=	Normally a message is printed for each file deleted from the archives. If you want to suppress the messages, use this parameter. LIST=NONE suppresses all messages. ALL is the default.
SCRATCH=	This keyword parameter tells BrightStor CA-Disk for USS whether to scratch the archive file on disk when the ARCHVOLS record is being deleted. SCRATCH=YES is the default. SCRATCH=NO keeps the archive file on disk even though the ARCHVOLS record is being deleted.

Execution

VOLDELETE is accomplished by supplying the above commands to the IXMAINTU procedure. The IXMAINTU procedure is an MVS procedure that is run in the MVS environment.

Example

Delete all expired ARCHVOLS and at the same time delete expired UNXINDEX records.

```
//job.....
//MAINT EXEC IXMAINTU
//SYSIN DD *
(Code1)VOLDELETE EXPFILS
/*
```

Condition Code Settings

The possible condition codes that can be issued while executing the VOLDELETE command are listed in the table in Condition Code Settings under PATHDELETE in this chapter.

PURGE

Use the PURGE command to delete entries from the archive index regardless of their expiration dates. It applies to both archived files (UNXINDEX records) and archive volumes (ARCHVOLS records). Like the PATHDELETE command, when all files are deleted that exist on an archive volume, the archive volume is released from the archives.

The PURGE command performs an uncatalog action for the tape (per sysparm UNCATARC), uncatalogs archived files (per sysparm UNCATPSU), and interfaces to tape management systems to release the tape (per sysparm TMSCTLEX).

Following is a list of parameters that are associated with the PURGE command:

Parameter	Description
SIMULATE	This optional parameter causes messages to be issued as if processing has taken place, but no updates to the archive index are performed.
VOLUMES=	This parameter is required if the PATH= parameter is not specified; otherwise it is optional. It indicates that processing is to be limited to the ARCHVOLS specified in the supplied list. The list should consist of primary volumes only, not copy volumes. Any copy volume that is listed is ignored. Copy volumes and multi-volume chains are automatically removed when their associated primary volumes are purged. For backed up files on disk (rather than tape), this list consists of the generated keys rather than the actual volumes. From 1 to 40 ARCHVOLS, prefixes, or patterns can be supplied.
PATH=	This optional parameter (required only if the VOLUMES parameter is not specified) indicates that processing is to be limited to the names or patterns specified in the supplied list. From 1 to 10 path/file names can be entered. The names refer to UNXINDEX entries to be purged, not to ARCHVOLS entries. Note: Caution should be used when specifying patterns for this parameter. For example, PATH= is valid but should never be used because it deletes everything.
LIST=	Normally a message is printed for each file deleted from the archives. If you want to suppress the messages, use this parameter. LIST=NONE suppresses all messages. ALL is the default.
FILE=	If you want to limit the selection of files, specify this parameter with up to 10 names. Pattern matching is allowed in the name.
SCRATCH=	This keyword parameter tells BrightStor CA-Disk for USS whether to scratch the archive file on disk when the ARCHVOLS record is being deleted. SCRATCH=NO is the default. SCRATCH=YES scratches the archive file from disk after the ARCHVOLS record is deleted.

Execution

PURGE is accomplished by supplying the above commands to the IXMAINTU procedure. The IXMAINTU procedure is an MVS procedure that is run in the MVS environment.

Example

Delete all UNXINDEX records and the ARCHVOLS record for ARCHKEY 123456.

```
//job.....  
//MAINT EXEC IXMAINTU  
//SYSIN DD *  
PURGE VOL=123456  
/*
```

Condition Code Settings

For a list of possible condition codes that can be issued while executing the PURGE command, see Condition Code Settings under PATHDELETE in this chapter.

REBUILD

A utility is available that can be used to recreate index records from the archive tapes (or archive data sets on disk) that contain the backed up files.

Use of this facility would be appropriate in any of the following situations:

- Restoring files at a site other than that at which the files were backed up.
- The archive index has been destroyed and no suitable backup is available.
- The index entries for a given volume have either expired or have been purged, but the tape has not been overwritten yet, and one or more data sets must be restored.

Special Considerations

The rebuild process inserts UNXINDEX records into the Files Data Set in sequence based on the file name, backup date, and backup time of the files to which they correspond.

Note: A rebuild is not allowed if a volume already has existing entries in either the ARCHVOLS or UNXINDEX files. In such a case, the volume must first be purged and the rebuild resubmitted. To do this, use the PURGE command with the VOLUMES= parameter. For a description of this, see PURGE in this chapter.

If it is necessary to retrieve one or more files from a tape, rather than performing a PURGE on the tape, consider doing the following:

1. Allocate a small, temporary FILES data set
2. Perform a REBUILD on the tape pointing to the temporary FILES data set allocated in step one
3. RESTORE the files needed, again pointing to the temporary FILES data set
4. When steps 1-3 are complete, delete the temporary FILES data set

To avoid having UNXINDEX records point to copy archives, do not rebuild from archives that are considered copy volumes. Rebuild the primary archive, and then copy with the COPY utility. For a description of this utility, see Making Additional Copies of Archive Tapes in this chapter.

JCL - Rebuild Index

The ARCHVOLS and UNXINDEX records for files on an archive tape, or in an archive data set on disk, can be rebuilt using the following JCL:

```
//REBUILD EXEC REBUILDU,  
//          DSN=ddd,  
//          VOL=vvv,  
//          UNIT=uuu  
//SYSIN DD *  
REBUILD (parameters go here)
```

Note: In the above JCL, *ddd* refers to the “Data Set Name” of the ARCHVOL, *vvv* refers to the “True Vol” where *ddd* resides, and *uuu* refers to the “Device Type Unit” of *vvv*. These values are stored in the Files Data Set, and are displayed in Sample LISTV Report in this chapter.

REBUILD Command and Parameters

A REBUILD command is not needed unless you want to specify an expiration date to the ARCHVOLS index entry that is different from the default computation. See the EXPDT= parameter below. All other needed information is obtained from the DD statement for the input archive tape, or the input archive data set on disk. Symbolic parameters are provided in the JCL procedure to allow you to identify the input data set. If the archive data set spans multiple volumes, the volume serial numbers must be properly specified in the JCL.

The syntax of the REBUILD command is as follows:

```
REBUILD EXPDT=,UNIT=
```

Following is a description of the parameters associated with the REBUILD command:

Parameter	Description
EXPDT=	<p>The default expiration date for the archive volume is the highest date found on all of the individual archived data sets. This ensures that the volume is not dropped due to either a DELETE command or a MERGE PROCESS until every data set has indeed expired. This default calculation can, however, be overridden by supplying a date in any accepted BrightStor CA-Disk for USS. This can be desirable in cases where archive retention periods are set by an installation, and not by the data set expiration date or unit.</p> <p>Note: The create TIME for the ARCHVOLS record is x'0000' after REBUILD. The exact time is not available.</p>
UNIT=	<p>The UCB device type determines the default unit type. You can override the default for DYN unit types by specifying the same esoteric name as used in the JCL. This ensures the proper ARCFLAG2 setting in the ARCHVOL record.</p>

Reporting Commands

Following is a list of reporting commands:

Command	Description
LISTU	A report that lists the attributes stored with the index record for each file backed up.
LISTV	A report that lists the attributes stored with the ARCHVOL records. These are the records that track the media (disk or tape) that hold the backed up files.

LISTU

This report is a user-defined report. The user defines which fields are to go on the report and the report is generated accordingly.

Following is a list of the parameters associated with the LISTU command:

Parameter	Description
PATH=,	Specify the explicit path name to be displayed (no pattern characters are allowed). This restricts the LISTU report to just the directories specified. If you want to further restrict the list to certain file names, then specify those file names in the FILE= parameter, described in FILE= in this table.
VERSION=,	Use this parameter to indicate what version of the UNIX file record is to be displayed. Values of 0 (the default) to -99 can be specified.
FILE=,	This optional parameter displays the UNXINDEX records specified. Specify a list of one or more file names or patterns to restrict the display to those entries with matching names. A maximum of 20 entries is supported. Specifying this parameter without the PATH= parameter causes all files to be listed.
FIELDS=,	This required parameter contains the list of UNXINDEX record field names you want to appear on the LISTU report. Specify a list of field names you want in the column sequence.
NOWILD	This optional parameter is used to find paths that have sub-directories that include wild characters (?,!,*,\). The complete path must be specified in the PATH= parameter. A maximum of 50 fully qualified field names is supported, and their combined output length must be less than or equal to 132 characters.

Following is a list of values for the FIELDS= parameter of the LISTU command:

UNXINDEX Field	FIELDS= Value	Field Description
Calculated	RVER	This is the relative version number, ranging in values from 0 to 9999. Use the DUPLICATES command parameter for all versions.
UNXNAME	NAME44	44 characters of the UNIX file name.
	NAME28	28 characters of the UNIX file name.
UNXATIME	ATIME	Time at which the file was backed up.

UNIXINDEX Field	FIELDS= Value	Field Description
UNXADATE	ARCDT	Date on which the file was backed up.
UNXEXPDT	EXPDT	Expiration date. The date on which the UNIX file expires out of the Files Data Set.
UNXARKEY	ARKEY	Key of ARCHVOL record.
UNIXRSDT	RTDTE	This is the date the file was last restored.
UNIXRSTM	RTIME	This is the time the file was last restored. A 24-hour time clock is used.
UNIXUONR	UONR	This is the user owner of the file.
UNIXGONR	GONR	This is the group owner of the file.
UNIXFLSZ	FLSZ	This is the size of the file in bytes.
UNIXBLSZ	BLSZ	This is the size of the blocks of the file in bytes.
UNIXBLAL	BLAL	This is the number of blocks the file is using.
UNIXFTYP	FTYPE	This is the type of file.
UNIXLSAC	LSAC	This is the date and time the file was last accessed.
UNIXMODT	MODT	This is the date and time the file was last modified.
UNIXSTCH	STCH	This is the date and time when the file status was changed.
UNIXCRDT	CRDT	This is the creation date and time of the file.
UNIXRFDT	RFDT	This is the date and time the file was last referenced.

Execution

LISTU is accomplished by supplying the above commands to the LISTU procedure. The LISTU procedure is an MVS procedure that is run in the MVS environment.

Example

List the latest version of all the records for all the backed up UNIX files and display the first 28 characters of their name, the file type, the user owner, the group owner, the date the file was last referenced, the date the file was last modified, and the size of the file.

```
//job.....
//LIST EXEC LISTU
//SYSIN DD *
LISTU FIELDS=(NAME28, FTYPE, UONR, GONR, RFDT, MODT, FLSZ), PATH=\\
/*
```

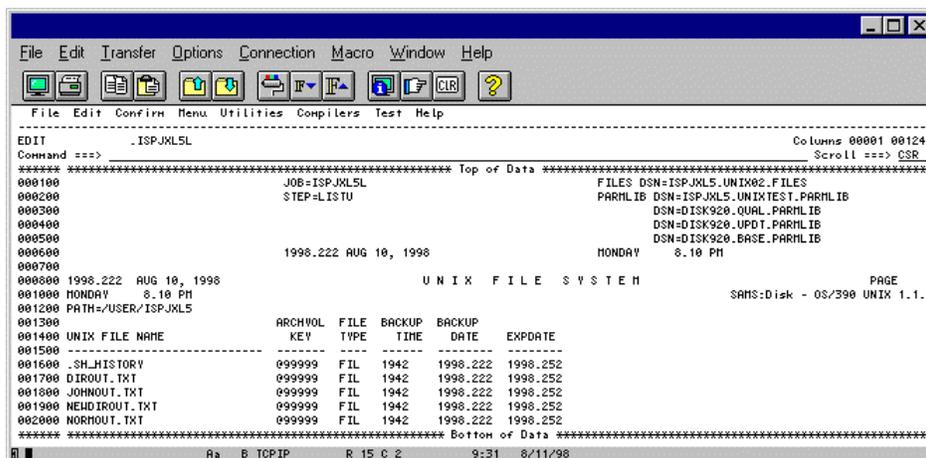
LISTU Condition Codes

BrightStor CA-Disk for USS index entries for archive volumes and archived files can be listed selectively or in aggregate. In either case, the following list contains the possible condition codes that can occur:

Condition Code	Description
0	All specified entries were listed correctly.
4	Nothing was selected for listing.
8	One or more unique entries specified were not found.
8+	Errors occurred during the list process.

Sample Report

The following is a sample report generated from the LISTU command:



LISTV Command

Following is a list of parameters associated with the LISTV command:

Parameter	Description
VOLUME=	By default, all ARCHVOL records are listed. Specify a list of one or more keys (or key prefixes) of the ARCHVOL records you wish to see. For tape volumes, the key is simply the tape volser. For disk ARCHVOLS, the key is generated by BrightStor CA-Disk for USS. The format of a key used for disk archiving is presented under the sysparm description for ARCVOLID@#\$\$+-%&[in Appendix E "Sysparms."
DSNAME=	This optional parameter restricts the list of volumes displayed to only those with file names matching the list of names or pattern names specified. For example, DSN=DMS.BACKUPS/ would list only tapes starting with these names. A tape volume with DSN=DMS.ARCHIVES would be skipped. Note: A maximum of 20 entries is supported.
DATE=	This optional parameter restricts the list of volumes displayed to only those created on the specified date. The date value can be entered in any accepted BrightStor CA-Disk for USS format.

LISTV Flags

The flag fields of the archive volume index records use individual bits. These bits have the following meanings:

Byte #	Binary	Hex	Meaning
First	1000 0000	80	volume is disabled
	0100 0000	40	volume is a copy of another volume
	0010 0000	20	volume is password-protected
	0001 0000	10	volume CLOSE failed
	0000 1000	08	volume not closed
	0000 0100	04	volume is full (next file did not fit)
	0000 0001	01	volume written at 6250 BPI
Second	1000 0000	80	DYN1 special allocation needed
	0100 0000	40	DYN2 special allocation needed
	0010 0000	20	DYN3 special allocation needed

Byte #	Binary	Hex	Meaning
	0000 1000	08	IDRC data compaction
	0000 0010	02	ARC#BYTE 6 byte count is present
	0000 0001	01	ARCDDEVT pseudo device type present

Execution

LISTV is accomplished by supplying the above commands to the LISTU procedure. The LISTU procedure is an MVS procedure that is run in the MVS environment.

Example

List all the ARCHVOLS records in the Files Data Set:

```
//job.....
//LIST EXEC LISTU
//SYSIN DD *
LISTV
/*
```

Condition Code Settings

The possible condition codes that can be issued while executing the LISTV command are listed in LISTU Condition Codes in this chapter.

Sample LISTV Report

The following is a sample report generated by the LISTV command:

ARCHIU KEY	DATA SET NAME	WRITTEN TIME	DATE	EXPDATE	SETS	BLOCK COUNT	K/M/G/T BYTES	RC FEET	CNT FLAG	COPY	TRUE SEQ	DEVICE	KEY CHAIN
204000	QDMSTEST.SAMS.DISK0136.LOADLIB.C0018000	1804	1997.293	1997.325	1	289	9.00M 17	20	1 000B		204000	1	3480CART
204027	ISPUVK1.ARCPRIM.DMSC.C1997259.T092755	0928	1997.259	1999.365	1	16030	500.79M	1325	3 0013		204027	1	3480CART 203804
204117	ISPUVK1.ARCPRIM.DMSC.C1997259.T103411	1034	1997.259	2000.001	1	8169	255.21M	675	2 0013		204117	2	3480CART
204149	ISPUVK1.ARCPRIM.DMSC.C1997259.T101535	1016	1997.259	1999.365	1	8125	253.04M	672	2 0013		204149	1	3480CART 203940
204171	ISPUVK1.ARCPRIM.DMSC.C1997295.T143237	1432	1997.295	1999.365	1	29	920.97K	3	1 0003		204171	1	3480CART
204254	ISPUVK1.ARCPRIM.DMSC.C1997295.T153406	1536	1997.295	1999.365	1	29	920.98K	3	1 0003		204254	1	3480CART
204382	QDMSTEST.QAR900.ARC0.DMSC.C1997252.T182449	1825	1997.252	1999.365			0		1 0009		204382	1	3480CART
204496	QDMSTEST.QAR900.ARC1.DMSC.C1997251.T192109	1921	1997.251	1999.365	20	177	5.00M	14	1 4207		204496	1	3480CART
204590	QDMSTEST.QAR900.MRP1.DMSC.C1997261.T203054	2031	1997.261	1999.365	70	126	2.16M	6	1 0203 200546		204590	1	3480CART
204596	QDMSTEST.SAMS.DISK0136	1805	1997.286	1997.318	1	289	9.00M 17	20	1 000B		204596	1	3480CART

Making Additional Copies of Archive Tapes

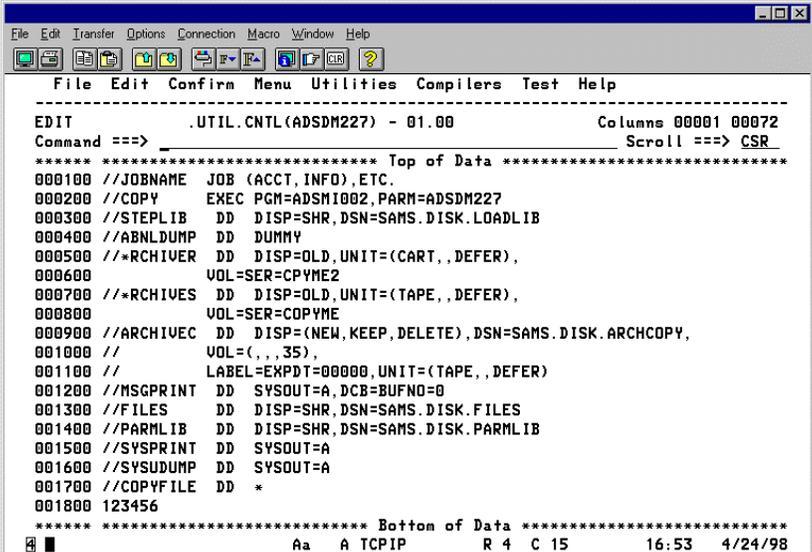
If you produce an archive tape without a duplicate copy, it is possible to create a copy after the fact. There are also other reasons why you might want to copy an archive tape. If you had a primary and duplicate copy, but the primary copy was lost, damaged, or otherwise not available, you might want to copy the duplicate copy to regain the protection in having two tapes.

BrightStor CA-Disk for USS must keep track of multiple copies of archive tapes, so when it makes a copy of an existing archive tape, the volume serial of the new copy is recorded in the index of the volume record for which the copy is made.

The volumes to be copied must already have entries in the BrightStor CA-Disk for USS archive tape record index (ARCHVOLS). An input stream contains records indicating the keys to the archive tapes that are to be copied. They are read from the COPYFILE DD statement. Beginning in columns 1 through 16, enter the 6-byte volume serial of the tape volume to be copied. If you are copying an archive file residing on disk rather than tape, enter the 6-byte key to the disk archvols record. Only one key can be supplied per statement. Multiple statements can be entered to copy multiple volumes.

Note: Only the first volume of a multivolume archive file should be specified in the input stream. The rest of the volumes in the sequence are automatically copied when the first volume is copied. If any one of the volumes of a multivolume archive file needs to be copied, all have to be copied so that proper volume chaining is maintained on the volume labels.

The following JCL is presented as an example for the utility. It causes a copy volume to be created for archive tape VOLSER 123456. This utility only works for tape.



```

EDIT          .UTIL.CNTL(ADSDM227) - 01.00          Columns 00001 00072
Command ==>                                     Scroll ==> CSR
***** ***** Top of Data *****
000100 //JOBNAME  JOB (ACCT,INFO),ETC.
000200 //COPY    EXEC PGM=ADSM1002,PARM=ADSDM227
000300 //STEPLIB DD DISP=SHR,DSN=SAMS.DISK.LOADLIB
000400 //ABNLDUMP DD DUMMY
000500 //RCHIVER  DD DISP=OLD,UNIT=(CART,,DEFER),
000600           VOL=SER=CPYME2
000700 //RCHIVES DD DISP=OLD,UNIT=(TAPE,,DEFER),
000800           VOL=SER=CPYME
000900 //ARCHIVEC DD DISP=(NEW,KEEP,DELETE),DSN=SAMS.DISK.ARCHCOPY,
001000 //           VOL=(, , 35),
001100 //           LABEL=EXPDT=00000,UNIT=(TAPE,,DEFER)
001200 //MSGPRINT DD SYSOUT=A,DCB=BUFNO=0
001300 //FILES   DD DISP=SHR,DSN=SAMS.DISK.FILES
001400 //PARMLIB  DD DISP=SHR,DSN=SAMS.DISK.PARMLIB
001500 //SYSPRINT DD SYSOUT=A
001600 //SYSUDUMP DD SYSOUT=A
001700 //COPYFILE DD *
001800 123456
***** ***** Bottom of Data *****
Aa  A TCP/IP      R 4  C 15      16:53  4/24/98

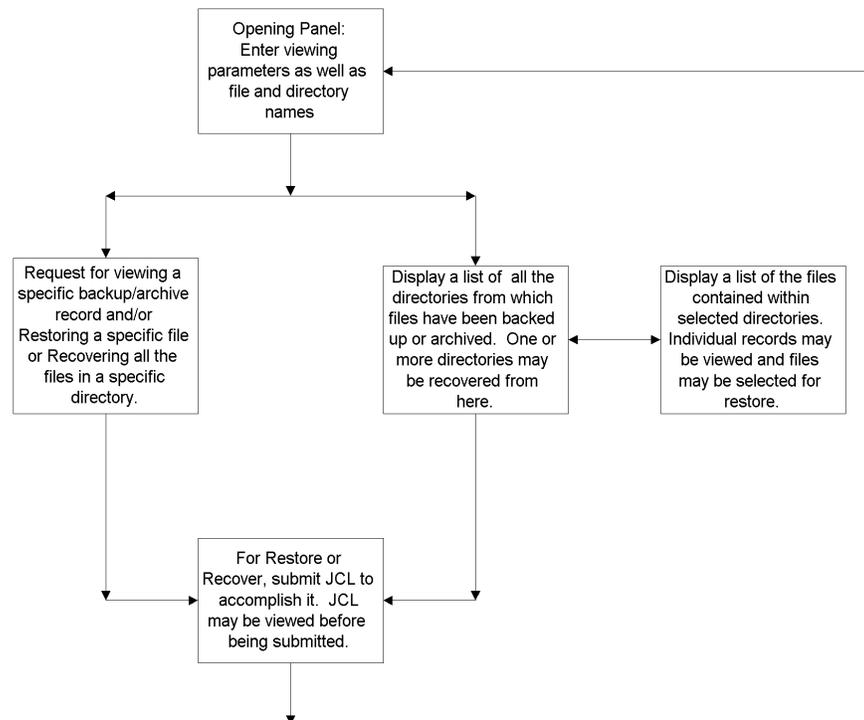
```


Using ISPF

An ISPF interface facilitates the management of UNIX files that exist in your file system or have been backed up, archived, or both, by BrightStor CA-Disk for USS. It is also a convenient way to view your files on the file system or in the Files Data Set, as well as to view the mounted file system table.

The Files Data Set stores the information about all files backed up and archived by BrightStor CA-Disk for USS. In it is the information about the files backed up/archived, the paths (or directories) they were contained in, and the storage media upon which the backup or archive was placed. BrightStor CA-Disk for USS queries the file system or the Files Data Set, depending on the options selected, for all of the information displayed on the ISPF selection lists.

The general flow of the ISPF panels is as follows:



The interface seeks to minimize having to drill down too many levels to accomplish any function; so most options are specified on the main panel. Also, path hierarchies are displayed in a flat view so as to more easily see their structure flow. If Generate DSCL is specified along with Fast JCL Gen, JCL is automatically generated using the inputs on the main screen without presenting a selection list. JCL can be viewed before it is submitted or it can be submitted automatically. When the JCL is viewed before being submitted, it can be optionally saved into a PDS or sequential file for reuse later.

General Considerations

Following are some general considerations concerning ISPF:

- When a user runs the application for the first time, the configuration screen is entered automatically. The user is required to specify the Files Data Set and parmlib data set names to use for processing. This information must be present before the interface can be used. Job card information and some default option values can also be specified at this time.
- The main panel allows the specification of a directory or file (using the complete path name), or a file system name. For List processing, wildcards can be used, however, the Show Attributes option requires an explicit directory, file, or file system name. In addition, the file repository option determines whether the 'live' file system or the Files Data Set is scanned for processing. If the Generate DSCL option is used, this information is used to create the SCAN or FIND command, if applicable. Either a specific path or a path hierarchy can be entered, or a specific HFS DSN or data set pattern can be entered.
- The view options allow the user to view the file system either as a set of directories, or as a set of file system names. Using the LIST line command, the user can drill down into the hierarchy as necessary.
- Selection criteria for particular file types can be entered on the main panel as well. The file types can be regular files, symbolic links, or device (character special) files. Directories are always included.
- If the user has Superuser authority, this can be specified on the main panel, causing the application to switch to Superuser mode for the list display, and causing the proper DSCL command to be generated for the JCL. If the user does not have Superuser authority, this switch has no effect.
- Choosing 'simulate' mode on the main panel has no effect on ISPF processing, but it generates the proper SIMULATE command on the generate JCL.

General Panel Overview

This section provides a general overview of the BrightStor CA-Disk for USS panels.

Main Panel

Upon invocation of the REXX procedure UNXSTART, the panel shown below appears. From here, all other actions take place.

```

Config Utilities Help
-----
                          CA Disk USS File Utility
Option ==> _____

blank Display List          A Show Attribs

Enter one of the following:
Path/File Name . . . _____
File System Name . . . _____

Additional Options:
Repository . . . 1 1. File System      Initial View . . . 1 1. Directory
                  2. Archives                2. File System

Processing Options          File Selection
/ Run as Superuser         / Regular Files
/ Generate DSCL            / Symbolic Links
                           / Special Files

DSCL Options
_ Fast Generate
_ Simulate Mode

| Copyright Computer Associates, Inc.(Created 1999, Unpublished) All Rights |
| Reserved |

```

- Option—Specify an option here to either list the objects matching the path or file system name, or display the attributes of an explicit directory, file, or file system.
 - Blank—List entries from the selected repository that match the path/file or file system name or pattern entered.
 - A—Show the attributes of the explicit directory, file, or file system specified.

- Path/File or File System Name—Specify a path (directory or file) or file system name. Only one or the other can be specified. If an explicit path or file name is specified, only that entity is processed either for the list or attribute display. If you want to display the entire hierarchy, use the backslash (\) at the end of the path you want to start with. If more than 44 characters are in the pathname, place a '+' as the first character and then press Enter. A 255-byte entry field is displayed. File system names may include wildcards, including the backslash (matches all subsequent node levels), asterisk (matches all subsequent characters), question mark (matches any individual character), or exclamation point (matches all previous characters). Wildcards are not valid for the Show Attributes option. When generating DSCL, if an explicit file system name (one surrounded by single quotes) is entered, the resulting command is a FIND HFS; otherwise, a SCAN HFS is generated. Paths/files entered on the main screen always result in a FIND PATH command.
- Repository—This option determines whether directories and files are read from the live file system or from the Files Data Set (archives). If the Generate DSCL option is chosen, this also determines which DSCL commands are presented to the user.
- Initial View—This option determines the first list to be displayed. The Directory view (option 1) generates a list of pathnames from the path or file system entered in the previous field. The File System view (option 2) generates a list of mounted file systems from the path or file system name entered.
- Processing Options—These options allow the user to decide how processing will occur.
 - Run As Superuser—Causes the application to attempt to use the Superuser option. If the user is authorized, Superuser mode is enabled and all files/paths are viewable in the list. If this option is not set, or the user does not have the authority, only files that the user has access to are listed. In addition, selecting this option causes the appropriate Superuser command to be generated for DSCL.
 - Generate DSCL—Presents several additional screens and generates JCL for the user to submit before batch processing. If the file system repository was selected, the user is allowed to choose from the Backup, Archive, Move, or Copy options. If the archives repository was selected, Restore and Recover options are available. The resulting list display also contains information in the 'status' field to indicate whether a directory or file is implicitly or explicitly selected/excluded by the DSCL commands. In addition, several additional commands are available, including SELECT, EXCLUDE, INCLUDE, PRECLUDE, and UNDO (for a description of these line commands, see below). Finally, after the selection list has been exited, a screen is presented allowing the user to specify command-specific overrides and parameters.

- File Selection—These options determine which types of files are displayed in the file list, and may build CRITERIA parameters for DSCL generation. If all files are selected (the default), no CRITERIA parameter is generated.

Setting Configuration Options

The Files Data Set and the parmlib names are specified in the configuration panel. Job card information for JCL generation is specified along with whether to view the JCL before being submitted:

```

Config Utilities Help
-----
| 1 1. Settings | CA Disk USS File Utility
|-----|

blank Display List          A Show Attrs

Enter one of the following:
Path/File Name . . .
File System Name . .

Additional Options:
Repository . . 1 1. File System      Initial View . . 1 1. Directory
                2. Archives          2. File System

Processing Options          File Selection
/ Run as Superuser         / Regular Files
/ Generate DSCL            / Symbolic Links
                            / Special Files

DSCL Options
Fast Generate
Simulate Mode

```

```
CA Disk USS Configuration
Command ==> _____

Set startup default processing parameters:

Enter "/" to select option
/ Edit JCL before submit
/ Run as Superuser
/ Generate JCL

Files DSN . . . 'KENDE02.UNIX.FILES'
Parmlib DSN . . 'KENDE02.UNIX.PARMLIB'

Enter/Modify Job Statements for DSCL job generation:
==> //KENDE02X JOB (01026),'DEAN KENT',CLASS=B,
==> //          MSGCLASS=X,NOTIFY=KENDE02
==> //*
==> //*
```

File System/Path List Panel

The file system/path list panel shows all or a subset of the file systems or paths that match the entry supplied in the main panel. File systems are listed by their full 44-character names, while paths are displayed as a string of up to 44 characters, with ellipses inserted for pathnames longer than this.

From the file system list, you can list the paths within each file system. If there is a file system mounted below the one selected, the path list does **not** display the paths from that data set. The archive repository does not currently support either the File System view or the specification of file system names.

The following displays are the result of entering OMVS* in the file system name field for the file system repository and an initial view setting of 'file system.' Then the OMVSUSR,KENDE02 file system is listed, resulting in a display of all directories in that file system.

```

----- CA-Disk 05/390 UNIX ----- Row 1 from 3
Listing for: OMVS*
COMMAND ==>

Command   Status   Path/DSN
-----
LIST      OMVSGRP.SB60.USERS
          OMVSSYS.WEBSRPHR.V4R0M0.SBBOHFS
          OMVSUSR.KENDE02
***** Bottom of data *****

```

```

----- CA-Disk 05/390 UNIX ----- Row 1 from 128
Listing for: OMVSUSR.KENDE02
COMMAND ==>

Command   Status   Path/DSN
-----
/a/kende02
/a/kende02/aoewebs
/a/kende02/attrtest
/a/kende02/attrtest/aoewebs
/a/kende02/attrtest/ftib
/a/kende02/attrtest/fti8
/a/kende02/attrtest/ispjxl
/a/kende02/attrtest/ispmwb1
/a/kende02/attrtest/ostar01
/a/kende02/attrtest/sssaoe
/a/kende02/attrtest/sssaoe/jni
/a/kende02/attrtest/sssaoe/xmi
/a/kende02/attrtest/sssaoe/Webserver
/a/kende02/attrtest/sssaho
/a/kende02/attrtest/test
/a/kende02/attrtest/test/aoewebs
/a/kende02/attrtest/test/ftib
/a/kende02/attrtest/test/fti8
/a/kende02/attrtest/test/ispjxl
/a/kende02/attrtest/test/ispmwb1
/a/kende02/attrtest/test/ostar01

```

- **Command Line** – System wide ISPF commands are entered here. For BrightStor CA-Disk for USS, RESET, CANCEL, and END are supported, depending on whether Generate DSCL was selected or not.
 - RESET (DSCL Generate mode only) – Removes all select/excludes stored previously, and resets the STATUS field back to the default.
 - CANCEL – Ends processing and returns back to the main menu.
 - END – Stores any SELECT/EXCLUDE commands and returns to the previous menu. When all list panels have been exited, if the Generate DSCL option was selected, the user is presented with the parameter screen for the DSCL command selected.
- **Line Commands** – This is where the various line action commands are entered. Multiple line commands are supported. Some line commands are available only when Generate DSCL is selected. In all processing modes, the file system list display only allows the LIST command, since DSCL does not recognize a SELECT or EXCLUDE for HFS or FSYS command. The line action commands available in all modes are:
 - L or LIST – Lists the contents of the file system or the path selected. If the path or file system has been selected or excluded, the status field of all entries in the resulting list reflects this with either a '+' or '', respectively.
 - A or ATTRIB – Displays the attributes of the file system or the path selected. Paths and files have two displays, one for the UNIX attributes and one for the 'repository' information. The repository may either be a UNIX file system or the Files Data Set.

The line action commands available only for DSCL Generation are:

- S or SELECT – Selects the path for processing and causes a SELECT command to be generated in the DSCL. The status field is updated with 'SELECT' for the path chosen. The status field contains a '+' character for all subdirectories, indicating that they will be implicitly selected during DSCL processing.
- I or INCLUDE – Selects only the specified path for processing, and causes a SELECT command to be generated in the DSCL. The status field is updated with 'INCLUDE' for the path chosen. No subdirectories are implicitly included.
- E or EXCLUDE – Excludes the path from processing and causes an EXCLUDE command to be generated in the DSCL. The status field is updated with 'EXCLUDE' for the path chosen. The status field contains a '' character for all subdirectories, indicating that they are implicitly excluded during DSCL processing. In addition, if any subdirectories have explicit SELECT or INCLUDE commands, the status field is updated for those directories to say 'Not Sel' or 'Not Incl' to indicate that the EXCLUDE will override those during DSCL processing.

- P or PRECLUDE – Excludes only the specified path from processing, and causes an EXCLUDE command to be generated in the DSCL. The status field is updated with 'PRECLUDE' for the path chosen. No subdirectories are implicitly excluded.
- U or UNDO – Remove the SELECT or EXCLUDE from the path selected.
- Status Field – When Generate DSCL is selected, the select/exclude status of each File System/Path is displayed here. The possible values this field might contain are:
 - Blank – The data set or path is either not selected or has been implicitly excluded.
 - + – The data set or path has been implicitly selected either due to a higher level SELECT, or due to a FIND command being generated from the main panel input.
 - SELECT – The path hierarchy has been explicitly selected. A SELECT statement is generated.
 - INCLUDE – The path has been explicitly selected. A SELECT statement is generated.
 - EXCLUDE – The path hierarchy has been explicitly excluded. An EXCLUDE statement is generated.
 - PRECLUDE – The path has been explicitly excluded. An EXCLUDE statement is generated.
 - Not Sel – The line has been explicitly selected, and a SELECT statement is generated, but a higher level EXCLUDE prevents this path from being processed.
 - Not Incl – The line has been explicitly included, and a SELECT statement is generated, but a higher level EXCLUDE prevents this path from being processed.

File List Panel

A list of all the files contained within a specific directory, or all files backed up from a particular directory, is displayed when the LIST line command is entered for a specific path. From this list, you can select or exclude each file for processing in DSCL, or display the attributes.

```

CA-Disk 05/390 UNIX Row 4 from 157
Listing for: /a/kende02/attrtest
COMMAND ==>

Command  Status  Type   File Name                Date      Time
-----
          Fil  .sh_history              2000.294 1017
          Fil  new.txt                  2001.172 2052
          Fil  test.txt                 2001.172 2041
          Fil  ListPds.java             2001.086 1202
***** Bottom of data *****

```

- **Command Line**—System wide ISPF commands are entered here. For BrightStor CA-Disk for USS, RESET, CANCEL, and END are supported, depending on whether Generate DSCL is selected.
 - RESET (DSCL Generation only)—Removes all select/excludes stored previously, and resets the STATUS field back to the default.
- **Line Commands**—This is where various BrightStor CA-Disk for USS line action commands are entered. Multiple line commands are supported. Some line commands are available only when 'Generate DSCL' is selected. The BrightStor CA-Disk for USS line action commands available in all modes are:
 - A or ATTRIB—Displays the attributes of the file selected. Files have two displays, one for the UNIX attributes and one for the 'repository' information. The repository may either be a UNIX file system or the Files Data Set.
 - S or SELECT—Selects the file for processing and causes a SELECT command to be generated in DSCL. The status field is updated with SELECT for the file chosen.
 - I or INCLUDE—Same as SELECT, except that the status field displays INCLUDE.
 - E or EXCLUDE—Excludes the file from processing and causes an EXCLUDE command to be generated in DSCL. The status field is updated with EXCLUDE for the file chosen.

-
- P or PRECLUDE – Same as EXCLUDE, except the status field displays PRECLUDE.
 - U or UNDO – Remove the SELECT or EXCLUDE from the path selected.
 - Status Field – The select/exclude status of each file is displayed here. The possible values this field might contain are:
 - Blank – The file is either not selected or has been implicitly excluded.
 - + – The file has been implicitly selected either due to a higher level SELECT, or due to a FIND command being generated from the main panel input.
 - SELECT – The file has been explicitly selected. A SELECT statement is generated.
 - INCLUDE – The file has been explicitly selected. A SELECT statement is generated.
 - EXCLUDE – The file has been explicitly excluded. An EXCLUDE statement is generated.
 - PRECLUDE – The file has been explicitly excluded. An EXCLUDE statement is generated.
 - Not Sel – The line has been explicitly selected, and a SELECT statement is generated, but a higher level EXCLUDE prevents this file from being processed.
 - Not Incl – The line has been explicitly included, and a SELECT statement is generated, but a higher level EXCLUDE prevents this file from being processed.
 - Each line of the file list contains 4 informational fields:
 - File Type – The file type, as follows:
 - ◆ FIL – Regular File
 - ◆ SYM – Symbolic Link
 - ◆ DEV – (Device) Character Special file
 - File Name – The name of the file.
 - Date – For the Archive repository, this is the date the file was backed up. For all other options, this is the last modification date (date that data was changed).
 - Time – For the Archive repository, this is the time that the file was backed up. For all other options, this is the last modification time.

Command Parameter Panels

Most commands have parameters that can be specified to modify processing. These panels provide the ability to optionally specify the values for these parameters.

The Backup and Archive parameter panels are essentially the same, except that the Archive parameter panel allows the file disposition to be specified. Other fields that can be specified and included on the generated DSCL are EXPDT and RETPD.

```
CA-Disk OS/390 Unix Path/File Backup Utility
Command ==> _____

Retention Information
Expdt: . . . . . _____
Retpd: . . . . . _____

Additional Options
Disposition . . . 1 1. Keep

Enter the END command to continue, or CANCEL to exit.
```

```
CA-Disk 05/390 Unix Path/File Archive Utility
Command ==> _____

Retention Information
Expdt: . . . . . _____
Retpd: . . . . . _____

Additional Options
Disposition . . . 1 1. Keep
                2. Delete

Enter the END command to continue, or CANCEL to exit.
```

The parameter panels for both Move and Copy are identical. Either TOHFS or TOPATH must be specified, with all other parameters optional. The FILE OVERRIDES switch displays another panel after this one if it is selected:

```

CA-Disk 05/390 Unix Path/File Copy Utility
Command ==> _____

Enter the destination path or HFS file:
  To Path: . . . . . _____
  To HFS : . . . . . _____

Additional Options
  Overwrite . . . . . 1 1. Yes          Enter "/" to select option
                               2. No           _ Create Directories
                               3. Newer        _ Delete Existing Files
                                         _ Delete Existing Paths
  Overrides
  _ File Overrides                _ Process Paths Only
                                   _ Do Not Copy Data
                                   _ Verify Files

Enter the END command to continue, or CANCEL to exit.

```

The Path/File Overrides panel allows the specification of a new pathname, owner, and permission information for the MOVE, COPY, RESTORE, and RECOVER commands, and causes the appropriate parameters to be generated in the resulting DSCL. This panel is displayed for Restore and Recover by default, but is only displayed for Move and Copy if the File Overrides option is selected:

```
CA-Disk 05/390 Unix Path/File Overrides

New Path Name
_____
_____
_____
_____

New Owner . . . . . _____ New Group . . . . . _____

New Permissions . . . . . _____

Enter the END command to continue, or CANCEL to exit.
```

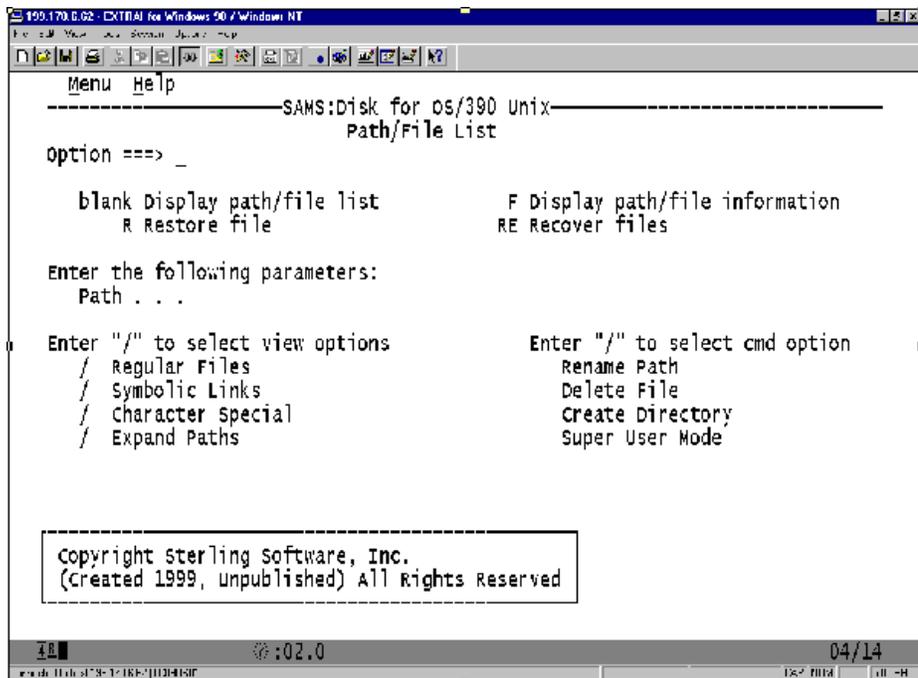
Using the Panels

The next section describes the panels and how to use them.

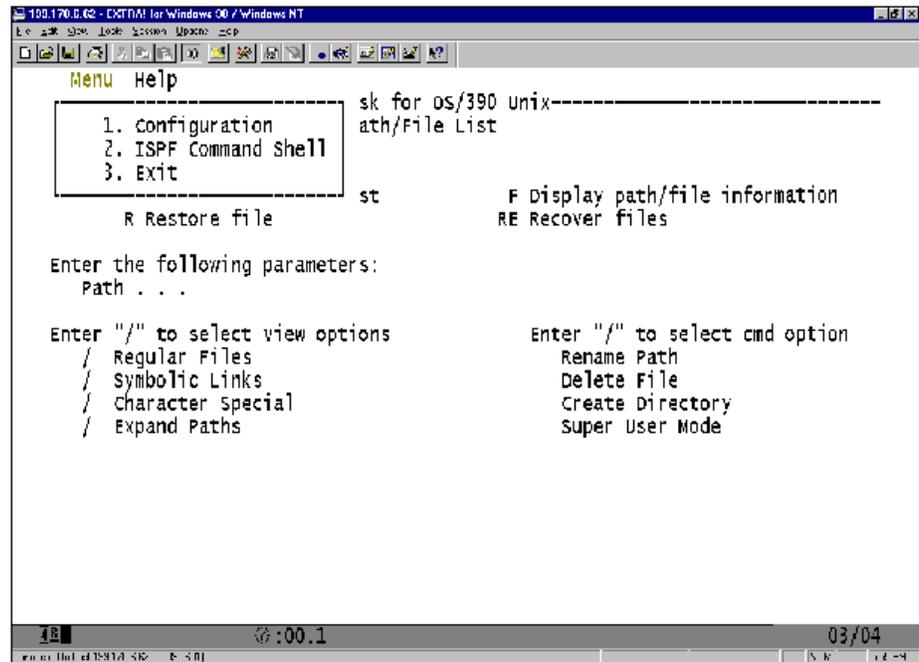
Setting Configuration Options

In the configuration panel, the Files Data Set and the parmlib names are specified. Job card information for JCL generation is specified along with whether to view the JCL before it is submitted.

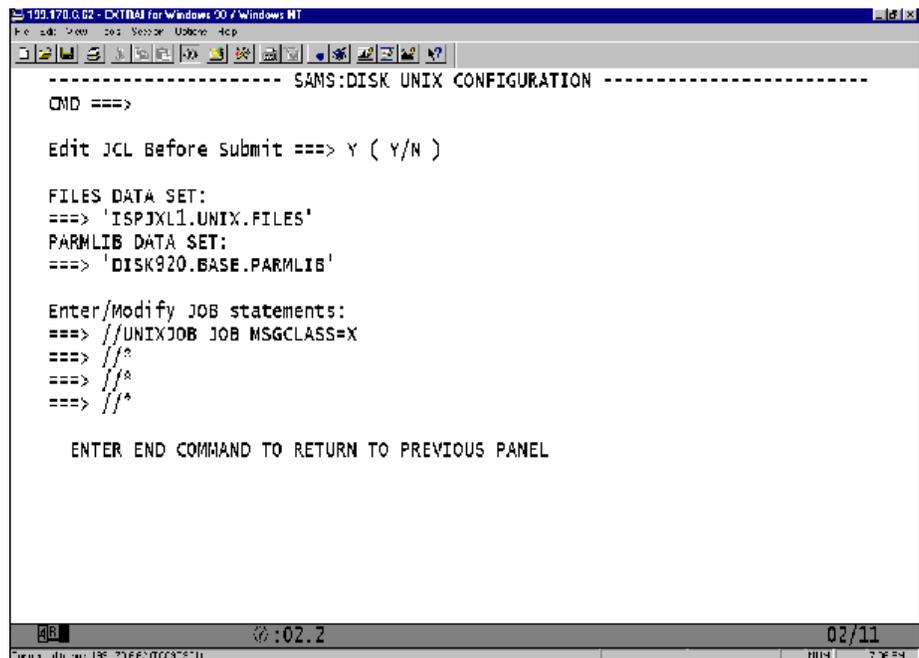
1. Move the cursor to the Menu line.
2. Press Enter.



3. Enter 1 on the line to select Configuration.
4. Press Enter.



5. Enter the Files Data Set name.
6. Enter the PARMLIB data set name.
7. Customize the JCL statements with job card information.
8. Set the flag (Y or N) to indicate whether to review the JCL before it is submitted.



Explicit File Record Display

This action displays details about a backed up or archived file.

1. On the opening panel OPTION line, enter F.
2. On the PATH line, enter the path and file name of the file whose backup/archive record you want to see.
3. Press Enter.

```

SAMS:Disk for OS/390 Unix
-----
Path/File List

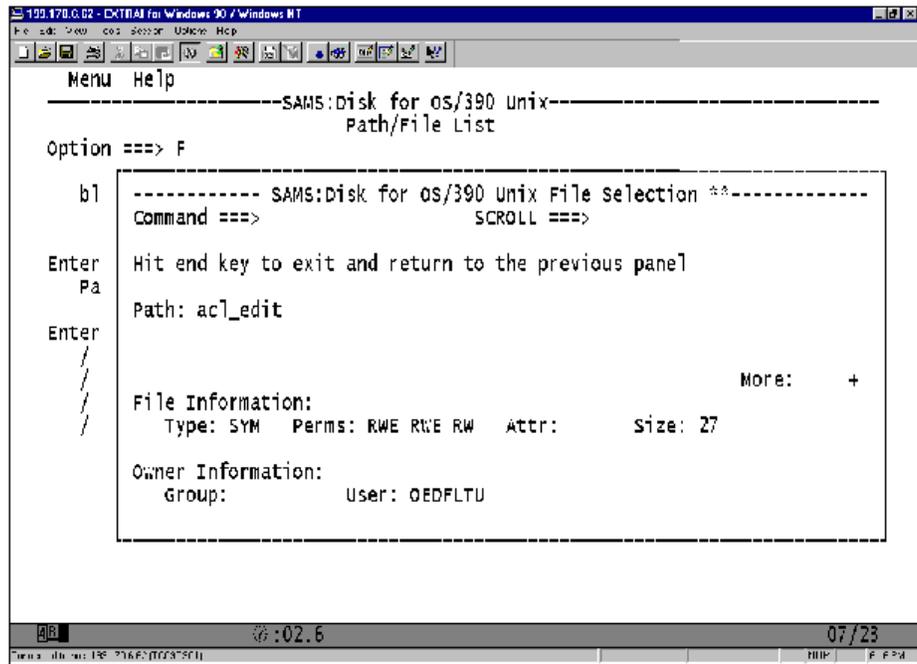
Option ==> F

blank Display path/file list      = Display path/file information
R Restore file                    R: Recover files

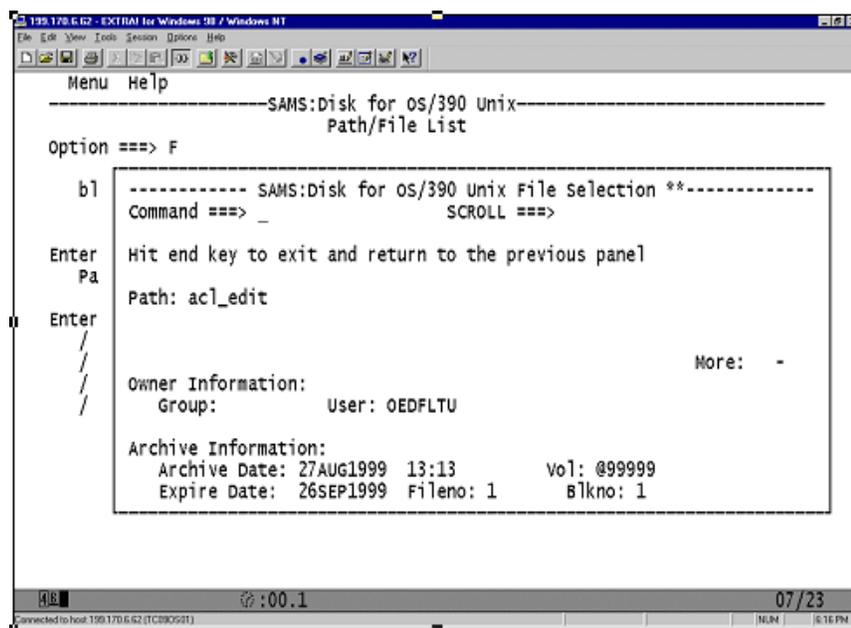
enter the following parameters:
Path . . . /bin/acl_edit

Enter "/" to select view options      Enter "/" to select end option
/ Regular Files                      Rename Path
/ Symbolic Links                     Delete File
/ Character Special                  Create Directory
/ Expand Paths                       Super User Mode
  
```

4. If the record is not found, a message appears. If the record is found, it appears in a scrollable box. Part 1 is shown.
 - PATH: File name.
 - Type: File type (Fil, Sym, Chr, and so on).
 - PERMS: Permission flags of the file at backup/archive time.
 - Attr: Currently not used.
 - Size: Size of the file. Size is in bytes unless otherwise indicated.
 - Group: RACF group ID.



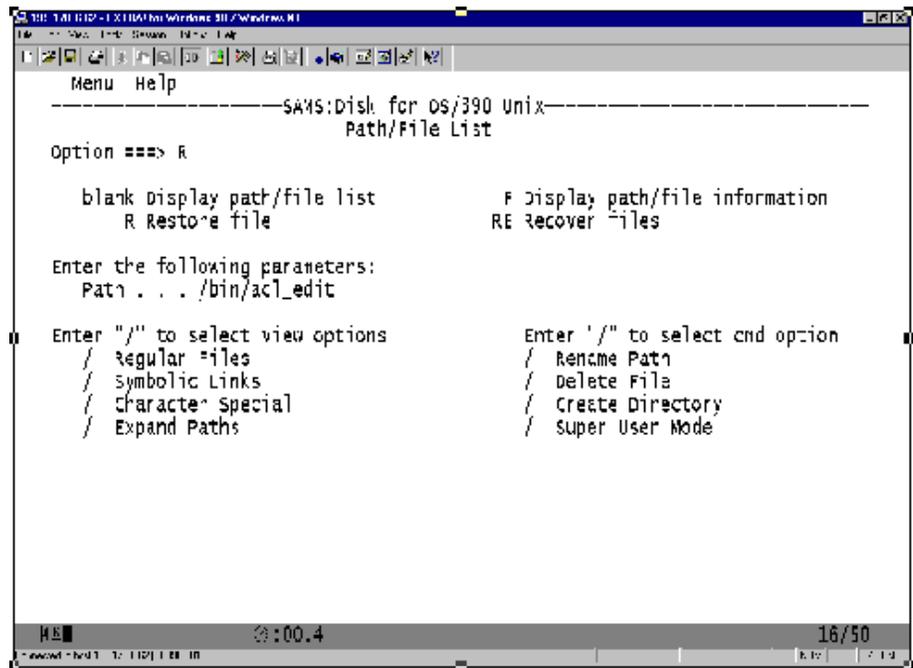
5. Press scroll down to see the rest of the record.
 - User: RACF user ID.
 - Archive Date: Date and time the file was backed up or archived.
 - Expire Date: Date the file expires and is deleted from the Files Data Set.
 - Vol: Key to the ARCHVOLS record that has information about the media upon which the backed up or archived file was placed.
 - Fileno: File position of the file on the backup/archive media.
 - Blkno: Block number of the file on the backup/archive media.



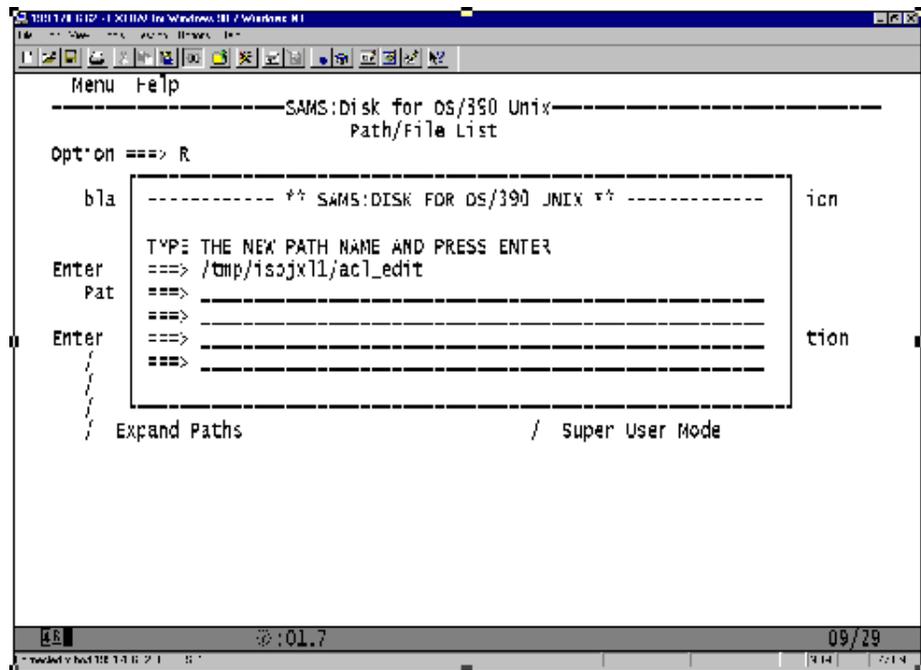
Explicit File Restore

This example restores a file with a new path name specified. It also deletes the file if it exists and creates the directory if it does not exist. Invoke Superuser mode to avoid any permissions issues.

1. On the opening panel OPTION line, enter R.
2. On the PATH line, enter the path and file name of the file to be restored.
3. Type / next to Rename Path.
4. Type / next to Delete File.
5. Type / next to Create Directory.
6. Type / next to Super User Mode.
7. Press Enter.



8. Enter in the new path and file name.
9. Press Enter.



10. JCL appears. You can modify it, as necessary.
11. Type SUBMIT at the COMMAND → line.

12. Press Enter.

```

JOB10000000-10000000 in Windows.MI / Windows.MI
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT      ISPJXL1.SPFTEMPL.CNTL          columns 00001 00072
Command ==>                               scroll ==> CSR
***** Top of Data *****
000001 //UNIXJOB JOB CLASS=X
000002 //*
000003 //*          JCLLIB ORDER=SDX1120.PROCLIB
000004 //*
000005 //*
000006 //*  RESTORE FILE
000007 //*
000008 //RESTORE EXEC RESTOREU
000009 //FILES  DC DISP=SHR,DSN=ISPJXL1.JNCL.FILES
000010 //PARMLIB DC DISP=SHR,DSN=DISK920.BASE.PARMLIB
000011 //SYSIN  DC *,DLM=SS
000012 RESTORE PATH=/bin/ac1_edit,
000013 NEWPATH=/tmp/ispjxl1/ac1_edit,
000014 SCRATCH,
000015 DIRCREAT,
000016 SUPRUSER,
000017 TIMEDATE=131327AUG1999
000018 $$
***** Bottom of Data *****
:00.1
04/15

```

Explicit Path Recovery

This example shows recovering a path with the new path name specified. It also deletes a file if it exists and creates the directory if it does not exist. Invoke Superuser mode to avoid permissions issues.

1. On the opening panel OPTION line, enter RE.
2. On the PATH line, enter the Path name to be recovered.
3. Type / next to Rename Path.
4. Type / next to Delete File.
5. Type / next to Create Directory.
6. Type / next to Super User Mode.
7. Press Enter.

```

Menu Help
-----SAMS:Disk for OS/390 Unix-----
Path/File List

option ==> RE

blank display path/file list          F Display path/file information
R Restore file                        RE Recover files

Enter the following parameters:
Path . . . /bin

Enter '/' to select view options      Enter '/' to select cmd option
/ Regular Files                       / Rename Path
/ Symbolic Links                      / Delete File
/ Character Special                   / Create Directory
/ Expand Paths                        / Super User Mode

```

8. Type the new path and file name.
9. Press Enter.

```

Menu Help
-----SAMS:Disk for OS/390 Unix-----
Path/File List

Option ==> RE

bla
Enter Pat
Enter
/
/ Expand Paths

----- ** SAMS:DISK FOR OS/390 UNIX ** -----
TYPE THE NEW PATH NAME AND PRESS ENTER
==> /tmp/ispjxl1
==>
==>
==>
/ Super User Mode

```

10. JCL appears. You can modify it, as necessary.
11. Type SUBMIT at the COMMAND → line.

12. Press Enter.

```

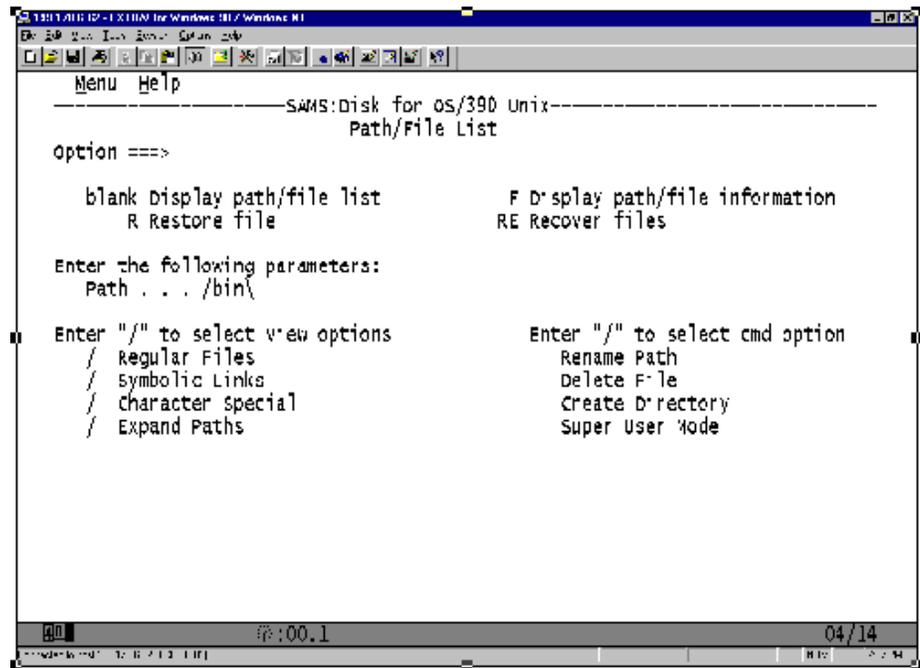
EDIT      ISPJXL1.SPFTEMPL.CNTL          Columns 00001 00072
Command ==>                               Scroll ==> CSR
***** Top of Data *****
000001 //UNIXJOB JOB MSGCLASS=X
000002 //*
000003 //*          JCLLIB ORDER=SDX1120.PROCLIB
000004 //*
000005 //*
000006 //* RECOVER PATH
000007 //*
000008 //RECOVER EXEC PMSU
000009 //FILES  DD DISP=SHR,DSN=ISPJXL1.JNX.FILES
000010 //PARMLIB DD DISP=SHR,DSN=DISK920.BASE.PARMLIB
000011 //SYSIN  DD *,DLM=SS
000012 SCAN UNXINDEX
000013 SELECT PATH=/bin
000014 RECOVER NEWPATH=/tmp/ispjx*1,
000015 SCRATCH,DIRCREAT,SUPRUSER
000016 $S
000017 //PRINT.FILES DD DISP=SHR,DSN=ISPJXL1.UNIX.FILES
000018 //PRINT.PARMLIB DD DISP=SHR,DSN=DISK920.BASE.PARMLIB
***** Bottom of Data *****

```

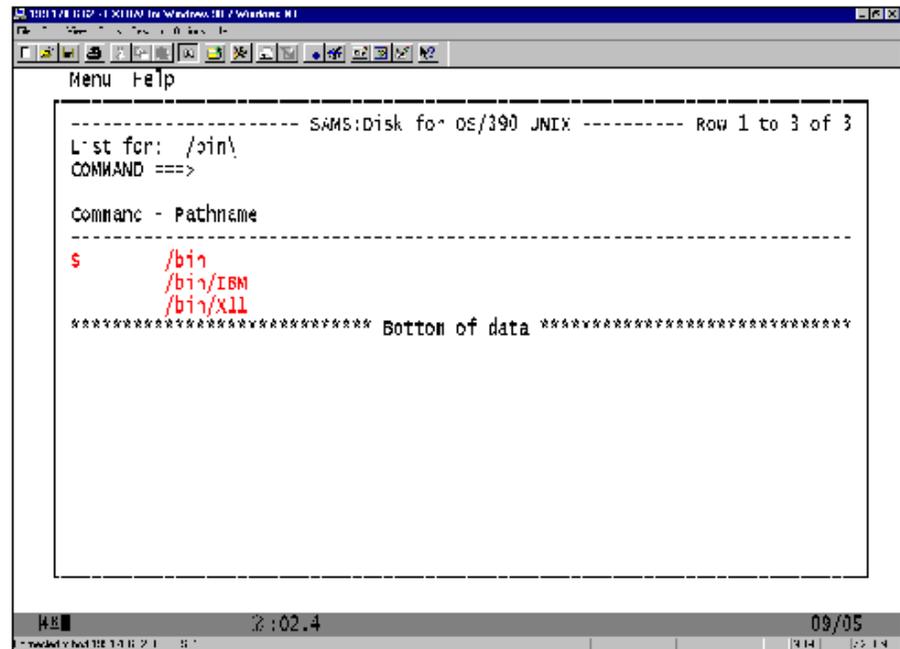
Displaying the Path List

This example displays a list of paths and files and the file backup/archive record.

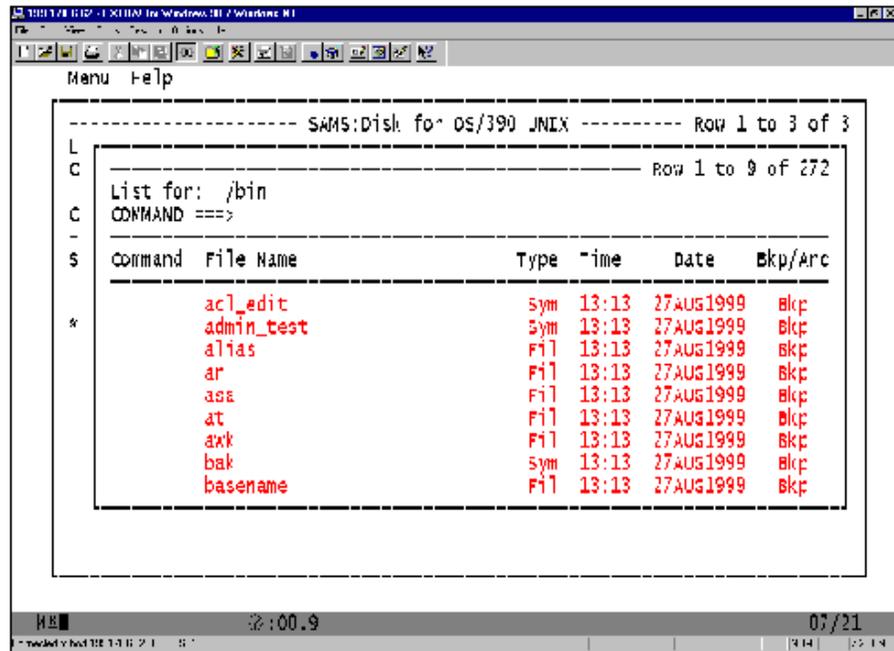
1. On the opening panel, leave the OPTION line blank.
2. On the PATH line, enter the path name of the path to be displayed. You can use patterns in the path name to list more than one path.
3. Press Enter.



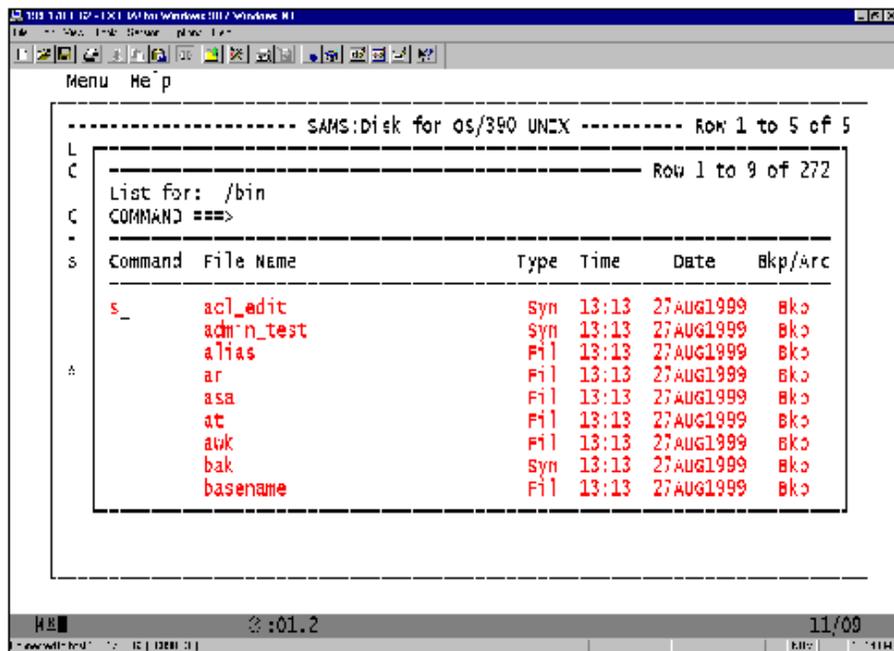
4. Type an S in the Command column next to the path whose files you want to see.
5. Press Enter.



- A list of files appears.



- Place an S in the Command column next to the path whose files you want to see.
- Press Enter.



8. The record is displayed in a scrollable box. Part 1 is currently displayed.
 - PATH: File name.
 - Type: File type (Fil, Sym, Chr, and so on)
 - PERMS: Permission flags of the file at backup/archive time.
 - Attr: Currently not used.
 - Size: Size of the file. Size is in bytes unless otherwise indicated.
 - Group: RACF group ID.
 - User: RACF user ID.

The screenshot shows a terminal window titled "193.170.6.62 - EXTRM for Windows 98 / Windows NT". The window contains a scrollable box with the following text:

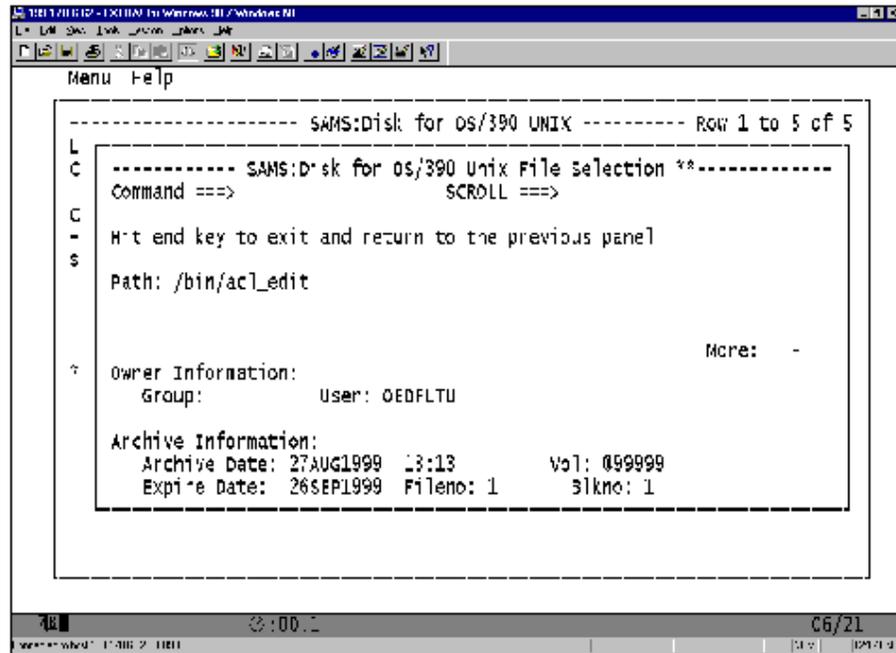
```

----- SAMS:Disk for OS/390 UNIX ----- Row 1 to 5 of 5
L
C ----- SAMS:Disk for OS/390 Unix File selection **-----
C Command ==> _ SCROLL ==>
- Hit end key to exit and return to the previous panel
S Path: /bin/acl_edit

* File Information:
  Type: SYM  Perms: RWE RWE RW  Attr:      size: 27
  Owner Information:
  Group:      User: OEDFLTU
  
```

At the bottom of the terminal window, it shows "06/21" and "Connected to host 193.170.6.62 (TCORGS32)".

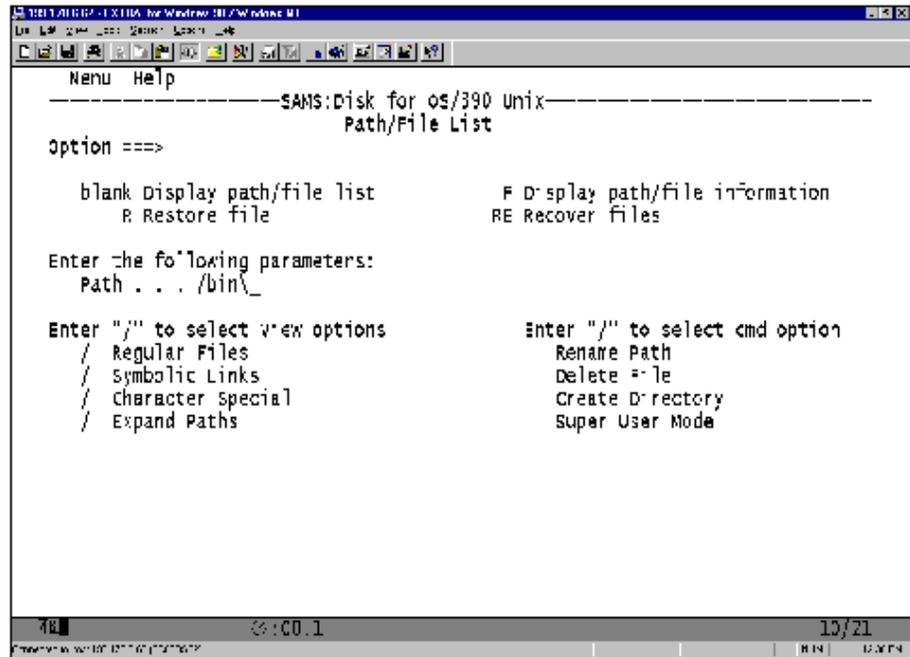
9. Scroll down to see the rest of the record.
 - Archive Date: Date and time the file was backed up or archived.
 - Expire Date: Date the file expires and is deleted out of the BrightStor CA-Disk for USS Files Data Set.
 - Vol: Key to the ARCHVOLS record that has information about the media upon which the backed up or archived file was placed.
 - Fileno: File position of the file on the backup/archive media.
 - Blkno: Block number of the file on the backup/archive media.



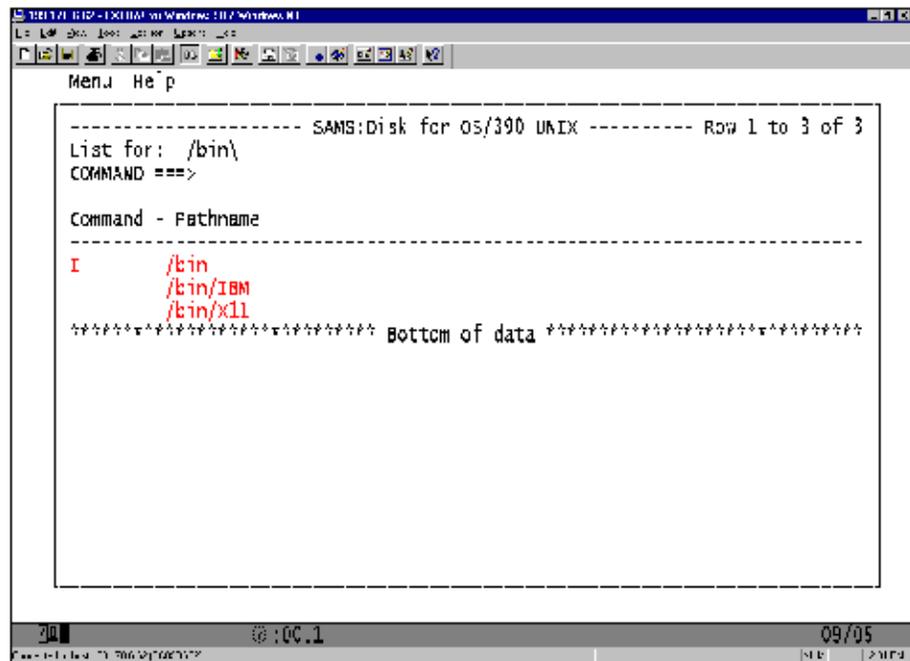
Recover Paths

This example shows how to recover all paths from a list of paths except one:

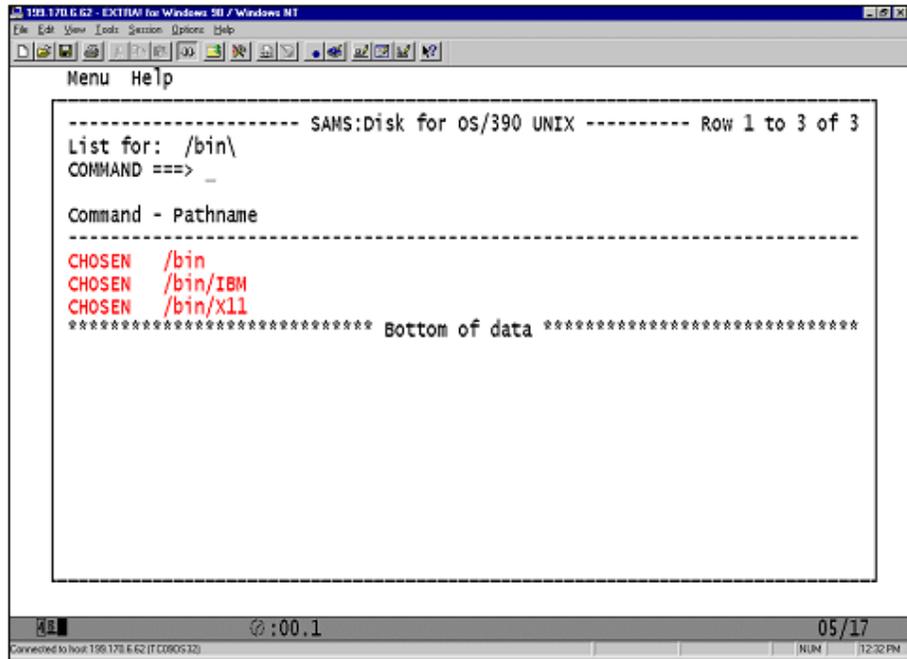
1. On the opening panel, leave the OPTION line blank.
2. On the PATH line, enter the path name of the path to be recovered. You can use patterns in the path name to list more than one path.
3. Press Enter.



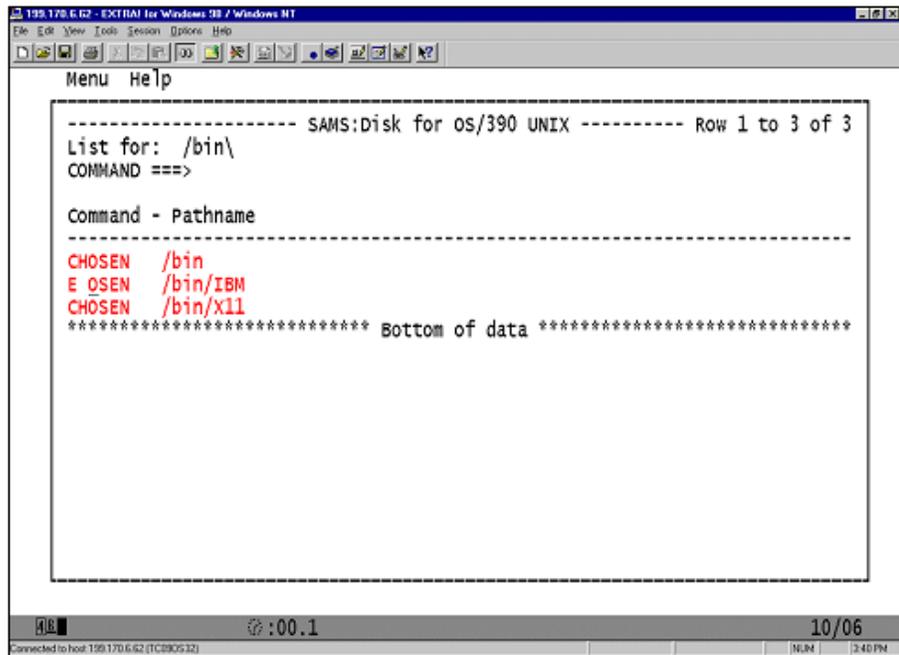
4. Place an I in the Command column next to the path whose files you want to see.
5. Press Enter.



- All subdirectories are automatically selected.



6. Type an E in the Command column next to the path whose files you want to exclude.
7. Press Enter.

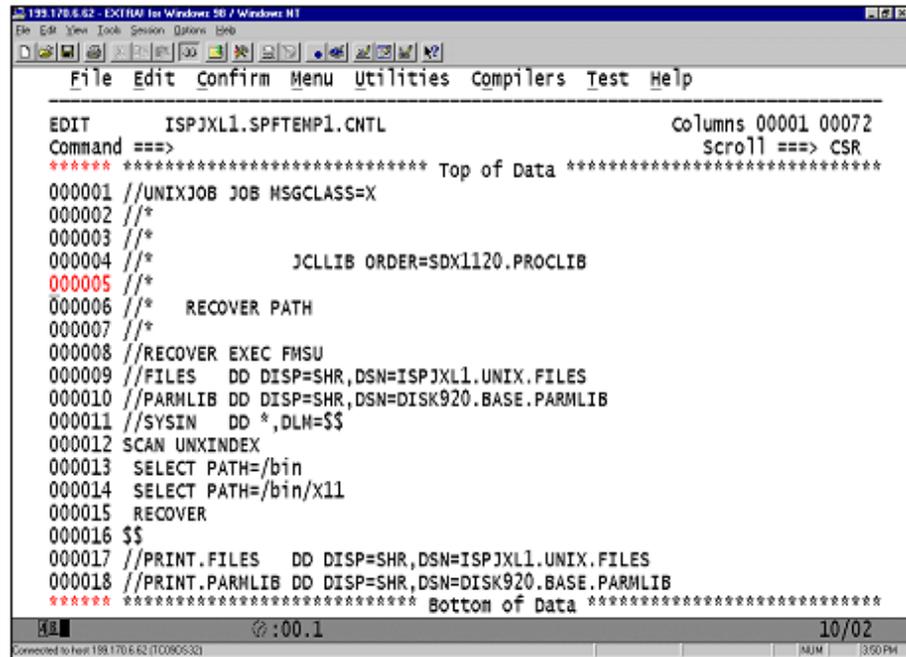


8. The word CHOSEN disappears from the command line.
9. Press the End key.

```
----- SAMS:Disk for OS/390 UNIX ----- Row 1 to 3 of 3
List for: /bin\
COMMAND ==> _

Command - Pathname
-----
CHOSEN  /bin
        /bin/IBM
CHOSEN  /bin/x11
***** Bottom of data *****
```

10. JCL is now displayed. You can modify, as necessary.
11. Type in SUBMIT at the COMMAND → line.
12. Press Enter.



```
199.170.6.62 - EXTRA for Windows 98 / Windows NT
File Edit Confirm Menu Utilities Compilers Test Help
EDIT      ISPJXL1.SPFTENPL.CNTL      Columns 00001 00072
Command ==>                          Scroll ==> CSR
***** ***** Top of Data *****
000001 //UNIXJOB JOB MSGCLASS=X
000002 //*
000003 //*
000004 //*          JCLLIB ORDER=SDX1120.PROCLIB
000005 //*
000006 //* RECOVER PATH
000007 //*
000008 //RECOVER EXEC FMSU
000009 //FILES DD DISP=SHR,DSN=ISPJXL1.UNIX.FILES
000010 //PARMLIB DD DISP=SHR,DSN=DISK920.BASE.PARMLIB
000011 //SYSIN DD *,DLM=$$
000012 SCAN UNINDEX
000013 SELECT PATH=/bin
000014 SELECT PATH=/bin/X11
000015 RECOVER
000016 $$
000017 //PRINT.FILES DD DISP=SHR,DSN=ISPJXL1.UNIX.FILES
000018 //PRINT.PARMLIB DD DISP=SHR,DSN=DISK920.BASE.PARMLIB
***** ***** Bottom of Data *****
:00.1 10/02
Connected to host 199.170.6.62 (TC090532) 3:50 PM
```

Restore from list

This example shows how to restore files from a list.

1. On the opening panel, leave the OPTION line blank.
2. On the PATH line, enter the path name of the path to be recovered. Patterns can be used in the path name to list more than one path.
3. Press Enter.

```

199.170.6.62 - EXTRA! for Windows 98 / Windows NT
File Edit View Tools Session Options Help
-----SAMS:Disk for OS/390 Unix-----
Path/File List

option ==>

blank Display path/file list          F Display path/file information
R Restore file                        RE Recover files

Enter the following parameters:
Path . . . /bin\_

Enter "/" to select view options      Enter "/" to select cmd option
/ Regular Files                       Rename Path
/ Symbolic Links                      Delete File
/ Character Special                   Create Directory
/ Expand Paths                        Super User Mode

10/21
Connected to host 199.170.6.62 (1 C080532)

```

4. Type an S in the Command column next to the path whose files you want to see.
5. Press Enter.

```

199.170.6.62 - EXTRA! for Windows 98 / Windows NT
File Edit View Tools Session Options Help
-----SAMS:Disk for OS/390 UNIX ----- Row 1 to 3 of 3
List for: /bin\
COMMAND ==>

Command - Pathname
-----
S_      /bin
        /bin/IBM
        /bin/x11
***** Bottom of data *****

09/05
Connected to host 199.170.6.62 (1 C080501)

```

6. A list of files appears.
7. Type an R next to the file you want to restore.

- Press Enter.

```

----- SAMS:Disk for OS/390 UNIX ----- Row 1 to 3 of 3
L
C
List for: /bin
C
COMMAND ==>
-----
s
Command File Name Type Time Date Bkp/Arc
-----
R_      acl_edit      Sym 13:13 27AUG1999 Bkp
*      admin_test   Sym 13:13 27AUG1999 Bkp
       alias       Fil 13:13 27AUG1999 Bkp
       ar         Fil 13:13 27AUG1999 Bkp
       asa       Fil 13:13 27AUG1999 Bkp
       at        Fil 13:13 27AUG1999 Bkp
       awk       Fil 13:13 27AUG1999 Bkp
       bak       Sym 13:13 27AUG1999 Bkp
       basename  Fil 13:13 27AUG1999 Bkp

```

- The word CHOSEN appears next to the file name.
- Press the End key.

```

----- SAMS:Disk for OS/390 UNIX ----- Row 1 to 3 of 3
L
C
List for: /bin
C
COMMAND ==> _
-----
s
Command File Name Type Time Date Bkp/Arc
-----
CHOSEN  acl_edit      Sym 13:13 27AUG1999 Bkp
*      admin_test   Sym 13:13 27AUG1999 Bkp
       alias       Fil 13:13 27AUG1999 Bkp
       ar         Fil 13:13 27AUG1999 Bkp
       asa       Fil 13:13 27AUG1999 Bkp
       at        Fil 13:13 27AUG1999 Bkp
       awk       Fil 13:13 27AUG1999 Bkp
       bak       Sym 13:13 27AUG1999 Bkp
       basename  Fil 13:13 27AUG1999 Bkp

```

- The Path List appears again.

The MOVE/COPY function of BrightStor CA-Disk for USS provides the ability to copy and move UNIX files around the UNIX file system. This capability also includes the ability to select files based upon the HFS data sets they reside in and to place them directly into other named HFS data sets.

Therefore, some functions that can be accomplished with the MOVE/COPY function are:

- Consolidate HFS data sets
- Expand HFS data sets
- Duplicate a directory structure without any files contained within it
- Duplicate a directory structure including the files but the files are initialized as empty
- Duplicate a directory structure including files
- During a MOVE/COPY, alter Permissions
- During a MOVE/COPY, alter user or group owner

Considerations

Following are some considerations that need to be taken into account when using the MOVE and COPY commands:

- The difference between a MOVE command and a COPY command is a MOVE command deletes the source directories and files after the move is completed. A COPY command leaves all source directories and files in place after the copy is completed.
- In a MOVE command, if the SELECT/EXCLUDE command contains only the PATH= parameter, this implies, after the directories and files are moved, a deletion of the source files and their directories. The deletion is not accomplished until the whole directory structure has been moved. This is done to ensure recovery should the MOVE action fail for whatever reason.

- In a MOVE command, if the SELECT/EXCLUDE command contains a PATH= and a FILE= parameter, this implies that only moved files will be deleted – directories will not be deleted (whether a PATH= parameter is present).
- The difference between the parameters SCRATCHFIL and OVERWRITE is as follows:
 - Use of SCRATCHFIL means that any processes with the file open will continue to see the file, but new processes will not.
 - Use of OVERWRITE causes all processes, even those that currently have the file open, to see the newly written data.
- The DSCL language provides a powerful capability to select, copy, move, and so on, files. It is possible to combine many action commands into one command stream. The more that are used, the more chance there is for errors that could produce undesirable results. It is highly recommended that you run the batch jobs in simulate mode, carefully reviewing the report output, before placing the commands in a live run. The following example illustrates an error that may be unintentional:

```
//job.....  
//EXEC DMSU  
//SYSIN DD *  
  SET MODE=SIMULATE  
  SCAN HFS=HFS.USER1  
  SELECT PATH=/user1\  
  MOVE TOHFS=HFS.USER1.EXPANDED  
  BACKUP RETPD=30  
  UREPORT UATT  
/*
```

In this case, the files and directories in /user1 are going to be moved to a new HFS file. Built into this command structure is also a backup of these directories and files as well as an attribute report. Because the PATH= parameter is specified, the source directories and files will not be deleted until after all the directories and files have been moved. Thus the backup and report will reflect the source location and not the target location.

MOVE/COPY Command and Parameters

Following is a description of the MOVE and COPY commands and their associated parameters.

COPY

The COPY command is used to indicate that a copy operation is to be performed. Following is a list of parameters associated with the COPY command:

Parameter	Description
TOHFS=	The name of the target HFS data set where files are to be copied. This can be an existing HFS data set, or one just allocated and mounted using the HFSCREATE command. This parameter is mutually exclusive with the TOPATH= parameter. Either the TOHFS= or the TOPATH= parameter (not both) must be specified. Example: TOHFS=HFS.DATA.SET
TOPATH=	Specify the path name where the selected paths/files are to be copied. This parameter is mutually exclusive with the TOHFS parameter. Either the TOHFS= or the TOPATH= parameter (not both) must be specified. Example: TOPATH=/tmp/users
DIRCREAT	Use this in conjunction with the TOPATH= and TOHFS= parameters. Specify this parameter to indicate that if a path does not exist, then create the path. If it is uncertain that all the directories exist, this parameter should be specified to ensure all directories are created. This is an optional parameter. Example: TOPATH=/tmp/users,DIRCREAT
GID=	To change the GID of the paths/files to be copied, specify the new numeric GID here. This parameter is mutually exclusive with the GOWNER parameter and is optional. Example: GID=25
GOWNER=	To change the GID of the paths/files to be copied, specify the new group name here. The associated numeric GID will be retrieved and placed in the path/file attributes. This parameter is mutually exclusive to the GID parameter and is optional. Example: GOWNER=SYSTEMS

Parameter	Description
MPOINT=	<p>If the TOHFS= data set is not mounted, specify the UNIX file system mount point with this parameter. This parameter is optional.</p> <p>Note: Specification of this parameter indicates that the file system will be mounted. If the HFSCREATE command was used with the MPOINT parameter to create the HFS data set being used in this command, then the HFS data set is already mounted and specification of this parameter results in an error.</p> <p>Example: MPOINT=/tmp/users</p>
NOLOAD	<p>With this parameter specified, all files are created as empty files. The data from the source file is not copied to the target files. This parameter is optional.</p> <p>Example: TOPATH=/tmp/users,NOLOAD</p>
OVERWRITE=	<p>This optional parameter specifies if it is okay to overwrite existing UNIX files that have the same name as the one being moved. The SCRATCHFIL parameter overrides the OVERWRITE parameter. The options are:</p> <ul style="list-style-type: none"> ■ NO—Do not overwrite ■ YES—Overwrite ■ NEWER—Overwrite only if the file being moved is newer. <p>Example: OVERWRITE=NEWER</p>
PATHONLY	<p>This parameter is used to only create a copy of the source directory structure. No files will be copied. This parameter is optional.</p> <p>Example: TOPATH=/tmp/users,PATHONLY</p>
PERMS	<p>By default, all permissions are copied and kept the same as the source location. To change the permissions, specify the new three-digit permission codes here. For more information, see UNIX File Permissions in the chapter “Basic System – DMSU Procedure.” This parameter is optional.</p> <p>Example: PERMS=751</p>
SCRATCHFIL	<p>To scratch an existing target UNIX file that has the same name as the one being copied, specify this parameter. This optional parameter overrides the OVERWRITE= parameter.</p> <p>Example: TOPATH=/tmp/users,SCRATCHFIL</p>

Parameter	Description
UID=	<p>To change the UID of the paths/files to be copied, specify the new numeric UID here. This optional parameter is mutually exclusive to the UOWNER parameter.</p> <p>Example: UID=301</p>
UOWNER=	<p>To change the UID of the paths/files to be copied, specify the new user ID here. The associated numeric UID is retrieved and placed in the path/file attributes. This optional parameter is mutually exclusive to the UID parameter.</p> <p>Note: The UOWNER parameter should be used with caution. Some installations assign user IDs with the same UID number. If an installation has the same UID for multiple users, then it is possible a user may not be the owner of a file after the copy is completed.</p> <p>Example: UOWNER=ispjx11</p>
VERIFY	<p>Specify this optional parameter to verify that certain attributes between the source file and the target files are identical. Use of this parameter increases the amount of time required to complete the copy. The attributes verified are dependent on the function being performed and the parameters specified. For example, if the PERMS parameter is specified, verification is done after the copy where the permissions changed.</p> <p>Example: TOHFS=HFS.DATA.SET,VERIFY</p>

MOVE

The MOVE command is used to indicate that a move operation is to be performed. Following is a list of parameters associated with the MOVE command:

Parameter	Description
TOHFS=	<p>The name of the target HFS data set where files are to be moved. This can be an existing HFS data set, or one just allocated and mounted using the HFSCREATE command. This parameter is mutually exclusive with the TOPATH= parameter. Either the TOHFS= or the TOPATH= parameter (not both) must be specified.</p> <p>Example: TOHFS=HFS.DATA.SET</p>
TOPATH=	<p>Specify the path name where the selected paths/files are to be moved. This parameter is mutually exclusive with the TOHFS parameter. Either the TOHFS= or the TOPATH= parameter (not both) must be specified.</p> <p>Example: TOPATH=/tmp/users</p>
DIRCREAT	<p>Used in conjunction with the TOPATH= and TOHFS= parameters. Specify this parameter to indicate that if a path does not exist, then create the path. If it is uncertain that all the directories exist, this parameter should be specified to ensure all directories are created. This is an optional parameter.</p> <p>Example: TOPATH=/tmp/users,DIRCREAT</p>
GID=	<p>To change the GID of the paths/files to be moved, specify the new numeric GID here. This parameter is mutually exclusive to the GOWNER parameter and is optional.</p> <p>Example: GID=25</p>
GOWNER=	<p>To change the GID of the paths/files to be moved, specify the new group name here. The associated numeric GID will be retrieved and placed in the path/file attributes. This parameter is mutually exclusive to the GID parameter and is optional.</p> <p>Example: GOWNER=SYSTEMS</p>

Parameter	Description
MPOINT=	<p>If the TOHFS= data set is not mounted, specify the UNIX file system mount point with this parameter. This parameter is optional.</p> <p>Note: Specification of this parameter indicates that the file system will be mounted. If the HFSCREATE command was used with the MPOINT parameter to create the HFS data set being used in this command, then the HFS data set is already mounted and specification of this parameter will result in an error.</p> <p>Example: MPOINT=/tmp/users</p>
OVERWRITE=	<p>This optional parameter specifies if it is okay to overwrite existing UNIX files that have the same name as the one being moved. The SCRATCHFIL parameter overrides the OVERWRITE parameter. The options are:</p> <ul style="list-style-type: none"> ■ NO – Do not overwrite ■ YES – Overwrite ■ NEWER – Overwrite only if the file being moved is newer. <p>Example: OVERWRITE=NEWER</p>
PERMS	<p>By default, all permissions are copied and kept the same as the source location. To change the permissions, specify the new three-digit permission codes here. For more information, see UNIX File Permissions in the chapter “Basic System – DMSU Procedure.” This parameter is optional.</p> <p>Example: PERMS=751</p>
SCRATCHFIL	<p>To scratch existing UNIX files that have the same name as the one being moved, then specify this optional parameter. This parameter overrides the OVERWRITE parameter.</p> <p>Example: TOPATH=/tmp/users, SCRATCHFIL</p>
UID=	<p>To change the UID of the paths/files to be moved, specify the new numeric UID here. This optional parameter is mutually exclusive to the UOWNER parameter.</p> <p>Example: UID=301</p>

Parameter	Description
UOWNER=	<p>To change the UID of the paths/files to be moved, specify the new user ID here. The associated numeric UID is retrieved and placed in the path/file attributes. This optional parameter is mutually exclusive to the UID parameter.</p> <p>Note: The UOWNER parameter should be used with caution. Some installations assign user IDs with the same UID number. If an installation has the same UID for multiple users, then it is possible a user may not be the owner of a file after the copy is completed.</p> <p>Example: UOWNER=ispjx11</p>
VERIFY	<p>Specify this optional parameter to verify that certain attributes between the source file and the target files are identical. Use of this parameter increases the amount of time required to complete the move. The attributes verified are dependent on the function being performed and the parameters specified. For example, if the PERMS parameter is specified, verification is done after the copy where the permissions changed.</p> <p>Example: TOHFS=HFS.DATA.SET,VERIFY</p>

User Examples:

Following are examples of the MOVE command:

- Move the directory /user1 and files and all its subdirectories and files to /users/user1 using the FIND command. Create the directory if it does not exist. If the files do exist, overwrite them only if they are newer files.

```
//job....  
//EXEC DMSU  
//SYSIN DD *  
FIND PATH=/user1\  
MOVE TOPATH=/users/user1,  
DIRCREAT,  
OVERWRITE=NEWER  
/*
```

- A new user is being added to the UNIX system. Create an HFS data set with the name of HFS.USER2 and make it SMS managed by assigning it a storage class of HFSUSERS. Duplicate the directory of user1 but without copying the files. The new user is user2 and the mount point for HFS data set is /users/user2. Set the UID of user2 to 85 to allow user2 access to these files.

```
//job.....
//EXEC DMSU
//SYSIN DD *
HFSCREATE DSN=HFS.USER2,
SPACE=(20,20),
CYLS,
STORCLAS=HFSUSERS,
MPOINT=/users/user2

FIND PATH=/users/user1\
MOVE TOPATH=/users/user2,
DIRCREAT,
PATHONLY
UID=85
/*
```

- This example shows how to use the MOVE/COPY function to increase the size of an HFS data set that has run out of space. In this case, user2 has run out of space. The example is in the form of a four step JCL job stream.

```
//job.....
//*****
//* STEP1 - CREATE A TEMPORARY HFS DATA SET AND *
//* COPY ALL THE FILES FROM HFS.USER2 *
//* TO IT. UNMOUNT HFS.USER2 WHEN DONE. *
//*****
//EXEC DMSU
//SYSIN DD *
**** Create the temp HFS data set
HFSCREATE DSN=HFS.TEMP,
SPACE=(200,50),
CYLS,
STORCLAS=HFSUSERS,
MPOINT=/tmp/user2
**** Copy files to it
FIND HFS=HFS.USER2,UNMOUNT
COPY TOHFS=HFS.TEMP,
DIRCREAT,
SCRATCHFILE,
SCRATCHPATH
/*
//*****
//* STEP2 - RENAME HFS.USER2 TO HFS.USER2.SAVE *
//*****
//RENAME EXEC IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
ALTER HFS.USER2 -
NEWNAME(HFS.USER2.SAVE)
/*
//*****
//* STEP 3 - CREATE A LARGER HFS.USER2 AND COPY*
//* THE FILES FROM THE TEMPORARY LOCATION TO *
//* IT. UNMOUNT THE TEMP HFS WHEN DONE *
//*****
```

```

//job....
//EXEC DMSU
//SYSIN DD *
HFSCREATE DSN=HFS.USER2,
SPACE=(200,50),
CYLS,
STORCLAS=HFSUSERS,
MPOINT=/users/user2
FIND HFS=HFS.TEMP,UNMOUNT
COPY TOHFS=HFS.USER2,
DIRCREAT,
SCRATCHFILE,
SCRATCHPATH
/*
//*****
//* STEP 4 - DELETE THE HFS.TEMP DATA SET *
//*****
//DELETE EXEC IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE DSN=HFS.TEMP
/*

```

- Consolidate all HFS files starting with the name of HFS.PROJECTX into one HFS file called HFS.PROJECTX.TOTAL. The new HFS file is to be SMS managed with a storage class of HFSPROJS. Before deleting the source directories and files, verify the target files user and group owners are the same as the source files as well as the permissions.

```

//job....
//EXEC DMSU
//SYSIN DD *
HFSCREATE DSN=HFS.PROJECTX.TOTAL,
SPACE=(500,250),
CYLS,
STORCLAS=HFSPROJS,
MPOINT=/projects/projectx
SCAN HFS=HFS.PROJECTX\,
EXCHFS=HFS.PROJECTX.TOTAL,
SELECT PATH=\,
MOVE TOHFS=HFS.PROJECTX.TOTAL,
VERIFY
/*

```

- A software project has just been completed and all the files need to be copied to a directory called /production/newapp. To help ensure the files are not modified once they are in their new location, change the permissions such that everyone only has read and execute authority for these files. The project to be copied is located in the /projects/projectx directory. Invoke Superuser mode to accomplish this.

```

//job....
//EXEC DMSU
//SYSIN DD *
SET UID=0
SCAN PATH=/projects/projectx\,
SELECT PATH=\,
COPY TOPATH=/production/newapp,
PERMS=(555)
/*

```

Installation

This appendix provides the information necessary to install, customize, evaluate, and maintain the BrightStor CA-Disk for USS system.

Overview

The overall steps involved to install BrightStor CA-Disk for USS are as follows:

1. Decide on a BrightStor CA-Disk for USS load library name and APF authorize it. For a description of this topic, see Step 1. Library Names and Locations in this chapter.
2. Customize and submit the RIMLIB member in the INSTALL library on the distribution tape. For a description of this topic, see Step 2. Download the Related Installation Materials.
3. Follow the SMP/E installation instructions beginning with Step 3. Prepare the SMP/E Environment in this chapter.
4. Create a Files Data Set as described in Appendix B “Files Data Set.”
5. Customize the procedures provided in the Proclib library:
 - DIMU – diagnostics
 - DMSPOOLU – synchronizing tape pools with the Files Data Set
 - DMSU – DSCL command processor for BACKUP and REPORTS
 - FMSU – DSCL command processor for implicit RECOVER
 - IXMAINTU – file data maintenance
 - LISTU – list entries within the Files Data Set
 - REBUILDU – recreate entries within the Files Data Set
 - RELOADU – restore the Files Data Set backed up with the UNLOADU procedure
6. Run test jobs as described in Running Tests in this chapter.

The SMP/E Process

The installation is done with SMP/E using the Receive-Apply-Accept method. SMP/E invokes the linkage editor and IEBCOPY to download the distribution tape, and to install it into the target and distribution libraries.

We have provided sample jobs to set up a complete SMP/E environment. If you already have an established SMP/E environment, skip the steps you do not need to perform.

All of the data sets you need to install BrightStor products with SMP/E are internally defined as DDDEF entries. There is no need for an SMP/E procedure. All you need to do is execute the SMP program using "PARM==..." as documented in the SMP/E manual. You can also use the ISPF interface provided with SMP/E.

This installation process installs BrightStor CA-Disk for USS under the FMID of **SDX1120**.

Note: Do not install BrightStor CA-Disk for USS into the same target and distribution zones as any other product or your operating system because naming conflicts can occur.

BrightStor CA-Disk for USS is distributed on a standard label tape. The VOLSER of the distribution tape and its format is documented in the current BrightStor CA-Disk for USS Cover Letter.

Step 1. Library Names and Locations

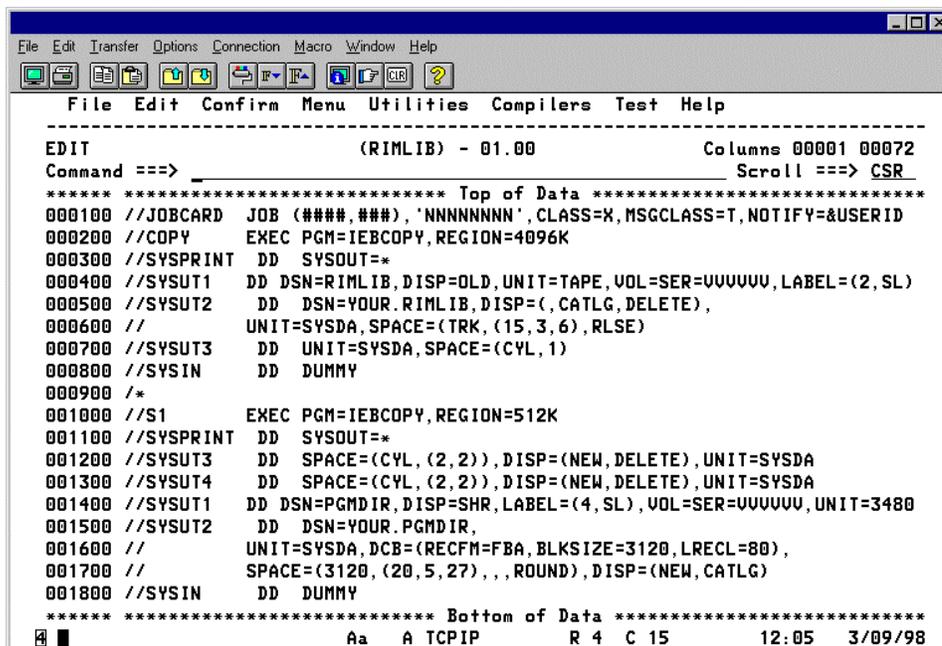
Decide the following:

- A name for the load library contained on the installation tape.
- Which DASD volume to place the load library on. This volume must be on the same system (image) the OS/390 UNIX is running on.
- APF Authorize the library. Doing this in advance permits you to test executions immediately after downloading the distribution tape. Otherwise, you must wait for an IPL to authorize the library.

- Names and DASD volumes for the following libraries:
 - INSTALL
 - PARMLIB
 - PROCLIB
 - Files Data Set
 - RIMLIB
 - PGMDIR

Step 2. Download the Related Installation Materials Library

The JCL needed to install BrightStor CA-Disk for USS is provided in the Related Installation Materials library (RIMLIB) on the distribution tape (file 2). Important information related to the installation process is provided in the Program Directory library (PGMDIR) on file 4. Run the following JCL to allocate both of these libraries and download them from the tape:



```

File Edit Transfer Options Connection Macro Window Help
-----
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT                               (RIMLIB) - 01.00                Columns 00001 00072
Command ==>                         Scroll ==> CSR
***** ***** Top of Data *****
000100 //JOB CARD JOB (###,###), 'NNNNNNN', CLASS=X, MSGCLASS=T, NOTIFY=&USERID
000200 //COPY EXEC PGM=IEBCOPY, REGION=4096K
000300 //SYSPRINT DD SYSOUT=*
000400 //SYSUT1 DD DSN=RIMLIB, DISP=OLD, UNIT=TAPE, VOL=SER=UUUUUU, LABEL=(2, SL)
000500 //SYSUT2 DD DSN=YOUR.RIMLIB, DISP=(, CATLG, DELETE),
000600 // UNIT=SYSDA, SPACE=(TRK, (15, 3, 6), RLSE)
000700 //SYSUT3 DD UNIT=SYSDA, SPACE=(CYL, 1)
000800 //SYSIN DD DUMMY
000900 /*
001000 //S1 EXEC PGM=IEBCOPY, REGION=512K
001100 //SYSPRINT DD SYSOUT=*
001200 //SYSUT3 DD SPACE=(CYL, (2, 2)), DISP=(NEW, DELETE), UNIT=SYSDA
001300 //SYSUT4 DD SPACE=(CYL, (2, 2)), DISP=(NEW, DELETE), UNIT=SYSDA
001400 //SYSUT1 DD DSN=PGMDIR, DISP=SHR, LABEL=(4, SL), VOL=SER=UUUUUU, UNIT=3480
001500 //SYSUT2 DD DSN=YOUR.PGMDIR,
001600 // UNIT=SYSDA, DCB=(RECFM=FBA, BLKSIZE=3120, LRECL=80),
001700 // SPACE=(3120, (20, 5, 27), , ROUND), DISP=(NEW, CATLG)
001800 //SYSIN DD DUMMY
***** ***** Bottom of Data *****
Aa A TCP/IP R 4 C 15 12:05 3/09/98

```

Note: SMP/E is equipped with ISPF panels that can help you download the RIMLIB and PGMDIR libraries.

The RIMLIB data set contains jobs to install the base product with SMP/E. Customization jobs are located in the installation library.

Full Product Distribution Tapes contain the base release and SMP/E format PTFs. If there are PTFs on the distribution tape, you can receive them at the same time that you receive the base product. However, you should apply and accept the base release before you apply any PTFs.

Step 3. Prepare the SMP/E Environment

Customize and submit RIMLIB member SMP01GBL to define the global CSI data set, allocate the SMPPTS and SMPLOG data sets, and initialize the global zone.

Customize and submit RIMLIB member SMP02TGT to define the target CSI data set, allocate the SMPMTS, SMPSCDS, and SMPSTS data sets, and initialize the target zone.

Customize and submit RIMLIB member SMP03DLB to define the distribution CSI data set and initialize the distribution zone.

Step 4. Allocate Data Sets

Customize and submit RIMLIB member UNX01ALC to create target and distribution data sets for BrightStor CA-Disk for USS.

Step 5. Set DDDEF Entries

Customize and submit RIMLIB member UNX02DDD to create DDDEF entries for BrightStor CA-Disk for USS in the target and distribution zones.

Step 6. Receive the Product

Customize and submit RIMLIB member UNX03REC to receive BrightStor CA-Disk for USS. If there are PTFs on the distribution tape, they are also received at this time. Do not apply the PTFs until after you have accepted the base function for BrightStor CA-Disk for USS.

You can use RIMLIB member RECEIVE to receive all sysmods and hold data that is downloaded from the Computer Associates website.

Step 7. Apply the Product

Customize and submit RIMLIB member UNX04APP to apply the base function for BrightStor CA-Disk for USS.

Step 8. Accept the Product

Customize and submit RIMLIB member UNX05ACC to accept the base function for BrightStor CA-Disk for USS.

Step 9. Apply Maintenance

Maintenance to BrightStor CA-Disk for USS is shipped on your distribution tape and is received at the same time as the base function. Apply this maintenance now.

Customize and submit RIMLIB member UNX06PTF to apply the PTFs. You can accept the PTFs according to the policy of your own installation. If you do not have a policy on accepting PTFs, you can customize and submit RIMLIB member UNX07PTF to accept the PTFs.

When you apply maintenance, you normally encounter SMP/E hold data. Computer Associates uses hold data to notify your SMP/E system of sysmods that have errors or special conditions.

There are two different types of hold data:

- **Internal hold data** – Data that is an instream part of the sysmod instructing you of special conditions:

Sysmod	Description
ACTION	You must perform special processing either before or after you apply this sysmod.
DEP	There is a dependency for this sysmod that you must externally verify.
DELETE	This sysmod deletes a load module. You cannot reverse this type of sysmod with the SMP/E RESTORE command.
DOC	There is a documentation change with this sysmod.
EC	This sysmod requires a hardware engineering change. An EC hold sysmod usually does not have an effect on the product unless the EC is present on the hardware device.
UNLIN	You need to perform a UCLIN either before or after you apply this sysmod.

You must code a bypass operand on your APPLY command to install sysmods that have internal holds. You should only code the bypass operand after you have performed the required action, or if you are performing the action after the apply, if that is appropriate.

- **External hold data**—External hold data is not a part of the PTF. It resides in a separate file. On BrightStor product tapes, there is a HOLDDATA file. External hold data is usually used for sysmods that have been distributed and later are discovered to cause problems.

To take advantage of the external hold data, you must receive it into your SMP/E environment. If you use the jobs supplied by Computer Associates, SMP/E receives the hold data.

If a sysmod has an unresolved hold error, SMP/E does not install it unless you add a bypass to your apply command. You can bypass an error hold in situations that do not affect you. This is a problem that only happens with a hardware device that you do not have, or in a product feature that you do not use.

When Computer Associates issues the sysmod that resolves the hold, the resolving sysmod supersedes the hold error. This allows you to apply the original sysmod in conjunction with the fixing sysmod.

There is a special hold data class called ERREL. This means that Computer Associates has determined that the problem fixed by the sysmod is more important than the one that it causes. Computer Associates recommends that you apply these sysmods.

The easiest and most reliable way to manage external hold data is to allow SMP/E to manage it automatically. When you allow SMP/E to manage the process, the only manual task you need to do is run a REPORT ERRSYSMODS. This report identifies any held sysmods that you have already applied to your system. If the resolving sysmod is in receive status, SMP/E identifies the sysmod that you need to apply to correct the situation.

The Files Data Set

This section discusses the Files Data Set.

Create a Files Definition Member in Parmlib

A Files Data Set must be created to hold the backup index records. To create the Files Data Set, copy the sample definitions (member FDSAMPLE in parmlib) into a new member of parmlib named FILEDEFN. This new member is used to initialize your Files Data Set.

Note: If you are a current BrightStor CA-Disk for MVS customer, you must separate your MVS DSNINDEX records from your UNIX UNXINDEX records. You must not have these two subfiles within the same Files Data Set.

Initialize the Files Data Set

For information on initializing the Files Data Set, see the procedures in Formatting the Files Data Set in Appendix B “Files Data Set.”

The PROCLIB Data Set

This section discusses the PROCLIB data set.

Modify the following Cataloged Procedures

Read the following procedures and make any environmental changes needed to run in your environment:

- Use the DIMU Procedure when you want diagnostics.
- Use the DMSPOOLU Procedure when synchronizing tape pools with the Files Data Set.
- Use the DMSU Procedure when performing UNIX file backups and reporting on your UNIX file system.
- Use the FMSU Procedure when performing implicit UNIX file recoveries.
- Use the IXMAINTU Procedure when performing UNIX file index maintenance.
- Use the LISTU Procedure to list the index records of backed up UNIX files.
- Use the REBUILDU Procedure to recreate entries within the Files Data Set.
- Use the RELOADU Procedure to restore the Files Data Set backed up with the UNLOADU procedure.
- Use the RESTOREU Procedure when performing explicit UNIX file restores.
- Use the UNLOADU Procedure to backup the Files Data Set.

Running Tests

The following tests are listed in the recommended order because they provide exposure to all the supported areas. After these tests are run, variations of them can be run to test how well the software runs in your environment.

Considerations

When running the various functions, all file systems to be scanned must be mounted.

Test 1

The purpose of this initial test is to validate that DSCL Backup works properly. This test scans mounted file systems and select files for backup.

Note: For detailed explanations of the parameters shown in the example, see the chapter “Backup/Archive.”

- For directions on how to set up the backup media, see the procedures documented in Appendix C “Tape Management” and Appendix D “Tailoring Options.” Files can be backed up to disk, tape, or both.
- Create JCL to scan the mounted file systems in simulate mode. A sample is as follows:

```
//JOB CARD .....  
//BACKUP EXEC DMSU  
//SYSIN DD *  
SET MODE=SIMULATE  
SCAN UNIX  
SELECT PATH=/  
BACKUP
```

- Before you submit this job, make sure that the ID that is running this test has Superuser authority.
- After the job is run, select a directory to backup and run in live mode. A sample is as follows:

```
//JOB CARD .....  
//BACKUP EXEC DMSU  
//SYSIN DD *  
SCAN UNIX  
SELECT PATH=/bin\  
BACKUP
```

This example backs up all files in the /bin directory. It also backs up any directories that are below it.

- Run this job two or three times to generate different versions within the Files Data Set of the files backed up.

Test 2

This test validates that the LISTU command works properly. This test lists out the index records stored in the Files Data Set during the backup runs.

Note: For detailed explanations of the parameters shown in the example, see Reporting Commands in the chapter “Index Maintenance.”

- Run LISTU to list out all the index records. A sample is as follows:

```
//job.....
//LIST EXEC LISTU
//SYSIN DD *
LISTU FIELDS=(NAME28,FTYP,UONR,GONR,RFDT,MODT,FLSZ),PATH=/
/*
```

Test 3

This test validates that the restore process successfully finds the index record in the Files Data Set, mounts the archive tape or finds the archive file on disk, and then restores the file.

Note: For detailed explanations of the parameters shown in the example, see the chapter “Restore.”

- From the LISTU output, select a file to restore.
- Run a job to restore the selected file. Use this restore example as a guide:

```
//job
//RESTR EXEC RESTOREU
//SYSIN DD *
RESTORE PATH=/user/ispjx11/myfile.txt,
NEWPATH=/user/ispjx11/delta/myfile.txt,
DIRCREAT,SCRATCH,VERSION=-1
```

Test 4

This test validates that the IXMAINT process works properly and that index records are deleted appropriately after their expiration date has expired.

Note: For detailed explanations of the parameters shown in the example, see Maintenance Commands in the chapter “Index Maintenance.”

- Run a PATHDELETE job using the date parameter with a date set far enough in the future to cause the index records to expire. A LISTU output can be run to see what the expiration dates are. The job should be run in both simulate and then in live mode. An example is as follows:

```
//job.....
//MAINT EXEC IXMAINTU
//SYSIN DD *
PATHDELETE DATE=1999.180,SIMULATE
/*
```

Customizing ISPF Support

BrightStor CA-Disk for OS/390 UNIX includes load, panel, message, and skeleton files, which must be concatenated to the standard ISPF ddnames plus a special BrightStor CA-Disk for USS ddname. These files are:

Ddname	Contains	DSN
ISPLLIB	Load Modules	SAMSDISK.UNIX.SAMLOAD
ISPLLIB	Panels	SAMSDISK.UNIX.SAMPNL0
ISPLLIB	Messages	SAMSDISK.UNIX.SAMMSG0
ISPLLIB	Skeleton JCL	SAMSDISK.UNIX.SAMSKL0
UNXUNLIB	Load Modules	SAMSDISK.UNIX.SAMLOAD

UNXUNLIB is a special ddname, which must reference the BrightStor CA-Disk for OS/390 UNIX load library. ISPLLIB must point to a library that contains the load module UNXSP202. This library can contain only this one module, or it can contain all BrightStor CA-Disk for OS/390 UNIX load modules. Only UNXSP202 is loaded from ISPLLIB, which then instructs BrightStor CA-Disk for USS to load all its modules from the ddname UNXUNLIB.

BrightStor CA-Disk for OS/390 UNIX includes a REXX executable, named UNXSTART, which allocates all necessary libraries to use the ISPF function. This executable then calls the main BrightStor CA-Disk for OS/390 UNIX ISPF REXX program – UNXFD001. UNXSTART must be tailored by the installation to reference the correct data set names where the BrightStor CA-Disk for USS ISPF members have been installed. Alternatively, the installation can select to allocate these libraries in the TSO LOGON JCL or in a CLIST that runs prior to starting ISPF, and call UNXFD001 directly.

To run UNXSTART, the TSO session must have allocated either a SYSEXEC or SYSPROC DD, which references the BrightStor CA-Disk for OS/390 UNIX REXX library.

The BrightStor CA-Disk for OS/390 UNIX feature can be run from ISPF option 6 by entering UNXSTART (or UNXFD001 if all libraries are allocated before starting ISPF) as the command name.

Customizing ISPF

Following are the steps to customize ISPF:

1. Decide how you are going to invoke this ISPF support. There are three options:

- JCL
- CLIST
- REXX executable

2. Set up the concatenation through TSO logon JCL or through a CLIST or REXX executable. The following example assumes that the libraries are allocated through LIBDEFs in a REXX executable prior to calling the BrightStor CA-Disk for OS/390 ISPF function:

```
"ISPEXEC LIBDEF ISPLLIB DATASET ID('SAMSDISK.UNIX.SAMLOAD') STACK"  
"ISPEXEC LIBDEF ISPPLIB DATASET ID('SAMSDISK.UNIX.SAMPNLO') STACK"  
"ISPEXEC LIBDEF ISPMLIB DATASET ID('SAMSDISK.UNIX.SAMMSG0') STACK"  
"ISPEXEC LIBDEF ISPSLIB DATASET ID('SAMSDISK.UNIX.SAMSKLO') STACK"  
ADDRESS TSO "ALLOC FI(UNXUNLIB) DA('SAMSDISK.UNIX.SAMLOAD') SHR REUSE"
```

3. Allocate either a SYSEXEC or SYSPROC DD that references the BrightStor CA-Disk for OS/390 UNIX REXX library, as in the following example:

```
ADDRESS TSO "ALLOC FI(SYSEXEC) DA('SAMSDISK.UNIX.SAMEXEC') SHR REUSE"
```

4. To include the ISPF feature directly onto the ISPF main menu, add the following line to the IBM startup panel:

```
12, 'CMD(UNXSTART)'
```

or

```
12, 'CMD(UNXFD001)'
```


Files Data Set

BrightStor CA-Disk for USS uses a direct access data set called the Files Data Set to store information about the files it operates on. It is a single volume data set that must be in one extent. The Files Data Set contains multiple subfiles of data. The term “subfile” refers to a group of logically related records kept within the Files Data Set. Each subfile is assigned an eight-character name by which it is identified. Subfiles are independent groups of records that do not necessarily have any relationship to records in other subfiles of the data set. They are processed as if they were separate data sets. This technique provides the convenience of having multiple DASD files without the task of having to support each one independently.

Note: If you are a current BrightStor CA-Disk for MVS customer, you must separate your MVS DSNINDEX records from your UNIX UNXINDEX records. You must not have these two subfiles within the same Files Data Set.

An EXCP I/O interface is used to retrieve, insert, update, and delete records. Records are referenced by their specific keys within each subfile, but can also be retrieved sequentially by subfile name alone.

For example, one subfile used by BrightStor CA-Disk for USS is the file index record subfile (UNXINDEX). This subfile contains one record for each file currently in the BrightStor CA-Disk for USS archives. When a file is archived, a record is placed in the UNXINDEX subfile. When the file is restored, the UNXINDEX subfile is accessed to determine the volume serial of the tape or disk containing the archived version of the file and its location within that volume. These records remain in the subfile until they are deleted by index maintenance.

Subfiles of the Files Data Set

Following is a list of the subfiles that are associated with the Files Data Set:

Parameter	Description
ARCHVOLS	A record exists in this file for each archive volume (tape or disk) to which one or more files were archived. An entry is also present for each backup copy of the archive volumes.
UNXINDEX	This file contains an entry for each file backed up or archived by the system. The records are keyed by file name, and multiple versions of a given file can be present.
UNXPATHS	An entry for each file placed in the UNXINDEX subfile generates an entry into this subfile. The contents of these entries consist of a variable path name related to the file, and all of its attributes.

Restrictions

A maximum of 12 subfiles is presently supported. However, you can limit this even further by specifying `sysparm FILESCNT`. For more information, see the `sysparm` description for `FILESCNT12`.

Subfile Entry Definition

All subfiles to be used are defined to BrightStor CA-Disk for USS by entries in the `FILEDEFN` member of `parmlib`. During files initialization, the member entries are read and used to determine how the data set is to be formatted.

A sample set of subfile definition entries is distributed with the system as member `FDSAMPLE` in `parmlib`. To activate the definitions, you must create a member named `FILEDEFN` and copy in the sample values. An entry is required in the `FILEDEFN` member for each subfile of the Files Data Set used.

With the exception of the `UNXINDEX` subfile, the capacity values supplied in `FDSAMPLE` should be sufficient for a substantial period of time. All of the fields within `FILEDEFN` are described below.

Subfile Definition Parameters

The subfile definition parameters discussed here are found in `dsn=dms.parmlib(FILEDEFN)`. When you are defining subfiles, remember that the maximum number of subfiles that can be defined is 12. Following is a description of the FILEDEFN record format:

Position	Field Name	Description
1-8	Subfile Name	The 8-character name of the subfile being defined.
9-11	Logical Record Length	The 3-digit length of the records that constitute the subfile.
12-14	Key Length	The 3-digit length of the record keys.
15		Character "C" to indicate capacity field follows.
16-23	Capacity (maximum record count) needed	The number of records the subfile must be able to index. This number is the theoretical maximum allowed; for example, if inserts were in the perfect sequence to fill every data block – in other words, when totally "compressed", this number of records can be indexed. In the real world, inserts are random, resulting in "block splits" which cause embedded free space (as much as 50 percent) in any given data block (or possibly all blocks). For this reason, it is recommended that you set the capacity value to roughly twice the number of records you estimate are inserted.
24	Shared DASD	The letter Y in this column indicates that the file can be accessed concurrently by multiple tasks.
25	No Duplicates Allowed	The letter "Y" in this column indicates that the file does not permit entries with duplicate keys.
26	Deferred Write	The letter "Y" in this column indicates that the nature of the processing of this file does not require that each record added be written back immediately. This flag should be specified as "Y" only for file name DASDSPCB. All others should be specified as "N."
27	File validate	The letter "Y" in this column indicates that information regarding the file structure is to be validated during file reload.

Files Data Set Management

This section discusses Files Data Set management.

Formatting the Files Data Set

After the FILEDEFN members are created (see Subfile Entry Definition in this chapter), the Files Data Set can be created and formatted with the following FILEINIT. During file initialization, the FILEDEFN member entries are read and used to determine how the Files Data Set is to be formatted. These FILEDEFN member entries are accessed only during file initialization, after which all information is carried within the data set itself. A Files Data Set status report (see Files Data Set Status Report in this chapter) is created upon completion of the FILEINIT JCL.

```
//FILEINIT EXEC PGM=ADSMI002, PARM=ADSDM100, REGION=512K
//STEPLIB DD DISP=SHR, DSN=SAMS.UNIX.LOADLIB
//ABNLDUMP DD DUMMY
//FILES DD DISP=(,CATLG,DELETE), DSN=SAMS.UNIX.FILES,
// UNIT=SYSDA, SPACE=(CYL,10,,CONTIG),
// DCB=(DSORG=DA)
//MSGPRINT DD SYSOUT=A
//PARMLIB DD DISP=SHR, DSN=SAMS.UNIX.PARMLIB
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
```

The subfiles are formatted for optimum performance based upon the physical block size of the Files Data Set. The physical block size of the Files Data Set is the only value common to all of its subfiles. BrightStor CA-Disk for USS automatically assigns the optimum block size according to the device type being used. You can override the default BLKSIZE by specifying a value in your JCL, but it is **not** recommended. The major concern is to specify an appropriate value in the FILEDEFN entries.

Determining the Allocation Size of the Files Data Set

To determine the approximate size required for the Files Data Set, you must first have an idea as to what is stored within the Files Data Set. This is important because the size is affected not only by the number of UNXINDEX records that are stored but also by the number of UNXPATHS that are stored and the number of ARCHVOLS that are needed.

The path records in the UNXPATHS subfile are unique in that only one copy is ever stored for each path. Growth then is rapid in the beginning but slows down quickly as a record for all paths is recorded.

The file records in the UNXINDEX subfile can have multiple copies and so version requirements need to be considered.

One block can hold 37 UNXPATHS records (Logical Block length = 55,500). With half-track blocking on a 3390 device, it takes one track to hold one block.

One block can hold 653 UNXINDEX records (Logical Block length = 391,800). With half-track blocking on a 3390 device, it takes seven tracks to hold one block.

One block can hold 254 ARCHVOLS records (Logical Block length = 27,940). With half-track blocking on a 3390 device, it takes one track to hold two blocks.

For example, if 1,000 ARCHVOLS are used to process 50,000 directories and one million records, then the capacity is as follows:

Archvols: (1,000 / 254)	= 2 tracks	= 1 Cylinder
Directories: (50,000 / 37)	= 1,352 Tracks	= 91 Cylinders (3390)
Files: (1,000,000 / 653) x 7	= 10,720 Tracks	= 715 Cylinders (3390)

 Total allocation needed is: 12,074 Tracks or 807 Cylinders (3390)

Files Data Set Status Report

The following JCL can be used to produce the Files Data Set status report:

```
//STATUS EXEC PGM=ADSMI002, PARM=ADSDM107, REGION=512K
//STEPLIB DD DISP=SHR, DSN=SAMS . UNIX . LOADLIB
//ABNLDUMP DD DUMMY
//FILES DD DISP=SHR, DSN=SAMS . UNIX . FILES
//MSGPRINT DD SYSOUT=A
//PARMLIB DD DISP=SHR, DSN=SAMS . UNIX . PARMLIB
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
```

This JCL produces a report that indicates the current usage for each subfile defined to the Files Data Set. A critical field that should be inspected regularly is the OVERFLOW BLOCK COUNT. Whenever overflow blocks exist, the definition entry for each subfile with overflow blocks must have its record capacity increased. Failure to do so results in performance degradation and unpredictable results.

```

File Edit Transfer Options Connection Macro Window Help
-----
Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY ISPRLN3F JOB05882 DSID 105 LINE 74 COLUMNS 02- 133
COMMAND INPUT ==> SCROLL ==> CSR

                                COPYRIGHT 1992, 1994
                                S T E R L I N G   S O F T W A R E   ,   I N C .
                                SAMS-DISK DATA STORAGE MANAGEMENT SYSTEM IS A
                                SOFTWARE INC.. USE OF THIS SOFTWARE BY UNAUTHORIZED PERSONS IS PROHIBITED.

                                JOB=ISPRLN3F                                FILES DSN=ISPRLN3.UNIX.FILES2
                                STEP=FDSINT                                PARMLIB DSN=ISPJXL5.UNIXTEST.PARMLIB
                                                                DSN=DISK920.BUILD2.PARMLIB
                                1998.238 AUG 26, 1998                                WEDNESDAY 12.45 PM

1998.238 AUG 26, 1998                                F I L E S   D A T A   S E T   S T A T U S                                PAGE 5
WEDNESDAY 12.45 PM                                SAMS:Disk - OS/390 UNIX 1.1.0
DATA SET NAME ISPRLN3.UNIX.FILES2
PHYSICAL BLOCK SIZE 32768                                TOTAL BLOCKS 75
BLOCKS IN USE 25                                BLOCKS REMAINING 50
-----OPTION FLAGS-----
SUB-FILE RECORD KEY RECORD MAXIMUM BLOCK REC PER BLOCK COUNT OVERFLOW
NAME LENGTH LENGTH COUNT RECORDS LENGTH BLOCK BLOCKS COUNT BLOCKS SHARE NODUP DEFER VALID GEN
-----
UNXPATMS 1500 8 0 234694 64500 43 5458 1 0 Y Y Y
UNXINDEX 600 256 0 548924 261600 436 1259 1 0 Y Y Y
ARCHIOLS 110 6 0 972081 32670 297 3273 1 0 Y Y Y
***** BOTTOM OF DATA *****
Aa B TCP/IP R 4 C 21 14:34 8/26/98

```

Field Descriptions—Status Report

Following is a list of the fields on the status report and their description:

Field	Description
BrightStor CA-Disk for USS v.r.m	The release number appears where v.r.m is shown.
Physical block size	The actual block size for the Files Data Set. BrightStor CA-Disk for USS automatically assigns an optimum value for the device unless you specify a BLKSIZE value in the initialization JCL.
Total blocks	The total number of physical blocks within the Files Data Set. Some of these contain data (blocks in use), and others are “free blocks”, available for record inserts (blocks remaining).
Blocks in use	When initialized, the Files Data Set is formatted from beginning to end with physical blocks of binary zeros (see Total blocks). A minimum of one index block and one data block are then “allocated” to each subfile. All other blocks are “free” (see Blocks remaining), and are subsequently allocated only when record inserts require more data blocks. The “blocks in use” number is the total number of physical blocks that have been allocated due to record inserts.

Field	Description
Blocks remaining	The number of physical blocks initially formatted with binary zeros that have not yet had any records inserted. The number of unused, or free blocks in the data set.
Subfile name	The name of either a BrightStor CA-Disk for USS or user-defined subfile.
Record length	The length of the logical record within the named subfile.
Key length	The length of the key portion within the logical record. For example, the key to the UNXINDEX record is the data set name, which is 44 bytes of the 256-byte record.
Record count	The current number of records contained within the subfile.
Maximum records	The maximum number of records that can be inserted and still indexed. This value is computed at initialization time from the capacity fields you specified in the FILEDEFN member of parmlib. When the actual record count exceeds half of this maximum count, you should consider specifying a higher capacity value in the FILEDEFN entry.
Block length	Records for each subfile are grouped together to form a "logical" block that is also a "best fit" into a physical block (or multiples of physical blocks). Therefore the value displayed should be slightly less than a multiple of the physical block size.
Rec per block	The number of records grouped together to form a "logical" block for the named subfile.
Max blocks	The maximum number of logical data blocks that can be indexed for the subfile. Multiplying max blocks times rec per block results in the maximum number of records that can be inserted and still indexed.

Field	Description
Block count	The number of logical data blocks in use (indexed) for the subfile.
Overflow blocks	When an additional logical data block is allocated to a subfile but there is no room left in the index to keep track of it, it is called an overflow block. Overflow blocks are chained from the relevant data blocks, and cause considerable overhead or other unpredictable errors. The capacity for the subfile should be increased.

Option flags

Following is a description of the option flags and their meanings:

Option Flag	Meaning
Share: Y	Indicates that integrity is maintained during concurrent access by multiple tasks.
Nodup: Y	Indicates that entries with duplicate keys are not allowed.
Defer: Y	Indicates that I/O can be buffered. This option requires special processing and is supported for only the DASDSPCB subfile.
Valid: Y	Activates internal validation routines when deemed appropriate.

Files Data Set Utilities

Following is a description of the Files Data Set utilities:

UNLOAD and RELOAD

Two utility functions are provided to backup and restore the Files Data Set: UNLOAD and RELOAD.

- UNLOAD – unloads the Files Data Set to a sequential file to create a backup.
- RELOAD – reloads the Files Data Set from the sequential file.

Selection of a subset of the Files Data Set is permissible during both the unload and reload operations. Caution should be exercised, however, when specific subfiles are selected since the **entire** Files Data Set is **reformatted every time** a RELOAD is done, that is, a RELOAD does not insert records at the logical subfile level leaving the other subfiles intact. All subfiles are reformatted and then the unloaded records are reloaded. Be careful to ensure that all subfiles with records to be saved are both unloaded and reloaded.

The unload/reload support is normally used for one of two purposes. One role is to permit the allocation of a larger Files Data Set to accommodate file growth. The new file capacities must be entered in the FILEDEFN member prior to executing the reload program. Since the reload program reformats the entire Files Data Set, destroying the contents of all subfiles, every subfile must first be unloaded.

The other role is to re-order the data blocks and reduce block splits. The Files Data Set itself can be large enough, but growth in one or more subfiles causes an increasing number of block splits that increases the number of I/Os performed.

UNLOAD

This section describes the UNLOAD utility.

JCL — UNLOAD Files

When you UNLOAD ALL subfiles, BrightStor CA-Disk for USS dynamically allocates the sequential files into which the files are unloaded. If it is present in the JCL, the SEQFILES DD is used. If you let BrightStor CA-Disk for USS dynamically allocate this file, your unloaded copy of the Files Data Set is called "files.dsname.current-date.current-time." If the Files Data Set name exceeds the allowable amount of characters for the date appendage, truncation of the Files Data Set name takes place at position 27 of the Files Data Set name. Note that the enqueue on the Files Data Set is held for the entire UNLOAD process. When the sequential file is dynamically allocated, the expiration date used for the file is taken from sysparm DYNEXPDT.

The SIMULATE parameter is provided with the UNLOAD function of BrightStor CA-Disk for USS. When specified, all reports and messages are generated, but the actual UNLOAD does not take place until the SIMULATE parameter is removed from the command.

Unloading the contents of the Files Data Set from disk to tape (a sequential file on disk works equally well) is accomplished with the following JCL. A DSN=any.gdg(+1) is suggested.

```

EDIT                                (UNLOAD) - 01.01                                Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000100 //XXXXXXXX JOB (XXXX,XXX), 'NNNNNNNN', CLASS=X, MSGCLASS=T, NOTIFY=&USERID
000200 //UNLD EXEC PGM=ADSMI000, PARM=ADSDM177, REGION=4096K
000300 //STEPLIB DD DISP=SHR, DSN=SAMS.UNIX.LOADLIB
000400 //ABNDUMP DD DUMMY
000500 //CHDPRINT DD SYSOUT=*
000600 //FILES DD DISP=SHR, DSN=SAMS.UNIX.FILES
000700 //MSGPRINT DD SYSOUT=*
000800 //PARMLIB DD DISP=SHR, DSN=SAMS.UNIX.PARMLIB
000900 // DD DISP=SHR, DSN=WORKSHOP.R11.PARMLIB
001000 //SYSPRINT DD SYSOUT=*
001100 //SYSUDUMP DD SYSOUT=*
001200 //SEQFILES DD DISP=(,CATLG,DELETE),
001300 // DSN=SAMS.DISK.SEQFILES,
001400 // DCB=(LRECL=1512, BLKSIZE=6144, DSORG=PS, RECFM=UB),
001500 // SPACE=(6144, (360,360), RLSE), UNIT=SYSALLDA
001600 //SYSIN DD *
001700 UNLOAD ALL
001800 //
***** ***** Bottom of Data *****
Aa A TCPIP R 20 C 9 10:08 9/25/98

```

UNLOAD Command and Parameters

Following is an example of the UNLOAD command:

```
UNLOAD ALL, FILES=, SIM, UNIT=, PRI=, SEC=
```

Following is a list of parameters that are associated with the UNLOAD command:

Parameter	Description
ALL	The presence of this parameter causes all subfiles to be unloaded.
FILES=	A list of one or more subfile names can be specified to confine the output to a subset of the Files Data Set. Omission of this optional parameter results in all files being unloaded.
SIM	This parameter causes the output reports to be produced as if processing had been performed; however no files are unloaded or deleted.
UNIT=	This is a list of names for the units allocated for the UNLOAD file when the unload is considered a backup (ALL is specified and the DAYS= parameter is used). If the unit is a disk device, use an esoteric name. If the unit is a tape device, enter TAPE, 3480, DYN1, DYN2 or DYN3, and the DYNxUNIT sysparm values are used.

Parameter	Description
PRI=	This is the primary space allocation amount in cylinders for the UNLOAD file. The default amount is one cylinder.
SEC=	This is the secondary space allocation amount in cylinders for the UNLOAD file. The default amount is one cylinder.

RELOAD

This section describes the RELOAD utility.

JCL — RELOAD Files

The RELOADING of an unloaded set of files to the Files Data Set is accomplished with the following JCL. Care should be taken to ensure that the sequential input file contains all of the needed files, since the reload reformats the entire Files Data Set prior to doing the reload.

```

EDIT          SBDLM.UTIL.CNTL (RELOAD) - 01.01          Columns 00001 00072
Command ==>                                         Scroll ==> CSR
***** ***** Top of Data *****
000100 //XXXXXXX JOB (XXXX,XXX), 'NNNNNNNN',
000200 //          CLASS=X,
000300 //          MSGCLASS=T,
000400 //          NOTIFY=&USERID
000500 //RELD      EXEC PGM=ADSMI002,PARM=ADSDM192,REGION=4096K
000600 //STEPLIB DD DISP=SHR,DSN=SAMS.UNIX.LOADLIB
000700 //ABNDUMP DD DUMMY
000800 //CMDPRINT DD SYSOUT=*
000900 //FILES DD DISP=SHR,DSN=SAMS.UNIX.FILES
001000 //MSGPRINT DD SYSOUT=*
001100 //PARMLIB DD DISP=SHR,DSN=SAMS.UNIX.PARMLIB
001200 //          DD DISP=SHR,DSN=WORKSHOP.R11.PARMLIB
001300 //SYSPRINT DD SYSOUT=*
001400 //SYSUDUMP DD SYSOUT=*
001500 //SEQFILES DD DISP=OLD,DSN=SAMS.UNIX.SEQFILES
001600 //SYSIN DD *
001700 RELOAD ALL,FORMAT
001800 /*
***** ***** Bottom of Data *****
Aa A TCPIP R 21 C 9 10:29 9/25/98

```

RELOAD Command and Parameters

Following is an example of the RELOAD command:

```
RELOAD ALL,FORMAT,MODULES=,FILES=,FREESP=
```

Following is a list of parameters that are associated with the RELOAD command:

Parameter	Description
ALL	This parameter reloads all of the unloaded records.
FILES=	A list of one or more subfile names can be specified to confine the load to a subset of the records on the sequential storage medium. Omission of this parameter results in all records from the sequential file being loaded, regardless of the subfile from which they came.
FORMAT	This required parameter causes the Files Data Set to be entirely reformatted prior to loading the records from the specified subfiles.
MODULES=	A list of one or more names of modules to be given control in order to inspect or alter each record reloaded.
FREESP=	By default, RELOAD processing reloads subfile records into each logical data block, until the maximum record limit is reached. After which a BLOCK SPLIT is required to insert other entries into the logical bock. Specifying this parameter with a percentage value helps alleviate block splits. This parameter is limited to only the UNXINDEX subfile. For example, a value of 20 allows 20 percent of free space for each logical data block.

Tape Management

This section describes the following topics regarding the management of backup tapes:

- Allocation Considerations
- Ddnames Used
- Producing Duplicate Archive Tapes Concurrently
- Cataloging the Tape Archives
- Assigning Tape Expiration Dates
- Tapepool Considerations
- Customizing Tape Management Support

Additional information related to managing tapes can be found in the following section:

- VOLDELETE

Allocation Considerations

This section discusses allocation considerations for backup tapes.

Ddnames Used

BrightStor CA-Disk for USS uses the following ddnames for tape and cartridge units:

- //ARCHIVE0—creates the primary archive tape/cartridge
- //ARCHIVEC—creates the duplicate copy tape/cartridge
- //ARCHIVEN—creates the primary merge output tape/cartridge
- //ARCHIVCn—creates the duplicate copy merge tape/cartridge
- //ARCHIVES—reads an archive 3420 tape (that is, restore or merge)
- //ARCHIVER—reads an archive cartridge (that is, restore or merge)

Producing Duplicate Archive Tapes Concurrently

When you run archive or backup jobs, two copies of archive volumes are produced concurrently unless you nullify the allocation of the `//ARCHIVEC DD` statement. To suppress the duplicate copy, specify `sysparm ARCCTYPE` with a value of `NULL`, or provide the `DD` statement in your JCL as `//ARCHIVEC DD DUMMY`.

The volume serial number of a duplicate copy tape is carried in the archive volume record for the primary tape. The duplicate copy tape also has an archive volume record built for it. If another copy is made (the triplicate copy), the volume record for the duplicate points to the triplicate. In each case, the volume record points to the copy record, which points to the next copy, and so on until the end of the chain is reached, indicated by blanks in the volume serial field for the next copy tape.

In the event that a primary archive tape is damaged or otherwise becomes unavailable, the `RESET` command can be used to flag it as disabled, and cause subsequent `RESTORE` or `MERGE` requests to automatically call for the duplicate copy backup tape.

If you produce an archive volume without a duplicate copy, it is possible to create a copy after the fact, just as you can produce the triplicate from the duplicate. For more information, see *Making Additional Copies of Archive Tapes* in the chapter “Index Maintenance.”

Cataloging the Tape Archives

As previously described, if you want to catalog your archive tapes, specify a value of `C` for `sysparm ARCTNAME`. If you also use a security system, see the authorization requirement documented in `ARCTNAMEd`.

To uncatalog tapes that are expired by `MERGE` or `IXMAINT`, specify `sysparm UNCATARC` with a value of `Y`.

Assigning Tape Expiration Dates

Expiration dates can be assigned to archive tapes either explicitly or implicitly. An explicit expiration date is created by hard-coding its value in `sysparm DYNEXPDT` for dynamic allocation or by specifying the `EXPDT=` or `RETPD=` parameter in the JCL for the `//ARCHIVE0 DD` statement.

BrightStor CA-Disk for USS honors any non-zero expiration date and places it in both the tape label and in the archive volume index record. Hard-coding an expiration date, however, does create the possibility of expiring archive tape volumes prior to the expiration of all archived files on the archive volume.

Computer Associates recommends using an explicit expiration date of “99365” on your archive tapes. This is considered a “never scratch” date by BrightStor CA-Disk for USS and protects against the previously mentioned case of a tape being written over before all of its files have expired. This is the technique called for by the distributed JCL procedures and system parameter default. For related information, see Year 2000 Considerations in the chapter “General Information.”

If you have implemented tape management support, it is recommended that all archive tape DD statements use the default EXPDT=99365. BrightStor CA-Disk for USS notifies the tape management interface as soon as any given tape can be returned to scratch status. For more information, see Customizing Tape Management Support in this chapter.

Tapepool Considerations

The BrightStor CA-Disk for USS tapepool support provides you with the ability to define up to 30 different pools of tapes. You can define 30 pools in parmlib member POOLDEFS. However, this member is also used to define disk archive pools. The sum of tape and disk pool definitions cannot exceed 30. Each pool name is associated with (mapped to) a data set name on an archive DD statement. Thus, if all archive tapes have the same name or prefix, one pool is sufficient. However, if the duplicate copy tapes have different names than the primary tapes (as in the default), it is possible to define a separate pool of tapes for each. Similarly, backup tapes can be separated from archive tapes by giving them different names and associated pools.

The association (or mapping) of pool names to archive data set names is made by entries in member POOLDEFS in the parmlib data set. Each entry consists of an eight-character pool name (of your choice) and the tape data set name. Remember, the data set names are generated from sysparms ARC0NAME and ARCCNAME or are provided in the JCL. Each entry must be enclosed in quotes, with one or more blanks between the pool name and data set name. For example:

'ARCHPRIM DMS.ARCHPRIM/'	pool for primary archives
'ARCHCOPY DMS.ARCHCOPY/'	pool for duplicate archives
	or perhaps
'ARCHPOOL DMS.ARCH/'	pool for all archive tapes
'BKUPPOOL DMS.BACK/'	pool for all backup tapes

The first step in establishing tapepool support is to decide how many different data set names (categories of tapes) you have a real need for. One suggestion is to separate your archive from your backup tapes, and if you create duplicate copy tapes, to separate them from the primary tapes as well. After deciding this, make sure that the names you selected are provided to BrightStor CA-Disk for USS through overriding sysparms for each job, or in the JCL for each job. Then create member POOLDEFS in your parmlib data set with the appropriate entries. You can copy in member SAMPDEFS as a starting point.

The next step is to place scratch tapes into the pools in preparation for use.

At this point you are ready to activate the support. This is done by adding sysparm TAPEPOOLy to member SYSPARMS in the parmlib data set. After this is done, new scratch tapes for archive, backup, or merge functions are located in the appropriate pool, and their status is updated to show that they are in-use. When subsequent merge or index maintenance functions determine that no unexpired files remain on a tape volume, its status is updated to indicate that it is available again for use as a scratch tape.

When a group of tapes are entered into a tapepool, they are entered at the bottom of the current list of scratch tapes for the named pool. Tapes that are scratched by MERGE or IXMAINT processing are also put at the bottom of the tapepool scratch list, in the order in which they are scratched. Scratch tapes needed by ARCHIVE, RETAIN, or MERGE are taken from the top of the scratch list. This ensures that the tapes are cycled, much like tape drives are cycled. Due to the complex logic that MERGE and IXMAINT use to scratch BrightStor CA-Disk for USS archive tapes, the tapepool scratch list quickly loses its initial VOLSER order. This causes no problems for BrightStor CA-Disk for USS, however. The tapepool LIST command described below can be used to inspect the status of the tapepools.

Tapepool support is also designed to maintain the integrity of the pools in the following manner:

- The same volume cannot be placed into more than one pool.
- Only the ISPF or batch utility program adds a volume into a pool; that is, scratch tapes mounted by operators or expired tapes being “returned to scratch status” by index maintenance are never added to a pool. The volume must already exist in the pool such that BrightStor CA-Disk for USS merely updates its status.
- When creating new output tapes with the pool support active, a valid pool name to use must be found or processing terminates.

Tape volumes can be supplied by any combination of the following three techniques. It should be noted that these methods are progressive, in that processing begins with the first that is applicable and continues until either no more tapes are needed or the operator is unable to supply an acceptable volume serial.

From JCL

If tapepool support is being used, from one to five volumes can be listed in the VOL=SER= parameter of the //ARCHIVE0 and //ARCHIVEC DD statements. These volumes are used before accessing the defined tapepool.

For non-tapepool support, up to five volumes can be supplied on the VOL=SER= parameter of the //ARCHIVE0 and //ARCHIVEC DD statements. If more than five volumes are required, the operator is requested to mount scratch tapes.

From a Tapepool

If the tapepool support is being used, volume serial entries are taken from the pool when the VOL=SER= parameter is not specified in the JCL, or when the JCL list of volume serials has been exhausted.

From the Operators

If the tapepool support is being used but the pool is out of scratch tapes, a WTOR message is issued to allow the console operator to enter the volume serial of the next scratch tape to be used.

It is permissible to cancel an archive run when a volume serial cannot be supplied in response to the WTOR message, since the processing for any volume written up to that point is complete.

Customizing Tape Management Support

Tapes have traditionally been managed by the expiration dates written in the tape labels. Standard IBM support for date-protected files requires operator intervention to rewrite a protected tape before its expiration date. Tape management systems usually extend this support in several ways. They interpret certain expiration dates as codes to indicate the type of control governing the use of the tape. A master control file, or "tape management catalog," records this information, and controls access to each tape. If the control file indicates a tape as an available scratch tape, it allows it to be rewritten without operator intervention, regardless of the expiration date in the label.

For example, in BrightStor CA-1 Tape Management (BrightStor CA-1), Computer Associates uses an EXPDT=99000 to mean that a tape is eligible as a scratch tape when it becomes uncataloged. As long as the tape is cataloged, it is protected from being used as a scratch tape. This is commonly known as “catalog control.” An EXPDT=99365 is often used as the code to mean “permanently protected” – the tape cannot be reused as a scratch tape unless the control file is updated to change this status. If you are using LDATE/ddd (to retain ddd days after the tape was last used), be aware that BrightStor CA-1 assigns an expiration date of 98ddd that might be confusing since it shows up as 1998ddd in some of our reports. Consult the documentation for your tape management system to find the various means of control it provides through expiration date codes.

With the introduction of MVS/XA 2.2 (DFP 2.3.0), IBM also recognizes 99365 as a “never expire” date. This means that for standard applications, operator intervention is required to rewrite such tapes even in year 2000 and beyond. Beginning with DFP 1.1, however, IBM has provided an exit to allow operator intervention to be avoided. BrightStor CA-Disk for USS optionally makes use of this exit, based upon the value of sysparm TAPEFSCR.

BrightStor CA-Disk for USS uses standard techniques to open and write tape files, which causes tape labels and their expiration dates to be created by normal means. As distributed, BrightStor CA-Disk for USS dynamically allocates all tapes that are needed and assigns them an expiration date of Julian 99365 (from the default value of sysparm DYNEXPDT). This value guarantees the integrity of the data by ensuring that a tape never expires before the data that it contains. BrightStor CA-Disk for USS determines when all of the data on each tape has expired, and at that point returns it as an eligible scratch tape.

BrightStor CA-Disk for USS also provides an option to catalog each tape data set that it creates. By specifying expiration dates and catalog options correctly, BrightStor CA-Disk for USS is fully compatible with all major tape management systems. As mentioned above, sysparm DYNEXPDT controls the expiration date for dynamically allocated tapes. If you supply DD statements for the output tapes, the expiration date you supply in the JCL is used.

For a description of several implementation options relating to tape management in general, see Tapepool Considerations in this chapter. BrightStor CA-Disk for USS provides three methods for controlling tapes. Select one of these methods and follow the instructions for installing the tape management support. For related information, see Year 2000 Considerations in the chapter “General Information.”

Method 1—Controlling Tapes Through the EDM

Beginning with Release 4.8 of BrightStor CA-1, a facility was provided (External Data Manager (EDM) interface) to allow other system software products to manage their own tapes. BrightStor CA-Disk for USS is one such system. It is through this interface that the greatest amount of control with the least amount of risk is available. Review the BrightStor CA-1 manuals for a complete description of the EDM interface. The advantages of using this method to control BrightStor CA-Disk for USS-created tapes over any other methods are as follows:

- Activation of the EDM interface for BrightStor CA-Disk for USS is greatly simplified.
- Changes within BrightStor CA-1 or BrightStor CA-Disk for USS do not require the reinstallation of the interface.
- A single point of control is established to determine when a tape should be scratched.
- BrightStor CA-1 does not attempt to prematurely scratch a BrightStor CA-Disk for USS-owned tape.
- BrightStor CA-1 does not allow other programs to overwrite a BrightStor CA-Disk for USS-owned tape.
- If BrightStor CA-Disk for USS creates a tape and then later abends, the tape is not scratched, as it normally would be without the EDM facility.
- Future changes in BrightStor CA-Disk for USS or BrightStor CA-1 do not increase the likelihood of tapes being scratched prematurely.

When BrightStor CA-Disk for USS is identified as an EDM to BrightStor CA-1, BrightStor CA-Disk for USS manages any archive/backup tapes that are created by BrightStor CA-Disk for USS (that is, BrightStor CA-Disk for USS informs BrightStor CA-1 when to scratch the tape). BrightStor CA-1 exempts these tapes from its normal processing. Any tapes created through sequential migrate – or UNLOAD to tape – processing is not considered externally managed, however, and are scratched by BrightStor CA-1 based on the expiration date of the tape.

A tape can be identified as externally managed by its expiration date (99365) and with an indicator flag within the TMC record of the volume. The actual expiration date of the tape is kept in the ARCHVOLS record within the Files Data Set. Normally this is 1999365, if the default value of sysparm DYNEXPDT is used. If this is so, the date is scratched when the last file on the tape expires. When BrightStor CA-Disk for USS determines that all data on the tape has expired, it expires the tape through the EDM interface. Directly changing the expiration date for a tape must be done through the appropriate BrightStor CA-Disk for USS commands, not through BrightStor CA-1. Only when a tape is expired by BrightStor CA-Disk for USS should the status within the TMC change.

BrightStor CA-1 Releases 4.8 and 4.9 allow for only one EDM. If you are unable to identify BrightStor CA-Disk for USS as an EDM because you already have another system identified, consider using Method 2 described below. The default expiration date for all tapes should be used, however, since there are several limitations in using the direct interface recommended in Method 2.

The activation of EDM support is done in two parts:

1. Sysparm TMSCTLEX must be set to indicate to use the BrightStor CA-Disk for USS EDM program as the tape management interface. Set this sysparm to ADSTH014.
2. BrightStor CA-Disk for USS must be identified as the EDM within BrightStor CA-1. This is accomplished by modifying the TMOEDMxx member of your CALPPOPTION data set. The easiest method of identifying BrightStor CA-Disk for USS as an EDM is by program name. For example:

```
EDM=SAMS, PGM=ADSMI002
```

For more information on the External Data Manager Interface, see your BrightStor CA-1 manuals.

If you previously created archive/backup tapes with BrightStor CA-Disk for USS, you also need to change their status within the TMC to indicate that they are also externally managed. This action is required because BrightStor CA-Disk for USS attempts to scratch the tapes through the EDM interface as soon as sysparm TMSCTLEX is set.

Converting existing tapes to EDM control can be done using documented BrightStor CA-1 utilities. The volume record for each tape must have the following fields set:

Volume Record	Value
1STVOL	hex zeros
NEXTVOL	hex zeros
PREVVOL	hex zeros
FLAG3	hex zeros
VOLSEQ	hex '0001'
EXPDT	99365

Note: If the conversion of existing tapes is not done properly, BrightStor CA-1 begins issuing TMSTVEXT-08 messages. You can circumvent this problem by executing the TMSUPDTE utility, changing flag '3' to '20'. For more information about this utility, see your BrightStor CA-1 manuals.

More detailed procedures can be obtained from BrightStor CA-1 Technical Support for the release of BrightStor CA-1 you are running. More general procedures can be obtained from the BrightStor CA-Disk for USS Technical Support Center.

Other BrightStor CA-Disk for USS sysparms that control the data set name, expiration date, and catalog action of BrightStor CA-Disk for USS tapes are set as described below. The expiration date on the internal label of an externally managed tape is not affected by the EDM facility.

Method 2—Controlling Tapes by Expiration Date

As distributed, BrightStor CA-Disk for USS defaults to assigning expiration dates of 99365 to all tapes, but does not catalog them. This technique is intended for those installations that do not have a tape management system, and as an option for those using earlier releases of BrightStor CA-1. When BrightStor CA-Disk for USS determines that all data on the tape has expired, it is released (through a direct interface) and made eligible as a scratch tape.

Note: If BrightStor CA-1 Release 5.0 or above is installed, do not use the BrightStor CA-Disk for USS/BrightStor CA-1 direct interface. BrightStor CA-1 macro changes have made this interface incompatible.

Whether you also catalog the tapes that are actually under expiration date control is a choice you should make, based upon whether it provides you with any additional benefit. Neither BrightStor CA-Disk for USS nor BrightStor CA-1 uses the catalog status, but it does allow for a simple list of the catalog entries as a means to find the BrightStor CA-Disk for USS tapes.

To have BrightStor CA-Disk for USS catalog the tapes it creates, do not specify `DISP=(NEW,CATLG)` in the JCL. Instead, specify sysparm `ARCTNAME` with an appended value of `C`. BrightStor CA-Disk for USS generates a unique name for each tape it creates, and then catalogs it. Specify sysparm `UNCATARC` with a value of `Y`, to have BrightStor CA-Disk for USS help keep your system catalog clean by uncataloging each tape when it returns it to the scratch pool. Review the use of sysparm `TAPEFSCR` with a value of `Y`, as well. BrightStor CA-1 users should not specify it, but it is recommended for all other users.

For expiration date control within a BrightStor CA-1 (Release 4.7 and below) environment, we also recommend that you install the direct interface that BrightStor CA-Disk for USS additionally provides. Review the information for sysparm TMSCTLEXmmmmmmmm in System Parameter Definitions in Appendix E “Sysparms” and user exit TMSCTLEX – Tape Management System Control Interface Exit in Appendix F “User Exits,” as it pertains to the direct interface. Then decide if the direct interface to BrightStor CA-1 is applicable and desirable at your installation. If it is not, you should change the expiration date being assigned through sysparm DYNEXPDT, described in DYNEXPDTE99365, or your JCL to specify either a true expiration date or the value 99000, which can then be used to place the tapes under catalog control as described in Method 2 above. For other applicable rules, see Assigning Tape Expiration Dates in this chapter.

Method 3—Controlling Tapes by Catalog Status

Consider this technique if your tape management system supports an option to designate tapes as in-use as long as they are cataloged, and as available scratch tapes when they are uncataloged. This technique is appropriate for either TLMS or BrightStor CA-1, both from Computer Associates. When BrightStor CA-Disk for USS determines that all data on a tape has expired, BrightStor CA-Disk for USS releases the tape and makes it eligible to be a scratch tape by uncataloging it.

To implement catalog control, you must use the expiration date that your tape management system defines for that purpose. For tapes dynamically allocated by BrightStor CA-Disk for USS, you must specify this value in sysparm DYNEXPDT. If at some point you decide to override dynamic allocation by providing JCL for the output tapes, you must provide this value in the LABEL=EXPDT=yyddd parameter for the tape DD statements.

To have BrightStor CA-Disk for USS catalog the tapes it creates, do not specify DISP=(NEW,CATLG) in the JCL. Instead, specify sysparm ARCTNAME with an appended value of C. BrightStor CA-Disk for USS generates a unique name for each tape it creates, and then catalogs it. You should also specify sysparm UNCATARC with a value of Y, to have BrightStor CA-Disk for USS help keep your system catalog clean by uncataloging each tape when it returns it to the scratch pool.

WARNING! *If you lose the catalog that your BrightStor CA-Disk for USS tapes are cataloged in, your tape management system can scratch all BrightStor CA-Disk for USS tapes. Take appropriate steps to prevent this from occurring.*

Method 4—Controlling Tapes Using TLMS/EDM

This support provides a module for the TMSCTLEX user exit that interfaces with the EDM developed for release 5.0.4 of TLMS. To enable the support, sysparms TMSCTLEX and DYNEXPDT must be set to the values ADSTH017 and E99365 respectively. The TLMS documentation supplied by Computer Associates must also be performed. For detailed information regarding this support, see ADSTH017 – Interface for TLMS in Appendix F “User Exits.”

Interfacing Considerations With BrightStor CA-1

We recommend that you run the PATHDELETE command of IXMAINTU (see PATHDELETE in the chapter “Index Maintenance”) nightly before the BrightStor CA-1 scratch and clean functions are run. This makes the expired BrightStor CA-Disk for USS tapes available to BrightStor CA-1 as scratch tapes.

For users who do not use the EDM interface, if a job abends while writing a tape, BrightStor CA-1 defaults to considering that output tape a scratch. To BrightStor CA-Disk for USS, however, the partial tape is a good tape and must be kept. You should take special precautions to prevent BrightStor CA-1 from marking BrightStor CA-Disk for USS output tapes as scratch tapes after an abend. This can be done manually, or you can make use of one of the exits within the BrightStor CA-1 system. A BrightStor CA-Disk for USS user has supplied a sample for this exit in member SLI035 of the user mod library. Contact your local BrightStor CA-1 support staff or Computer Associates directly, if necessary, if you have further questions regarding their exit.

Tailoring Options

BrightStor CA-Disk for USS provides two primary means of tailoring the system to more specific needs. Become familiar with the capabilities each provides so that you can find relatively simple solutions to conditions that are unique to your installation.

- BrightStor CA-Disk for USS provides for system parameters (sysparms) to be specified in member=SYSPARMS of the parmlib data set. The contents are retrieved at execution time and processing paths are varied accordingly. Default values for all sysparms are generated internally. You need to create member=SYSPARMS and specify a sysparm value **only if you want to change a defined default**. For a list of the available sysparms, see the table in System Parameters in Appendix E “Sysparms.” There is no need to read and review every one of them until an actual need is made evident from your use of the various functions. Some of the more commonly used ones, or ones that are more important to consider in advance, are presented here to acquaint you with their use.
- User exits are also supported. For a complete list of the exits available, see the table in User Exit Cross Reference Matrix in Appendix F “User Exits.”

In addition to sysparms and user exits, the following topics are highlighted in this section:

- Performance Tips
- Backup Considerations
 - Specifying the Archive Medium
 - Using Multiple Types of Media
 - Naming the Archives
 - Assigning an Expiration Date to the Archive Data Set
- Archive Integrity and Security
 - Archive Tape Protection
 - Checks Against Overwriting Valid Tapes
 - Tape Status After Abends
- User-Specified Condition Codes

Suggested System Parameters

Following are suggested system parameters.

System Parameter for Lines Per Page

The number of lines per page on BrightStor CA-Disk for USS-generated output can easily be changed from its default of 58. For details, see RPTLINES in System Parameter Definitions in Appendix E “Sysparms.”

System Parameter for Separator Pages

The number of separator pages can be changed or suppressed through sysparm FLYCOUNT2 described in System Parameter Definitions in Appendix E “Sysparms.”

User Exit for All Sysout Print Lines

For more extensive monitoring or modifications of printed output, such as special routing for certain messages, see SYSOUTEX – SYSOUT Exit in Appendix F “User Exits.”

System Parameters for Backup

By default, BrightStor CA-Disk for USS keeps UNXINDEX records for those files backed up with the BACKUP command for 30 days. To change this setting, see RETRETPDnnnnn in System Parameter Definitions in Appendix E “Sysparms.”

Performance Tips

This section contains performance tips for using BrightStor CA-Disk for USS.

Files Data Set

Because the Files Data Set contains the archive indexes, it is an active and critical data set when doing implicit archiving, backup, or any index maintenance function. ENQ and RESERVE macros are also issued against the data set and its containing volume. Improper selection of the containing volume can result in performance degradation of BrightStor CA-Disk for USS or other jobs trying to access the volume.

For important information, see RSUPPRESn in System Parameter Definitions in Appendix E “Sysparms.”

Back Up Considerations

BrightStor CA-Disk for USS permits numerous tailoring options for archiving and back up functions. Some of the more important options to consider are discussed in this section.

Specifying the Archive Medium

By default, BrightStor CA-Disk for USS dynamically allocates both the primary and duplicate copy to tape. The DD statement used for the primary copy is `//ARCHIVE0` and for the duplicate copy it is `//ARCHIVEC`. Sysparms `ARC0TYPE` and `ARCCTYPE` allow you to change the defaults from tape to either disk or 3480 cartridge units. Specify these sysparms with a value of `DISK`, `TAPE`, or `3480`, and place them in the `SYSPARMS` member of the `parmlib` data set. There are two ways to indicate that no duplicate copy is to be made:

- Specify “NULL” to sysparm `ARCCTYPE` as “NULL”
- In JCL, specify “DUMMY” to the `//ARCHIVEC` DD statement

For example, to direct the primary copy to disk and the duplicate copy to a 3480 cartridge unit, specify `ARC0TYPEDISK` and `ARCCTYPE3480`.

You can also specify the archive medium through JCL. If tape or 3480 devices are specified through JCL, the devices allocated at job initiation are used throughout the run; that is, the device is not deallocated at the end of each tape, but only at the end of the job. The dynamic allocation option can be made to behave in much the same way by specifying sysparm `DYNUKEEP`. The only exception is when more than five volumes are needed to contain a single archive data set, in which case dynamic allocation is used.

You can also specify archival to disk in JCL by allocating any disk device to the `//ARCHIVE0` DD statement. For example:

```
//ARCHIVE0 DD UNIT=SYSDA,VOL=SER=anyvol,DISP=SHR
```

Notice that this only indicates to BrightStor CA-Disk for USS to use disk archive. The volume (or data set) named on the DD statement has no meaning to BrightStor CA-Disk for USS, and dynamic allocation of the `ARCHVOL` is still done.

Using Multiple Types of Media

This section describes how BrightStor CA-Disk for USS can use multiple types of media (that is, 3480s, 3490Es, SILOs, and so on), sysparms to use, IDRC considerations, and data compression.

Selecting the Output Media

The user selects the media BrightStor CA-Disk for USS writes to by specifying a value to the following sysparms, which are often entered as overrides in BrightStor CA-Disk for USS jobs, or kept in the SYSPARMS member in PARMLIB. Following is a list of the output media sysparms:

Function	Primary Sysparm	Copy Sysparm
ARCHIVE	ARC0TYPE	ARCCTYPE
BACKUP	ARC0TYPE	ARCCTYPE

The following table shows the media options that can be specified with the above sysparms. The primary and its copy can be directed to different media. These two output media types are synchronized so that if one is shorter than the other and fills up, both are closed and new tapes are mounted, new disk archive data sets are allocated, or both. Other sysparms that are in effect when the respective media option is used are shown below:

Media Option	Type of Device	Unit for Writing or Reading	Tape Length
TAPE	“Round” tape (default)	DYNTUNITtape	TAPEFEET2300
3480	3480/3490 cartridge	DYNCUNIT3480	CARTFEET0494
DYN1	Special device type 1	DYN1UNIT	TAPEFEET2300
DYN2	Special device type 2	DYN2UNIT	TAPEFEET2300
DYN3	Special device type 3	DYN3UNIT	TAPEFEET2300
DISK	DASD volumes	n/a	n/a
NULL	Disables copy sysparms	n/a	n/a

Specifying the ESOTERIC Unit Name

The media options shown above relate to sysparms that provide the esoteric unit name BrightStor CA-Disk for USS uses to allocate the proper device for writing, reading, and auto restoring. These sysparms are placed in the SYSPARMS member in PARMLIB and should not be changed unless the esoteric name changes, or a media type has been totally removed from BrightStor CA-Disk for USS control. The DYNn sysparm provides the flexibility of defining additional device types to BrightStor CA-Disk for USS that can be installed at various sites.

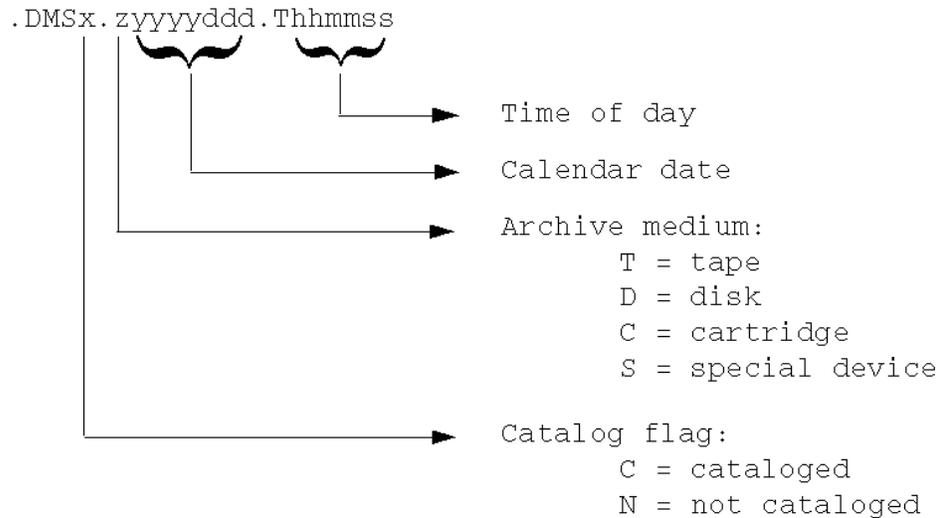
When a tape is created, BrightStor CA-Disk for USS stores the device type in the ARCHVOLS record along with a flag if the DYN1, DYN2, or DYN3 option was used. With this information, BrightStor CA-Disk for USS retrieves the esoteric unit name from the DYN_UNIT or ARESUNI_ sysparm so the proper device can be allocated.

Naming the Archives

In theory, the names you assign to archive data sets are of no consequence to BrightStor CA-Disk for USS. It keeps track of the archive data sets regardless of their names. However, other issues dictate that naming conventions be observed, including the IBM operating system and tape management concerns. BrightStor CA-Disk for USS generates unique archive data set names, providing flexibility in managing the archives and allowing them to be cataloged. Unique data set names also eliminate contention during dynamic allocation that would occur if non-unique data sets were used.

BrightStor CA-Disk for USS sysparm defaults generate unique archive data set names. The base names are taken from either sysparm ARC0NAME or ARCHIVE0 DD statement (for the primary copy) and sysparm ARCCNAME or ARCHIVEC DD statement (for the duplicate copy). If both copies are being produced, the base names should be different, such as DMS.ARCHPRIM and DMS.ARCHCOPY.

Specify up to 22 characters for the base name and BrightStor CA-Disk for USS generates a 22-character appendage as follows:



The calendar date above is the current date in Julian format, and the time of day is in hours/minutes/seconds. Hence the archive data set name provides the date and time it was created, and is always unique.

For disk archive data sets, this generated name is always used and is always cataloged. For tape and cartridge archive data sets, the default is to use the generated name but not to catalog it. You can tell BrightStor CA-Disk for USS to catalog the tape or cartridge archive data set name by specifying a value of C for sysparm ARCTNAME.

JCL Overrides and DSNNAME Usage

As just explained, BrightStor CA-Disk for USS defaults to using dynamic allocation for the archive data sets. For tape and 3480 devices, however, you can override dynamic allocation by supplying the needed information on the appropriate DD statements in the JCL. BrightStor CA-Disk for USS creates the data set name from the DSNNAME you provide in the JCL unless you omitted it completely or specified a temporary name (DSN=&&anyname). In both of these cases, BrightStor CA-Disk for USS uses the appropriate sysparm (ARC0NAME or ARCCNAME) to create the data set name.

Assigning an Expiration Date to the Archive Data Set

By default, BrightStor CA-Disk for USS assigns the “never expire” date of 99365 to each of the archive volumes. This guarantees that an archive volume is never expired before all of the files that it contains. Therefore your only concern is the retention period being assigned to each file being archived. This is recommended if you are using the EDM interface of your Tape Management System.

When BrightStor CA-Disk for USS determines that all files on an archive volume have expired, it automatically expires the volume as well. You can, however, specify a different expiration date through dynamic allocation with sysparm DYNEXPDT, or in JCL through the LABEL= parameter. Append either Eyyddd or Rddddd, where E indicates that a Julian date follows and R indicates that a 5-digit retention period follows.

For related information, see Year 2000 Considerations in the chapter “General Information” and Customizing Tape Management Support in Appendix C “Tape Management.”

Backing Up to Disk: Requirements and Recommendations

Traditionally, backups of files have been directed to tape. BrightStor CA-Disk for USS provides the option of writing backup files to disk. Using disk devices rather than tape or cartridge units for archive data sets eliminates the need for an operator to mount a tape when the backup task is performed or when restoring data.

The following summary outlines the steps necessary to implement a backup to disk. Where necessary, an expanded explanation is provided for some of the summary items.

1. Specify a back up to disk by one of the following methods:
 - If your DMS JCL procedures do not contain //ARCHIVE0 or //ARCHIVEC DD statements, specify sysparm ARC0TYPE with a value of DISK for the primary, ARCCTYPE, or both, with a value of DISK for the copy to allocate them dynamically.
 - Supply JCL statements for the ddnames above that cause allocation to any disk device. BrightStor CA-Disk for USS deallocates and then allocates the device as needed to perform the backup. For example, the following DD statement activates backup to disk for the primary copy:
`//ARCHIVE0 DD UNIT=3380,VOL=SER=anyvo1,DISP=SHR`

2. Specify the base name for the disk archive data sets using the following sysparms. BrightStor CA-Disk for USS appends a unique name to the end of the base name, based on the current date and time.
 - ARCONAME – for the name of the primary
 - ARCCNAME – for the name of the copy
3. Map the names you gave above to proper diskpools (members in your parmlib data set) from which target volumes are selected. This mapping is specified with entries in member POOLDEFS of the parmlib data set. The general form of the entries is:
`'VOLU3380 DMS/'`
4. Place target volumes in your defined pools by creating the parmlib member by the same name as your pool, in this case "VOLU3380," and inserting statements such as:
`VOL=(vo1001,vo1002,vo1/)`
5. Specify an expiration date using sysparm DYNEXPDT, or by JCL. BrightStor CA-Disk for USS assigns the default expiration date of 99365 if none is specified. For related information, see Year 2000 Considerations in the chapter "General Information."
6. Determine if the default values block size is appropriate. If not, override the default with the sysparm ARCDSKBZ.
7. Determine if the default manner of calculating primary and secondary space allocation is appropriate. If not, override the default by specifying the following sysparms:
 - SPACEPRImmm – megabytes to allocate for primary
 - SPACESECmmm – megabytes to allocate for secondary
8. Determine if default values for index maintenance and rebuild functions are appropriate, and if not, specify sysparm to override the defaults.

Step 1—Specifying Back Up to Disk

BrightStor CA-Disk for USS is distributed with default sysparms that cause backup copies of files to be written to tape, creating both a primary and a duplicate copy tape concurrently. You can tell BrightStor CA-Disk for USS to write backup copies of files to disk instead. You can write both the primary and duplicate copies to disk, or assign them to different media. You can also "dummy out" the duplicate copy. Indicate your choice of archive media either through JCL or BrightStor CA-Disk for USS sysparms ARC0TYPE and ARCCTYPE.

Step 2 — Naming the Disk Archive Data Set

For further discussion of this step, see Naming the Archives in this appendix.

Step 3 — Map Data Set Names to Diskpools

Candidate volumes for backing up to disk must be provided through “diskpools,” which reside as members in the parmlib data set. The name of the diskpool is simply the member name. BrightStor CA-Disk for USS determines which diskpool (member) to use in the same manner in which tapepool names are determined. That is, the POOLDEFS member of parmlib is used to associate (map) a data set name to a poolname. For example:

```

File Edit Transfer Options Connection Macro Window Help
-----
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          .DMS90.PARMLIB (POOLDEFS) - 01.02          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000001 'ARCPool0 DMS.ARCHPRIM/'
000002 'BKPPool0 DMS.ARCHCOPY/'
000003 'BKPPool0 DMS.BKUPPRIM/'
000004 'BKPPool0 DMS.BKUPCOPY/'
000005 'ARCPool0 SBDLM./'
***** ***** Bottom of Data *****

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          .DMS90.PARMLIB (ARCPool0) - 01.01          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000001 VOL=(VOL001, VOL002, VOL003, VOL004, VOL005, VOL006, VOL007, VOL008, VOL009)
000002 VOL=(VOL010, VOL011)
000003 VOL=(VOLA/, VOLB/, VOLC/, VOLD/, VOLE/, VOLF/, VOLG/)
000004 VOL=(/)
***** ***** Bottom of Data *****

Aa A TCP/IP R 17 C 15 15:08 3/11/98

```

Step 4 — Place Target Volumes in Defined Pools

Designate your candidate volumes in diskpool=ARCPool0 by creating a member in the parmlib data set with that name and inserting entries. For an example, examine the lower portion of the screen sample above.

A maximum of 100 volumes or patterns can be entered, with one or more volumes allowed per statement. Do not try to continue the volume list on multiple lines (that is, continuation lines are not supported). Instead, enter multiple VOL= statements.

If both the primary and the duplicate copy are being directed to disk, each pool must contain volumes of the same device types as the other pool; that is, if pool 1 has both 3380s and 3350s, pool 2 must also have some 3380s and 3350s. This is required because a volume from pool 1 is selected first, and then pool 2 is searched for a volume of the same device type.

By default, BrightStor CA-Disk for USS automatically separates the copy from the primary; that is, if a volume appears in both pools, BrightStor CA-Disk for USS does not select the same volume for the copy as is used for the primary. This is to prevent the loss of both copies if the disk pack is destroyed due to oxidation, head crashes, or other reasons. If you want to allow both copies to go to the same volume, specify `sysparm ARCSEPC0` with a value of N.

Step 5 — Specifying an Expiration Date

For further information about this step, see *Assigning an Expiration Date to the Archive Data Set* in this appendix.

Step 6 — Specifying Block Size

When backing up to disk, BrightStor CA-Disk for USS automatically adjusts the optimal block size value for the device type of the volume; that is, half-track blocking on 3375s and 3380s, and full track blocking on other devices. You can override the defaults by specifying `sysparm ARCDSKBZ`. The block size selected for a disk device is also used for any tape or 3480 copy that is being created concurrently.

If you have more than one device type in your archive pool, use `sysparm ARCDSKBZ` to specify a block size appropriate for all devices; that is, one that provides optimal performance and space usage on all the devices.

Step 7 — Specifying Space Allocation

BrightStor CA-Disk for USS determines the amount of space to allocate (the amount of space it needs on a diskpool volume) in the following manner. It assumes that quite frequently the primary copy is on disk and the duplicate is on tape or a 3480 cartridge. The target volume leads BrightStor CA-Disk for USS to the device type, which in turn dictates the block size to be used. This, coupled with the density and length of tape being used, allows an easy and very accurate calculation of the tape capacity. Since the ends of 3480 tapes cannot be clipped off, the most common variable in the calculation has been removed.

The optimal disk allocation is the exact same capacity as its backup medium, such that when one is filled and both must be closed, nothing is wasted on the other medium. To do this on an exact equivalence, however, would require very large amounts of free space to be available. 1/16th of that value is used as the default primary and secondary space allocations. In many cases, a backup run does not need a full tape, and one or two extents are more than enough space. However, when a large backup run is made and the diskpool volumes contain plenty of free space, BrightStor CA-Disk for USS obtains as many of the 16 extents as possible before closing the archive data set. This maintains high tape usage as well. Any excess or unused disk space is released immediately when the data set is closed.

BrightStor CA-Disk for USS also considers the size of the files being backed up when determining how much space to allocate. If the first file to be backed up is larger than the default primary space, the primary space is reset to the size of the file. If free space equal to this new primary value is not available but the default value is, BrightStor CA-Disk for USS attempts to backup the input file to a multivolume output data set.

After the target archive data set is allocated and the first file to be backed up is copied to it, BrightStor CA-Disk for USS continues to copy additional files (the second through nth) as long as there is sufficient space to hold them. To determine whether sufficient space is available, BrightStor CA-Disk for USS examines the unused space in the current extent, and checks to see if additional extents can be obtained. If there is not sufficient space, the current disk archive data set is closed and a new one is allocated (like swapping to a new tape).

If your analysis or practical experience indicates that the default method of calculating space allocations should be changed, specify sysparms SPACEPRI and SPACESEC with a three-digit number representing the number of megabytes of disk space to allocate for each.

When backing up to disk, you normally want the archive data sets to be large, since restoring from a large data set is as fast, or faster, than restoring from several small data sets. BrightStor CA-Disk for USS provides options for balancing the need to conserve storage space with the need to keep overhead to a minimum. If you set the primary space value (sysparm SPACEPRI) to the size you want to use for staging, then set the secondary space value (sysparm SPACESEC) to 000. BrightStor CA-Disk for USS creates smaller archive data sets and more of them.

Step 8—IXMAINT and REBUILD

To accommodate backup to disk, the following options are available within the functions listed below:

Function	Option
IXMAINT	Parameter SCRATCH=YES/NO is available on the DSNDELETE, VOLDELETE, and PURGE commands. DSNDELETE and VOLDELETE default to SCRATCH=YES, which means to scratch and uncatalog the disk archive data set when the archvols record is being deleted. PURGE defaults to SCRATCH=NO, because it is frequently used just prior to running REBUILD. If a disk archive data set is scratched, there is nothing to rebuild it with. If you do not intend to follow PURGE with a REBUILD command, specify SCRATCH=YES. Otherwise you are creating “disconnected” archive data sets on disk that are never used or deleted by any other means.
REBUILD	Rebuild accepts the archive data set to be read as input from either tape or disk. The supplied JCL procedure requires that you provide the correct //ARCHIVES DD statement pointing at the tape or disk data set to be rebuilt. No input cards are required. If they are provided, only the expiration date parameter is used.

Archive Integrity and Security

This section discusses archive integrity and security.

Archive Tape Protection

You can set your security package to assign write (update) access to users who must run backup jobs to BrightStor CA-Disk for USS archive data sets on tape. You can then assign access of none to all other users. This allows only authorized users to read, modify, or delete these archives through utilities outside of BrightStor CA-Disk for USS.

BrightStor CA-Disk for USS restore processing takes care of its own access to the archives, if the BrightStor CA-Disk for USS sysparm TAPEPRTR is specified with the default of Y. To determine the data set names of the BrightStor CA-Disk for USS archives, see Naming the Archives in this appendix.

BrightStor CA-Disk for USS password-indicates each archive tape, but does not create any passwords for them, if you specify the BrightStor CA-Disk for USS `sysparm TAPEPROT` with a value of Y. Also, this password-indication does not allow users to read, modify, or delete these archives through utilities outside of BrightStor CA-Disk for USS. This protection is selectable, and supplements any protection provided by your security package.

Checks Against Overwriting Valid Tapes

When a new backup tape is to be created, always check the archive index to see if it is already listed as having valid archive data. This can happen under the following conditions:

- An operator mounts the wrong tape in response to a request to mount a scratch tape.
- You have placed tapes into a pool for the first time, but they have already been used.
- The status of the UNXINDEX has not been kept synchronized with your tape management control data set. Your tape management system indicated that the tape is an available scratch, but maintenance jobs to indicate it as such have not yet been run.

BrightStor CA-Disk for USS rejects a tape that is mounted but found to be already in use (it still has valid data on it), and calls for a new tape. Additionally, if short tapes are mounted and tapepools are being used, the next tape is selected from the appropriate pool.

For non-DFP systems (which exist only in the MVS/370 environment) there are two different cases:

- If tapepools are being used, the tapes are validated just as in DFP systems.
- When tapepool support is not being used, however, BrightStor CA-Disk for USS does not know the volume serial of the tape until after the operator has mounted it in response to the “mount scratch tape” request. If BrightStor CA-Disk for USS sees that it is already in the archives at this point, data that is possibly valuable is on the verge of being overwritten. In fact, since the tape label has already been written, the first file on the tape can be unrecoverable. To prevent any further damage, BrightStor CA-Disk for USS issues appropriate messages for the tape that can need to be recovered, and then shuts down, keeping the remaining files on the tape intact.

Tape Status After Abends

The BrightStor CA-Disk for USS UNXINDEX is updated immediately after each disk data set has been successfully copied to a tape. With this implementation, every backup tape created by BrightStor CA-Disk for USS is a good tape, even though a system failure or job cancel could have occurred.

If you are using a tape management system, its default action can be the opposite, that is, it can return a tape in use at abend time to scratch status. See your tape management personnel to either change this action for tapes created by BrightStor CA-Disk for USS or take other precautions to ensure that a partial backup tape is not marked as a scratch tape, and can be written over by some other job.

Using RMM as their tape management system should review APARs OW17304 and OW17377 and consider using the optional JOBNAME Control with program name ADSMI002 and for Unload, ADSMI000. This functionality allows specific tapes to be kept if they are left in an Open status or were being created when an abend occurred. Any questions on these APARS should be directed to RMM support at IBM.

Using the EDM interface of a tape management system with BrightStor CA-Disk for USS is highly recommended. It provides the best method of securing BrightStor CA-Disk for USS tapes, including those being created as a result of an abend.

If BrightStor CA-1 Release 4.8 or higher is installed and the BrightStor CA-Disk for USS/BrightStor CA-1 EDM interface is used, a tape created by an abending BrightStor CA-Disk for USS job is controlled by BrightStor CA-Disk for USS and released only when the files on the tape have expired. See Customizing Tape Management Support in Appendix C “Tape Management.”

Contact Computer Associates Customer Support for information about other EDM interfaces that are available.

If BrightStor CA-1 Release 4.7 or below is installed and the BrightStor CA-Disk for USS/BrightStor CA-1 direct interface is used, a BrightStor CA-1 exit (member SLI035 in the USERMODS library) is available for the proper treatment of BrightStor CA-Disk for USS abend tapes. See TMSCTLEX – Tape Management System Control Interface Exit in Appendix F “User Exits.”

Note: If BrightStor CA-1 Release 5.0 or above is installed, do not use the BrightStor CA-Disk for USS/BrightStor CA-1 direct interface. BrightStor CA-1 macro changes have made this interface incompatible.

If BrightStor CA-Disk for USS tapepool support is used, tapes are usually indicated by the tape management system as being assigned permanently to BrightStor CA-Disk for USS and physically separated from the general scratch pool available to non-BrightStor CA-Disk for USS jobs. Since BrightStor CA-Disk for USS knows that the partial tape was written, it is not called from the pool until its expiration date has passed. For more details, see Tapepool Considerations in Appendix C “Tape Management.”

User-Specified Condition Codes

BrightStor CA-Disk for USS assigns a step completion condition code to every step. The code accompanies a message and indicates the circumstances under which the step finished. In most cases, this BrightStor CA-Disk for USS-generated condition code is the best value for your installation. However, you can override these generated condition codes if your installation has special requirements. This override allows you to select the code that accompanies a message upon the completion of a step. These user-specified condition codes are optional.

To specify user condition codes, you must create a member in PARMLIB called CCSET. This member is not provided with the system, although a member called SAMPCSET is provided as a sample, and can be copied to create your initial CCSET. User condition codes are then specified by an entry in this member.

When using this option, several technical points must be kept in mind with respect to user condition code processing:

- After an N (no-override) is encountered during processing, BrightStor CA-Disk for USS determines, sets, and assigns condition codes for all the message number entries that follow without regard for the overrides specified in member CCSET.
- Whenever a message is issued during step processing, the highest condition code is set without regard to its origin.
- Before BrightStor CA-Disk for USS returns control to the operating system, a check of the BrightStor CA-Disk for USS-generated return code is made. If the BrightStor CA-Disk for USS return code is odd and the user condition code option is in effect, then the user override condition code is incremented by one. For example, if the user specified an override condition code of 4 and the BrightStor CA-Disk for USS generated condition code was found to be odd, then the user override condition code of 4 would be incremented to 5. This is needed to produce sorted report listings.
- No duplicate message number entries are allowed in member CCSET. If duplicates are detected at table initialization, the program abends with a user 200.

Each CCSET entry is composed of a message number, a state flag (O=override and N=no-override), and one of five possible condition codes. The message number is the first four positions followed by a blank. The next two positions consist of the state flag and a condition code respectively.

To add, change, or delete entries in the CCSET member, use your online editor or other conventional means for updating a PDS member in the following manner:

1. Create a member named CCSET in the parmlib data set by copying and renaming the sample member SAMPCSET.
2. Specify the message display control entry in CCSET. This must be the **first** entry in CCSET and is required. This entry controls the display of the message that notifies users that the user condition code option is in effect. It is recommended that you try the default first, which displays the message in BATCH and TSO. The format for this entry is:

Field	Meaning		
\$FLG	\$FLG		
X	BLANK		
y	BATCH FLAG	B	Display message under BATCH
		BLANK	Do not display message
z	TSO FLAG	T	Display message under TSO
		BLANK	Do not display message

In order to suppress the display of the notify message under BATCH or TSO processing, simply blank out the respective flag. If you wish to suppress the display of the notify message under BATCH and TSO processing, blank out both flags.

3. Specify the message number, state flag, and condition code enclosed by apostrophes in the CCSET member. An example could be:
'3179 OE'

The format for message number, state code, and condition code is:

Field	Meaning
MMMM	Message number
B	Blank
s	State flag
c	Condition code

The valid entries for these are:

Field	Possible Values
MMMM	Any valid four-digit BrightStor CA-Disk for USS message number
B	Must be blank
s	O Oooverride N No-override
c	G 0, good, successful completion I 4, informational message R 8, resource error E 12, error occurred during BrightStor CA-Disk for USS processing C 16, command error, DSCL command error detected

Sysparms

Each installation using BrightStor CA-Disk for USS to manage data storage has unique needs, requiring a variety of techniques for implementing BrightStor CA-Disk for USS to provide compatibility with each environment. To meet these various needs, BrightStor CA-Disk for USS was developed to use system parameters (sysparms) to control various aspects of each function. You can change the default value of each sysparm to meet the operating requirements of your installation. Most system parameters can also be specified temporarily, through overriding DD statements, to accommodate any special circumstances of a particular job run.

Each function in BrightStor CA-Disk for USS has logic to access a sysparm when it is necessary to determine the value in effect. After the value has been retrieved, program logic invokes the code to perform the function according to the sysparm value specified.

SYSPARMS Member in PARMLIB

BrightStor CA-Disk for USS assigns a default value to every system parameter and executes according to this default value unless a different value is assigned through an entry in the parmlib data set member SYSPARMS or through an overriding DD statement.

The BrightStor CA-Disk for USS default value of a sysparm is assigned based on "typical" installation needs. In many cases, the default value is the best value for your installation and no adjustments are required. Only if the default value does not meet the needs of your installation do you need to make any changes.

To change a sysparm to a value other than its default value, first create a member in the parmlib data set called SYSPARMS. This member is not provided with the system. A member called SAMPARMS is provided as a sample, and can be copied in to create your initial SYSPARMS member. System parameters are then specified in the SYSPARMS member, one parameter per line. If more than one entry is entered on a line, all but the first are treated as comments.

Each system parameter is composed of a parameter name and a parameter value. The name is the first eight characters and the value is appended immediately after the name. All parameters have unique names; the length and type of data required in the value differ. The default value and the acceptable user-specified values for each sysparm are documented later in this section under each sysparm name.

To add, change, or delete entries in the SYSPARMS member, use any online editor or other conventional means for updating a member of a partitioned data set.

System Parameter Names Reserved for Users

You can create your own system parameters, perhaps in conjunction with a user exit or for user-supplied applications using BrightStor CA-Disk for USS modules. In order to ensure that sysparms you create do not conflict with BrightStor CA-Disk for USS-supplied sysparms in a future release, specify sysparms you create with a name beginning with a dollar sign (\$). By definition, all sysparm names beginning with a \$ are created and specified by the user or installation.

Overriding Sysparms Instream

To provide maximum flexibility, BrightStor CA-Disk for USS also permits the user to supply an input stream of overriding system parameters through the //SYSPARMS DD statement. Sysparms thus supplied either supplement those specified in your SYSPARMS member of parmlib or override them if already present.

The DD statement must point to a sequential file with DCB attributes of LRECL=80 and RECFM=F or FB. It can be a dd * type input stream. Sysparms start in column 1 of the record image, as shown in the example below:

```
//SYSPARMS DD *
* COMMENTS HAVE ASTERISK IN COLUMN ONE.
ADPCTPDS025
FLYCOUNT0
PRIEXMODMYEXIT                ADD A USER EXIT
'DSDISPEX '                    BLANK OUT USER EXIT
/*
```

The capability to override sysparms through an input stream is enabled by specifying sysparm SYSPARMO with a value of Y or S in the SYSPARMS member of parmlib. For details, see SYSPARMO. The default value of N disables this feature.

Note that the sysparm values specified in the input stream are temporary overrides. They do not update the contents of the SYSPARMS member of the parmlib data set. To protect against unauthorized use of restricted functions or operations, certain sysparms cannot be overridden by this technique.

A cross-reference matrix of all sysparms and a more detailed discussion of each, including the default value, are provided in the remainder of this section.

System Parameters

Following is a cross-reference matrix of system parameters:

Sysparm	Sysparm									Sysparm	Sysparm								
	ARCH	REPORTS	BACKUP	RESTORE	RECOVER	LISTU/V	IXMAINT	FILES MGT	ARCHIVE		REPORTS	BACKUP	RESTORE	RECOVER	LISTU/V	IXMAINT	FILES MGT		
ARCCNAME	•		•						ARCCTYPE	•	•								
ARCDKBZ	•		•						ARCDNEX	•	•								
ARCEODEX	•		•		•				ARCHPACK	•	•	•	•		•				
ARCHSORT	•		•						ARCSEPC0	•	•								
ARCTNAME	•		•				•		ARCVOLID	•	•								
ARCONAME	•		•						ARC0TYPE	•	•								
CARTCALC	•		•						CARTFEET	•	•								
CARTPRCT	•		•						DASDPMAX				•	•					
DASDVMAX					•	•			DATECUTO	•	•	•	•	•	•	•			
DIAGUNIX _n	•		•						DISKPROT	•	•								
DSCLRPLY	•	•	•						DSNSELRE	•	•	•							
DYNCUNIT	•		•	•	•				DYNEXPDT	•	•								
DYNTUNIT	•		•	•	•				DYNUKEEP	•	•	•	•						
DYN _n UNIT	•		•	•	•				DYNVREQD	•	•								
ENQADDTL	•		•	•	•				ENQMSGES	•	•	•	•						
ENQSTPOL	•	•	•	•	•				ENQVOLWT	•	•	•	•						
FILESCNT							•		FLYCOUNT	•	•	•	•	•	•	•			
FORCPOOL	•		•				•		IDRCRATE	•	•								
IDRCUSED	•		•						IOCHNBLK _n	•	•								
IXMCLEAN							•	•	IXMQTIME	•	•	•	•		•	•			
IXUSRNOT							•		LISTSORT			•							
MERC _n NAM									MERC _n TYPE										

Sysparm								Sysparm								
	ARCH	REPORTS	BACKUP	RESTORE	RECOVER	LISTU/V	IXMAINT		FILES MGT	ARCHIVE	REPORTS	BACKUP	RESTORE	RECOVER	LISTU/V	IXMAINT
MERFASTS								MERPnNAM								
MERPnTYP								MERREPLY								
POINTABS				•	•			RESCOPYT				•	•			
RETRETPD				•	•			RPTLINES		•						
RSUPPRES							•	SELECTRE	•	•	•					
SORTPADD		•						SPACEPRI	•		•					
SPACESEC	•		•					SYSOUTEX	•	•	•	•	•	•	•	•
SYSPARMO	•	•	•	•	•	•	•	SYSPARMS	•	•	•	•	•	•	•	•
TABLESDD		•	•					TAOPPOEX	•		•					
TAOPPREX	•		•					TAPCTLEX								•
TAPEDLEX							•	TAPEFEET	•		•					
TAPEFSCR	•		•					TAPEPOOL	•		•					•
TAPEPROT	•		•					TAPEPRTR	•		•					
TAPEVVER	•		•					TMSCTLEX								•
TPMGMTX							•	UCC1BYP5								•
UCC1EXPR							•	UCC1EXTN								•
UCC1SCRT							•	UNCATARC								•

System Parameter Definitions

To place in effect a value other than the default value for any of the following system parameters, create a one-line entry in member SYSPARMS of the parmlib data set, enter the name of the sysparm, and append the appropriate value:

Parameter	Description
ARCCNAMEnnn	If MVS dynamic allocation allocates the //ARCHIVEC DD statement (for the duplicate copy being written to the archives), this sysparm provides the file name to be used. The nnn value is the 1- to 22-character base name, the complete name, or the tape GDG index, depending on installation JCL and other sysparm options. The default value is DMS.ARCHCOPY.
ARCCTYPEtape	If the //ARCHIVEC DD statement is not provided in the JCL, this sysparm indicates the type of device to be dynamically allocated. It must specify one of the following: TAPE, 3480, DISK, DYN1, DYN2, DYN3, or NULL. TAPE is the default.
ARCDSKBZbbbb	BrightStor CA-Disk for USS assigns an optimal block size to an archive data set being written to disk, based upon the disk device type. To override the computed value, specify the block size for the bbbbb value. The default of 00000 indicates that the computed value is used.
ARCDSNEXmmmm mmmm	This sysparm indicates the module to be given control after a BrightStor CA-Disk for USS archive data set name is generated.
ARCEODEXmmmm mmmm	Specify this sysparm to provide the module name to be invoked for the ARCEODEX user exit.
ARCHPACKy	Specify this sysparm to control archive file packing. The default value is Y, which causes each archive data set to contain multiple files. A value of N creates one archive file for each file processed. This sysparm controls file packing for both the merge and archive functions. <i>WARNING! For archival to tape or cartridge, do not specify a value of N or every archive tape or cartridge contains only one file.</i>
ARCHSORTudv	Specify this sysparm to change the selection of archive reports produced. The default value of UDV causes archive reports to be sorted by the following fields: <ul style="list-style-type: none"> ■ U – print unsorted archive report ■ D – print archive report in file/path sequence ■ N – specify N for any report not wanted
ARCSEPC0y	When both the primary and the duplicate copy archive data sets are being written to disk, the default value for this sysparm is Y, which instructs BrightStor CA-Disk for USS to always separate the copy from the primary; that is, never put the backup copy on the same disk volume as the primary. Specifying N (no separation) permits allocation to the same volume.

Parameter	Description
ARCTNAMEd	<p>By specifying one of the following four values for this parameter, you specify how to generate archive tape data set names, and whether they are to be cataloged:</p> <ul style="list-style-type: none">■ D – generates a unique tape data set name but does not catalog it (default value)■ C – generates a unique tape data set name and catalogs it■ G – uses GDG support to generate a unique tape data set name and catalog it■ U – uses data set name specified in JCL, or the exact value of sysparm ARCxNAME <p>Options D (default value) or C are preferable to G or U and should be used wherever possible. These options generate unique data set names for each tape data set created. They provide significantly enhanced flexibility if you are using tape management programs such as BrightStor CA-1, and eliminate data set name contention during allocation. Option C also catalogs the tape data set names.</p> <p>Setting this sysparm to a value of G provides the benefits of generating unique tape data set names, requires building of GDG indexes, and is limited to 255 generations per index. It is maintained strictly to provide compatibility with prior releases.</p> <p>Setting this sysparm to a value of U forces BrightStor CA-Disk for USS to create output archvols based on either:</p> <ul style="list-style-type: none">■ The value specified for DSN= in JCL■ The value specified for ARCxNAME <p>If you specify a value of C or G, you should also specify sysparm UNCATARC with a value of Y to tell BrightStor CA-Disk for USS to uncatalog the archive data set when it expires the volume.</p> <p>If your security system protects tape volumes and you want to specify options C or G for this sysparm, you must also ensure that all users running backup or archive jobs have access to tape volume DMSTMP for the following reason.</p> <p>To serialize tape names, BrightStor CA-Disk for USS temporarily catalogs the data set name to the fictitious volume DMSTMP. BrightStor CA-Disk for USS then opens the tape, determines the actual tape volume that was mounted, and then adds it to the catalog entry, replacing the temporary value of DMSTMP. For this catalog action to take place correctly, the user running BrightStor CA-Disk for USS must have access to tape volume DMSTMP.</p>

Parameter	Description
ARCVOLID@#\$\$+-%&[<p>When archival to disk is being used, the key to the ARCHVOLS record is generated by BrightStor CA-Disk for USS. Taking the first character of this sysparm and appending 99999 to it builds each key. Each successive key decrements by one to make it unique. When 00000 is reached, the second character from this sysparm is used and the process starts over.</p> <p>The default value for this parameter is @#\$\$+-%&[. You can change the order, or specify other special characters. However, you must not specify any characters that are used in volume serial numbers at your data center, or any of the BrightStor CA-Disk for USS pattern characters (*,?,/,!). Valid characters are:</p> <p>[+] & \$; ^ - % _ : # @ ~ \</p>
ARCONAMEnnn	<p>If MVS dynamic allocation is used to allocate the //ARCHIVE0 DD statement (for the primary copy being written to the archives), this sysparm provides the data set name to be used. The nnn value is the 1- to 22-character base name, the complete name, or the tape GDG index, depending on installation JCL and other sysparm options. The default value is DMS.ARCHPRIM.</p>
ARCOTYPEtape	<p>If the //ARCHIVE0 DD statement is not provided in the JCL, this sysparm indicates the type of device to be dynamically allocated. It must specify one of the following: TAPE, 3480, DISK, DYN1, DYN2, or DYN3. TAPE is the default.</p>
CARTCALCn	<p>This system parameter instructs BrightStor CA-Disk for USS as to which method to use when determining cartridge capacities.</p> <p>The default value of N uses a manual calculation. For every output ARCHVOL on a cartridge, BrightStor CA-Disk for USS factors in the value specified for CARTFEET, along with the number of bytes, blocks, block size, density (fixed value at 38000 BPI), and the size of inter-record gaps to determine the capacity of the cartridge. The result is then added to the number of bytes and blocks previously written to the ARCHVOL. If the file fits, BrightStor CA-Disk for USS begins transferring the data. If the file does not fit, BrightStor CA-Disk for USS closes the ARCHVOL and begins transferring the data onto a new ARCHVOL.</p> <p>Following are some special considerations when setting CARTCALC to "N":</p> <ul style="list-style-type: none"> ■ No attempt is made to recognize the differences in cartridges, device types, or both. If your data center changes device types, cartridge brands, or begins to use software, hardware compression (IDRC), or both, you need to revisit your setting for CARTFEET. An incorrect setting can under use the capacity of your cartridges. ■ When using IDRC, the actual compression rate varies because different data compresses differently. Consider changing CARTFEET to 9999 for maximum tape utilization. Any other value results in wasted space. ■ Do not use the StorageTek Redwood or IBM Magstar devices in conjunction with this sysparm set to N. Even setting CARTFEET to 9999 results in wasted space on these devices because CARTFEET only supports devices of up to 4GB capacities.

Parameter	Description
	<p>Specify this sysparm with a value of Y to use a more automated calculation. For every output ARCHVOL on cartridge, BrightStor CA-Disk for USS obtains attributes from the mounted cartridge itself. IDRC compression is factored in allowing for greater precision in predicting usage. This information is stored in an internal table that associates specific attributes to a given cartridge. If necessary, these values can be adjusted through the use of CARTPRCT.</p> <p>Following are some special considerations when setting CARTCALC to "Y":</p> <ul style="list-style-type: none">■ When using IDRC, you can save on CPU usage by disabling software compression. In many cases, disabling software compression allows IDRC to achieve higher compression rates.■ CARTFEET is ignored in all cases, with the exception of the following:<ul style="list-style-type: none">– If the attributes did not map into a given table entry, the value of CARTFEET determines the capacity of the cartridge– There was no way of dynamically determining the capacity of the mounted cartridges for ARCHVOLS created prior to Release 9.0. In this situation, the value of CARTFEET is used to determine the capacity of the cartridge. However, if Y is specified for sysparm CARTCALC, new ARCHVOLS contains the appropriate information.■ The StorageTek Redwood and IBM Magstar devices must be specified as a DYNn unit.
CARTFEETnnnn	<p>The default value for this sysparm is 0494, approximately the uncompressed size of a 200MB 3480 cartridge.</p> <p>This sysparm is closely associated with CARTCALC. When CARTCALC is specified with a value of N, BrightStor CA-Disk for USS factors in the value specified for this sysparm, along with the number of bytes, blocks, block size, density (fixed value at 38000 BPI), and the size of inter-record gaps to determine the cartridge capacity. The result is then added to the number of bytes and blocks previously written to the ARCHVOL. If the file fits, BrightStor CA-Disk for USS begins transferring the data. If the file does not fit, BrightStor CA-Disk for USS closes the ARCHVOL and begins transferring the data onto a new ARCHVOL.</p>

Parameter	Description
CARTPRCTnnn	<p>This sysparm is closely associated with CARTCALC. When CARTCALC is specified with a value of Y, BrightStor CA-Disk for USS obtains attributes from the mounted cartridge itself. This information is stored in an internal table that associates specific attributes to a given cartridge. If necessary, these values can be adjusted through the use of this sysparm.</p> <p>This parameter is a percentage value relative to 100 that allows you to customize cartridge capacities. This is useful in cases where variations from different cartridge vendors result in substantial differences in cartridge lengths. When BrightStor CA-Disk for USS factors this percentage into the calculation derived from CARTCALC, it adjusts the cartridge capacity accordingly.</p> <p>Values above 100 force BrightStor CA-Disk for USS to assume that all cartridges are larger (that is, a value of 125 indicates that cartridges are 25% larger than the information in the BrightStor CA-Disk for USS internal tables). Values less than 100 indicate that cartridges hold less data than the information contained in the BrightStor CADisk for USS internal tables.</p>
DASDPMAXnnn	<p>Specify a one- to three-digit value indicating the total number of pools that can be defined in the DASDPOOL member of parmlib. This value is used to determine memory requirements before building a table with the number of pools and volumes available for pooling. See sysparm DASDVMAXnnnn in this table. The default value is 60.</p>
DASDVMAXnnnn	<p>Specify a one- to four-digit value indicating the total number of volumes that can be included in defined DASD pools (the sum of the volumes in all different pools). This also represents the maximum number of volumes that can be specified for a single pool. The default value is 100.</p>
DATECUTO75	<p>This sysparm instructs BrightStor CA-Disk for USS on how to treat two-digit year values entered from command input parameters or sysparms. Its purpose is to provide the user some flexibility in testing or implementing Year 2000 strategies. The default value is 75, indicating that any two-digit year value of 75 or less is considered as 20YY. These dates are stored as MM/DD/20YY. If the "YY" portion of the date is greater than 75, the dates are stored as MM/DD/19YY. You can specify any numeric value ranging from 00 to 99.</p> <p>This allows two digits for input to continue, or you can use four-digit notations. Examples with a default value of 75 are:</p> <ul style="list-style-type: none"> ■ EXPDT=01/01/00 or 00001 is saved as 2000.001 ■ EXPDT=01/01/75 or 75001 is saved as 2075.001 ■ EXPDT=01/01/76 or 76001 is saved as 1976.001 ■ EXPDT=01/01/99 or 99001 is saved as 1999.001 <p>Note: There is no 'special date' exclusion on inputs. Except for special testing needs, do not set DATECUTO to values of 97, 98, or 99.</p>

Parameter	Description
DIAGUNIXn	The default value of 'n' suppresses the printing of informational messages from USS. These messages can be useful to determine which USS functions are not completing correctly but are not severe enough to cause an error (that is, undefined UID/GID fields). Specifying 'y' may cause numerous messages to be printed in MSGPRINT.
DISKPROTy	<p>This sysparm provides password protection for archive data sets that are not DFSMS-controlled. The default value of this sysparm is Y, which causes BrightStor CA-Disk for USS to password-protect, but not generate passwords for, archive data sets on disk and bypass the normal security checks when opening them. BrightStor CA-Disk for USS password-protects these data sets by setting the password indicator DS1IND10 bit in the format-1 DSCB. The password indicator prevents access to archive data sets for volumes that are not DFSMS-controlled.</p> <p>When this sysparm is specified with a value of N, BrightStor CA-Disk for USS does not set the password indicator.</p> <p>The sysparm setting has no effect when the volumes containing the archive data sets are DFSMS-controlled because password protection is not supported in this situation. However, when the sysparm is specified with a value of N or the volumes are DFSMS-controlled, users with read access can browse disk archive data sets. To prevent this access, specify to your security package to grant no access for all users for BrightStor CA-Disk for USS archive data sets.</p> <p>For information on password-protection of archive tapes, see TAPROTn in this table.</p>
DSCLRPLYn	DSCL processing can be instructed to allow the operator to terminate any DSCL job normally through an operator reply. To activate this feature, specify Y for the value of this sysparm. The default value of N deactivates the feature.
DSNSELREn	<p>Specifying a value of Y for this sysparm requires you to code a PATH= parameter on all SELECT statements. The default value of N is equivalent to coding PATH=/ on a SELECT statement which causes BrightStor CA-Disk for USS to attempt to process all files selected by the SELECT command. This can allow unwanted processing to occur.</p> <p>Note: For other related sysparms, see the descriptions of SELECTREy in this table.</p>

Parameter	Description
DYNCUNITxxxxxxx x	<p>Specify for "xxxxxxx" the one- to eight-character unit name for the 3480 or 3490 devices to be used for dynamic allocation of BrightStor CA-Disk for USS archive cartridges needed for the archive, backup, restore, recover, or merge functions. The default unit name is "3480." A specific unit address can also be supplied to force all of these functions to use a single 3480 device.</p> <p>Note: While simulating the MERGE command and for reporting purposes only, you can specify a different cartridge density. This is only valid in simulate mode. In live mode, the actual density of the device is used.</p> <p>For instance, if you want to simulate other devices, change this sysparm to one of the following values:</p> <ul style="list-style-type: none"> ■ 34800 – the device simulated is a 3480 cartridge, and the capacity is 200MB ■ 34900 – the device simulated is a 3490 cartridge and the capacity is 400MB ■ 34901 – the device simulated is a 3490/E cartridge and the capacity is 800MB
DYNEXPDTE99365	<p>If the //ARCHIVE0 or //ARCHIVEC DD statements for the BACKUP function must be dynamically allocated, this sysparm provides the equivalent of specifying either the EXPDT or the RETPD value in the JCL. Specify Eyyddd, Eyyyyddd, or Rddddd, where E indicates that a Julian date follows, and R indicates that the following five digits are a retention period. A value of E00000 is not valid. The default value is E99365, which has a specific meaning to BrightStor CA-Disk for USS.</p>
DYNTUNITxxxxxxx x	<p>Specify for "xxxxxxx" the one- to eight-character unit name for the tape devices to be used for dynamic allocation of BrightStor CA-Disk for USS archive tapes needed for the archive, backup, restore, recover, or merge functions. The default unit name is TAPE. A specific unit address can also be supplied to force all of these functions to use a single tape device.</p> <p>Note: While simulating the MERGE command and for reporting purposes only, you can specify a different cartridge density. This is only valid in simulate mode. In live mode, the actual density of the device is used.</p> <p>For example, if you want to simulate other devices, change this sysparm to one of the following values:</p> <ul style="list-style-type: none"> ■ 01600 – simulates a 3420 device at 1600 BPI ■ 06250 – simulates a 3420 device at 6250 BPI

Parameter	Description
DYNUKEEPn	<p>This sysparm is used with dynamic allocation in all of the archive, backup, restore, recover, and merge functions to determine whether the tape device should be released each time a new tape volume is needed. It applies to both input and output tapes. If this sysparm is specified with a value of N, the default value, the device is freed at completion of each archvols tape. If the sysparm is specified with a value of Y, the same device is kept for the entire job step.</p> <p>Keeping the same device can aid the operations staff in knowing where tape mounts are needed, but it can also increase runtime since the next tape cannot be mounted until the prior one is rewound and dismounted.</p>
DYNNUNITxxxxxx x	<p>Specify a one- to eight-character unit name for special device type DYN1 (2 or 3). This unit name is used for dynamic allocation of archive volumes during archive, backup, restore, recover, retain, and merge functions, whenever DYN1 (2 or 3) has been specified as the device type to be used through one of the sysparms ARC0TYPE, ARCCTYPE, MERPnTYP, or MERCnTYP. When DYN1 (2 or 3) is used, a flag is set in the ARCHVOLS record and the unit name specified in DYN1UNIT (DYN2UNIT or DYN3UNIT) is used for all functions relating to that archive volume. A specific unit address or an esoteric name that maps only to the special devices is recommended as a value for these sysparms.</p> <p>An intended use is to relate MASSTOR M860 or Redwood devices, which can appear to be tape, 3480, or 3490 devices to the operating system, to the special unit names that are required to obtain proper device allocation.</p> <p>Note: While simulating the MERGE command and for reporting purposes only, you can specify a different cartridge density. This is only valid in simulate mode. In live mode, the actual density of the device is used.</p> <p>For instance, if you want to simulate other devices, change this sysparm to one of the following values:</p> <ul style="list-style-type: none"> ■ 01600 – simulates a 3420 device at 1600 BPI ■ 06250 – simulates a 3420 device at 6250 BPI ■ 34800 – simulates a 3480 cartridge with a capacity of 200MB ■ 34900 – simulates a 2390 cartridge with a capacity of 400 MB ■ 34901 – simulates a 3490/E cartridge with a capacity of 800MB ■ 34990 – simulates a Redwood cartridge with a capacity of 10GB ■ 34991 – simulates a Redwood cartridge with a capacity of 25GB ■ 34992 – simulates a Redwood cartridge with a capacity of 50GB ■ 35900 – simulates a 3590 Magstar cartridge with a capacity of 10GB ■ 35901 – simulates a 3590 Magstar cartridge with a capacity of 20GB

Parameter	Description
DYNVREQDn	<p>When dynamic allocation is being used, this sysparm controls whether mount requests for output archive volumes occurs at allocation time or at OPEN time. The default value of N means the archive volumes are mounted at OPEN time. If this sysparm is specified with a value of Y (and sysparm DYNUKEEP is also specified with a value of N), archive volumes are mounted at allocation time.</p> <p>Note: If you are using MASSTOR M860 or Redwood devices for your output archive volumes, you must specify this sysparm with a value of Y and DYNUKEEP with a value of N.</p>
ENQADDTLn	<p>If this sysparm is set to a value of Y and the USERSENQ user exit is supplied with a value of DMSNOENQ, an additional enqueue on the QNAME SYSDSN are performed to determine if a file is in use. If so, a message is issued stating which files were in use at the time of the back up. The default value is N.</p>
ENQMSGESn	<p>This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Computer Associates Technical Support.</p>
ENQSTPOLn	<p>When this sysparm is specified with the default value of N, BrightStor CA-Disk for USS does not process data sets allocated in the job step that is currently running. Enqueues for these data sets normally fail. This is to prevent BrightStor CA-Disk for USS from processing (that is, archive, delete, and so on) the data sets it has in use that can sometimes cause unpredictable results.</p> <p>Specifying this sysparm with a value of Y allows BrightStor CA-Disk for USS to process data sets in the job step that is currently running. Problems mentioned above can occur, however. Other errors can also occur. For example, you can receive SVC99 errors if you attempt to restore a data set that is also allocated to the currently running step.</p> <p>This sysparm does not apply to auto-restore processing.</p>
ENQVOLWTnnn	<p>This sysparm controls the length of time between issuance of WTOR message 3017, which tells the operator how long BrightStor CA-Disk for USS has been waiting for a tape volume that is being used by another job. The default value is eight, for eight minutes. You can specify a value from 1 to 999 minutes.</p>
FILESCNT12	<p>This sysparm controls how many subfiles can be contained in the Files Data Set. The default value is 12.</p>

Parameter	Description
FLYCOUNT2	<p>BrightStor CA-Disk for USS prints report separator pages on the SYSPRINT DD statement for all BrightStor CA-Disk for USS executions. It appears as the first pages printed before BrightStor CA-Disk for USS-generated output. Computer Associates refers to the separator pages as "fly sheets." They provide two benefits for BrightStor CA-Disk for USS users; first, as a separator page for different report steps, and second, to provide useful information about the environment in which BrightStor CA-Disk for USS is running.</p> <p>This sysparm controls the printing of BrightStor CA-Disk for USS fly sheets. It specifies the number of fly sheets to be printed. To suppress their printing, specify a value of zero for this sysparm. To obtain two fly sheets before printed output, specify a value of two, the default value. The value must be a single numeric digit.</p>
FORCPOOLnnnnnn nn	<p>When tape or disk archive pools are being used, the pool names associated with the archive tapes and disk volumes are specified in the POOLDEFS member of the parmlib data set. A pool name supplied in the "nnnnnnnn" field of this sysparm overrides the normal definitions, forcing all archive volumes to be taken and returned to the named pool. This sysparm is intended for use in special testing or recovery environments only.</p>
IDRCRATE00	<p>Specify for this sysparm the average amount of compression realized when using IDRC. This value, when non-zero, is the compression rate for all cartridge devices utilizing IDRC. The default value is 00, causing BrightStor CA-Disk for USS to query the device in order to automatically compute the compression rate. While this can result in better utilization and reporting, the performance overhead of some cartridge devices may not be desirable.</p>
IDRCUSEDx	<p>This sysparm indicates whether IDRC (Improved Data Recording Capability) is activated when creating output tapes. The values that can be assigned are blank, Y, or N. When the sysparm is specified with the default value of blank, it means that whatever system setting exists as a result of the installation default, JCL parameter specification, or both, should be used. When the sysparm is specified with a value of Y, it means that the compaction feature (IDRC) should be activated if the capability is available. When the sysparm is specified with a value of N, it means that the compaction feature should not be activated when creating output tapes.</p>
IOCHNBLKn	<p>Specify this parameter to turn on Archive Output Tape chaining. The value 'n', from one to nine, specifies the number of tape blocks to be written to the tape concurrently. A higher number reduces the number of EXCPs issued to the tape, reducing elapsed time due to tape channel conflicts. It increases the storage utilization by $(n-1) \times \text{tape blocksize} \times 2$. Using disk output media for either primary or secondary output sets this value to one. The default value is one.</p>

Parameter	Description
IXMCLEANy	<p>This sysparm indicates if the BrightStor CA-Disk for USS archive index maintenance utility (IXMAINT JCL procedure) is to remove invalid records from the UNXINDEX and ARCHVOLS subfiles during processing. ARCHVOLS records must relate to existing UNXINDEX records, and UNXINDEX records must relate to existing ARCHVOLS records. Invalid records are those for which corresponding records do not exist in the opposite file.</p> <p>The default value is Y, which indicates that invalid records are to be removed when found. A value of N indicates that records are to remain in the files.</p>
IXMQTIME0000	<p>Normally, BrightStor CA-Disk for USS index maintenance enqueues on the files file and maintains this enqueue for the duration of the process. This processing method causes other functions (that is, BACKUP, and so on) to wait for long periods of time before they can begin processing. This sysparm controls whether IXMAINT should periodically release the enqueue, and if so, for how long.</p> <p>The default value is 0000, which causes IXMAINT to acquire and maintain the enqueue for the duration of the process. Values greater than zero indicate how long (in tents of a second) IXMAINT can hold the enqueue in a locked or enqueued state.</p> <p>As an example, the following is a list of possible values that can be supplied to IXMQTIME:</p> <ul style="list-style-type: none"> ■ 0000 – This is the default value. IXMAINT does not release the enqueue until its processing is complete. The result is 0.0. ■ 0001 – Releases the enqueue 10 times a second. The result is 0.1. ■ 0005 – Releases the enqueue twice a second. The result is 0.5. ■ 0010 – Releases the enqueue after every second. The result is 1.0. ■ 0300 – Releases the enqueue after every 30 seconds. Since all data centers differ, this value might be a good starting point. The result is 30.0. ■ 1000 – Releases the enqueue after every 100 seconds (just over 1.6 minutes). The result is 100.0. ■ 999.9 – Releases the enqueue after every 1000 seconds (approximately 16 minutes). The result is 999.9. <p>Larger values allow IXMAINT to process more efficiently at the expense of running other functions at a slower pace. Smaller values allow other functions to process faster at the expense of running IXMAINT slower. Other functions acquire the enqueue only long enough to add or update a Files Data Set record, after which IXMAINT reacquires the enqueue and continues processing.</p>

Parameter	Description
IXUSRNOTn	<p>This sysparm indicates whether the archive index maintenance utility (IXMAINT JCL procedure) is to send a TSO message to a user ID if only one copy of a data set is left in the Files Data Set and that data set is uncataloged.</p> <p>The default value is N, which indicates that no message is to be sent.</p> <p>A value of Y indicates that a warning TSO message is to be sent. The TSO message warns that there is only one copy of the data set left. It is sent to the user ID that either archived or backed up the data set, or if the expiration date for the data set in the Files Data Set has been updated, it is sent to the user ID that performed the last update.</p>
LISTSORTn	<p>This sysparm is intended for debugging purposes only and is inactivated by default, which provides no debugging information. Specifying a value of Y is not recommended unless requested by Computer Associates Technical Support.</p>
MERCnNAMnnnnnn nnnnnnnnnnnnnnnn nn	<p>Merge processing can intelligently redistribute archived data sets on the basis of their expiration dates through the use of the LIMITS= parameter. Two limit values cause a two-way split, three cause a three-way split, with a maximum of a five-way split. Each of these new primary archive data sets can also have a duplicate copy created concurrently (duplexing). The new primary data sets are referenced through //ARCHIVE1 (2, 3, 4, 5) DD statements, and the copies by //ARCHIVC1 (2, 3, 4, 5).</p> <p>If MVS dynamic allocation is used to allocate the //ARCHIVC1 (2, 3, 4, 5) DD statements, these sysparms provide the data set names to be used. The nnn value is the 1- to 22-character base name, the complete name, or the tape GDG index, depending on installation JCL and other sysparm options. The default values are DMS.MERGCOP1 (2, 3, 4, 5) for the duplicate copy data sets created by the merge process.</p>
MERCnTYPtape	<p>If the //ARCHIVC1 (2, 3, 4, 5) DD statement is not provided in the JCL, these sysparms indicate the type of device to be dynamically allocated. It must specify one of the following: TAPE, 3480, DISK, DYN1, DYN2, DYN3, or NULL. TAPE is the default.</p> <p>Note: This feature is for MVS users only.</p>
MERFASTSy	<p>Normally, simulate processing of the merge function does not invoke BrightStor CA-Disk for USS user exits. This is done to improve the performance of processing. To cause simulate processing to invoke the user exits, specify this sysparm with a value of N.</p>

Parameter	Description
MERPnNAMnnnnnn nnnnnnnnnnnnnnnn nn	<p>Merge processing can intelligently redistribute archived data sets on the basis of their expiration dates, through the use of the LIMITS= parameter. Two limit values cause a two-way split, three cause a three-way split, with a maximum of a five-way split. Each of these new primary archive data sets can also have a duplicate copy created concurrently (duplexing). The new primary data sets are referenced through //ARCHIVE1 (2, 3, 4, 5) DD statements, and the copies by //ARCHIVC1 (2, 3, 4, 5).</p> <p>If MVS dynamic allocation is used to allocate the //ARCHIVE1 (2, 3, 4, 5) DD statements, these sysparms provide the data set names to be used. The nnn value is the 1- to 22-character base name, the complete name, or the tape GDG index, depending on installation JCL and other sysparm options. The default values are DMS.MERGPRI1 (2, 3, 4, 5) for the primary data sets created by the merge process.</p>
MERPnTYPtape	<p>If the //ARCHIVE1 (2, 3, 4, 5) DD statement is not provided in the JCL, these sysparms indicate the type of device to be dynamically allocated. It must specify one of the following: TAPE, 3480, DISK, DYN1, DYN2, or DYN3. TAPE is the default.</p> <p>Note: This feature is for MVS users only.</p>
MERREPLYn	<p>BrightStor CA-Disk for USS merge processing can be instructed to allow the operator to terminate the merge process normally through operator reply, between any input tape mount. To activate this feature, specify Y for the value of this parameter. The default value is N.</p>
POINTABSy	<p>The default value of Y allows BrightStor CA-Disk for USS to determine if a 3480 is in full function mode or not, and to use POINT TYPE=ABS (high speed search) whenever possible. You can specify N to instruct BrightStor CA-Disk for USS not to use the high-speed search facility.</p>
RESCOPYTn	<p>Specify a value of Y to instruct BrightStor CA-Disk for USS to use the duplicate copy tapes for all file restores, including volume recovery. Default processing calls for the primary archive tape unless it has been disabled.</p>
RETRETPDnnnnn	<p>When files are selected to be archived during archive or backup processing, the five-digit value of this sysparm is added to the current date to designate the archive retention period. This value is used unless an overriding parameter has been specified on the ARCHIVE or BACKUP command. The default value of this sysparm is 00030, to keep each file in the archives for 30 days.</p>

Parameter	Description
RPTLINES	<p>This sysparm, which defaults to 058, controls the number of lines per page for the following reports:</p> <ul style="list-style-type: none"> ■ UATT ■ USPC <p>All other reports can be controlled through the following zap:</p> <pre> NAME ADSUT190 LINEMAX VER 0000 058C 58 lines per page REP 0000 0nnC nn lines per page </pre>
RSUPPRESn	<p>Specify a value of Y to suppress issuance of the RESERVE macro on the volume where the Files Data Set resides during index maintenance functions. The integrity of the Files Data Set cannot be guaranteed if this parameter is specified with a Y. In a JES2 environment, you are responsible for ensuring that no other updates are occurring through a cross-system enqueue package. The default value is N.</p>
SELECTREy	<p>The intention of this sysparm is to help prevent runaway jobs from occurring. With the default of Y specified, DSCL requires that at least one SELECT statement be included on every SCAN command. This sysparm has no effect on the FMS PROC. Regardless of its value, no SELECT statements are required for the FIND command. However, if you specify a value of N and fail to provide a SELECT statement on the SCAN command, BrightStor CA-Disk for USS attempts to process ALL files found by the SCAN. This can allow runaway jobs to occur. For example, if a user unintentionally runs:</p> <pre> SCAN REALVOLS DELETE </pre>
SORTPADdy	<p>If the installation sort routine allows a sort key length greater than the actual record length for variable-length records, you can specify a value of N for this sysparm to prevent BrightStor CA-Disk for USS from padding all report records to a length equal to the key. The default value of Y permits the padding of the report records.</p>
SPACEPRIInnn	<p>When archiving to disk, you must specify a primary space allocation value for the archive data set. The default value of 999 causes BrightStor CA-Disk for USS to set the primary quantity equal to 1/16 of an assumed backup tape. You can specify a value from 1 to 998, which represents the number of megabytes to allocate.</p>
SPACESECnnn	<p>When archiving to disk, you must specify a secondary space allocation value for the archive data set. The default value of 999 causes BrightStor CA-Disk for USS to set the secondary quantity equal to 1/16 of an assumed backup tape. You can specify a value from 0 to 998, which represents the number of megabytes to allocate.</p>
SYSOUTEXmmmm mmmm	<p>Unlike other BrightStor CA-Disk for USS user exits, specifying a sysparm does not activate the SYSOUT exit. The module name must be super zapped into the BrightStor CA-Disk for USS output writer module. This entry in the sysparm section is maintained solely as a reference aid.</p>

Parameter	Description
SYSPARMO _n	<p>BrightStor CA-Disk for USS provides the capability for users to override default sysparm values and entries in the SYSPARMS member of the parmlib data set at execution time through the //SYSPARMS DD statement. This provides the flexibility of tailoring parameter values for various executions without having to update the parmlib data set.</p> <p>The default value of N prevents all users from overriding sysparms. To allow all users to override some or all sysparms, or to allow only authorized users this ability, specify this sysparm with a value of Y.</p>
SYSPARMS _n	<p>This sysparm controls the display of system parameters on the message DD statement during BrightStor CA-Disk for USS executions. It gives the user the option of seeing what sysparms were active during the execution.</p> <p>The default value of N indicates no display is to be generated. Other valid values are:</p> <ul style="list-style-type: none"> ■ O – print only those sysparms overridden by SYSPARMS DD ■ S – print only those sysparms supplied by user ■ D – print only the default sysparms ■ B – print all sysparms in effect
TABLESDDparmlib	<p>Specifying this sysparm instructs BrightStor CA-Disk for USS to use that ddname for retrieving the parmlib entries specified by TABLES= on the DSCL FIND, DSCL SELECT/EXCLUDE, and the FMSU SELECT/EXCLUDE commands. The value can be specified as any one- to eight-character ddname. The default is parmlib, which causes BrightStor CA-Disk for USS to use the PARMLIB ddname as normal.</p> <p>Note: Any entry specified on the TABLES= parameter must be found in the ddname specified, or the job terminates.</p> <p>Using this sysparm requires that the user add a DD statement in the jobstep equal to that of this sysparm.</p>
TAOPPOEX _{mmmm} mmmm	Specify this sysparm to provide the module name to be invoked for the TAOPPOEX user exit.
TAOPPREX _{mmmm} mmmm	Specify this sysparm to provide the module name to be invoked for the TAOPPREX user exit.
TAPCTLEX _{mmmm} mmmm	Specify this sysparm to provide the module name, other than ADSTH016, to be invoked for the TAPCTLEX user exit if more than one tape management system is used.
TAPEDLEX _{mmmm} mmmm	Specify this sysparm to provide the module name to be invoked for the TAPEDLEX user exit.

Parameter	Description
TAPEFEETnnnn	<p>The four-digit value specified for this sysparm is the number of feet to be used for writing files on each reel of tape. If a reel that has less than the specified number of feet is mounted, a file may overflow onto a second reel. However, the following file is archived to the beginning of another reel, precluding the possibility that more than one reel of tape must be passed to accommodate a subsequent restore request. The default value for this sysparm is 2300.</p>
TAPEFSCRn	<p>For systems with DFP 1.1 and above, specifying a value of Y allows BrightStor CA-Disk for USS to bypass expiration date and security checking when writing to a tape—in other words, it "forces" its use as a valid scratch tape. The standard operator prompt for permission to write on a date-protected tape is not issued. Similarly, the operating system does not abend the job when a security-protected tape is mounted for output. This can be particularly useful when BrightStor CA-Disk for USS tape pools are used. Tapes are recycled and, by definition, BrightStor CA-Disk for USS knows that it is okay to write on them again. The default value is N, however, since using this facility with general scratch tapes represents the loss of a safety check, not to mention the data, if wrong tapes are mounted.</p> <p>On systems without DFP 1.1 or above, you should keep in mind that IBM does not allow you to change tape file names after they have been protected. You must initialize the tapes to change the file names. This is documented in the <i>IBM OS/VS Tape Labels</i> manual.</p> <p>For example, if your archive tape data set name is DMS.ARCHPRIM and your backup tape data set name is DMS.BACKPRIM, you must initialize your protected archive tapes before they can be recycled and used as backup tapes.</p> <p>Note: If BrightStor CA-1 manages your tapes, do not specify a value of Y, or all BrightStor CA-1 processing is bypassed.</p>
TAPEPOOLn	<p>Specify a value of Y to cause BrightStor CA-Disk for USS to take output tapes for archiving out of the TAPEPOOL table in the DMSPOOLS subfile. The default value of N calls for the operator to mount any available scratch tape.</p>

Parameter	Description
TAPEPROTn	<p>When this sysparm is specified with the default value of N, BrightStor CA-Disk for USS does not password-protect archive data sets on tape and allows normal security checking when opening them.</p> <p>When this sysparm is specified with a value of Y, BrightStor CA-Disk for USS sets the password-indicator on the tape label, but does not generate any passwords. Also, BrightStor CA-Disk for USS bypasses all security checking on the archive output tape volumes.</p> <p>If this sysparm is specified with a value of Y, sysparm TAPEFSCR must also be specified with a value of Y to allow BrightStor CA-Disk for USS to reuse tapes. At last report, this was incompatible with the BrightStor CA-1 tape management system from Computer Associates. If a BrightStor CA-Disk for USS archive tape was created with sysparm TAPEPROT and specified with a value of Y, you must re-label the tape when it is released from the BrightStor CA-Disk for USS archives to turn off the password-indication of the tape label. This applies to BrightStor CA-Disk for USS as well as other applications.</p>
TAPEPRTRY	<p>Specify a value of N for this sysparm to cause BrightStor CA-Disk for USS to use normal security checking on reading archive tapes. BrightStor CA-Disk for USS prompts the operator for the password for archive tapes created with sysparm TAPEPROT specified with a Y.</p> <p>The default value of Y causes BrightStor CA-Disk for USS to bypass security checking when reading archive tapes. BrightStor CA-Disk for USS reads archive tapes created with the sysparm TAPEPROT specified with a Y without operator intervention.</p>
TAPEVVERY	<p>The default value of Y causes BrightStor CA-Disk for USS to use the IBM DFP volume verification exit to verify that each tape mounted for output is indeed a scratch tape, (not in the BrightStor CA-Disk for USS archives already) and, if it is not, to dynamically request another tape. All of this can be done without damaging the mounted tape, and processing can continue.</p> <p>If vendors of other products used in your installation state that they do not support this exit, then specify this sysparm with a value of N to deactivate its use. When a "wrong tape" is mounted under these conditions, BrightStor CA-Disk for USS issues appropriate messages and terminates processing in an effort to preserve the data already on the tape.</p>
TMSCTLEXmmmm mmmm	<p>Specify this sysparm to provide a module name for the TMSCTLEX user exit.</p> <p>Note: In order to activate the RMM support, this sysparm must be specified with a value of ADSTH016. If you are using more than one tape management system, see sysparm TAPCTLEXmmmmmmmmmm.</p>
TPMGMTExmmm mmmmmm	<p>Specify this sysparm to provide the module name to be invoked for the TPMGMTEx user exit.</p>

Parameter	Description
UCC1BYPSn	<p>This sysparm is valid only when TMSCTLEX is specified with a value of ADSTH025. If you want to have BrightStor CA-Disk for USS keep running even if the BrightStor CA-1 system is down, specify a value of Y. The default value of N causes BrightStor CA-Disk for USS to abend with an appropriate message.</p> <p>This sysparm is only used when the user exit program name for sysparm TMSCTLEX is "ADSTH025."</p>
UCC1EXPRy	<p>This sysparm is valid only when TMSCTLEX is specified with a value of ADSTH025. The default value of Y causes BrightStor CA-Disk for USS to expire tapes in the BrightStor CA-1 TMC by setting the expiration date of the tape to the current date. Specify a value of N if sysparm UCC1SCRT is specified with a value of Y.</p> <p>This sysparm is only used when the user exit program name for sysparm TMSCTLEX is "ADSTH025."</p>
UCC1EXTNnnnn	<p>The default value of 0000 for this sysparm causes BrightStor CA-Disk for USS (in BrightStor CA-1 support) to expire a tape by setting the expiration date of the tape to the current date. If you want to allow a grace period before the tape is expired, specify a one- to four-digit number indicating the number of days to be added to the current date to determine the expiration of the tape in the BrightStor CA-1 TMC.</p> <p>This sysparm is only used when the user exit program name for sysparm TMSCTLEX is "ADSTH025."</p>
UCC1SCRTn	<p>The default value of N for this sysparm causes BrightStor CA-Disk for USS (in BrightStor CA-1 support) to expire a tape by setting the expiration date of the tape to the current date. If you want BrightStor CA-1 to recognize the tape immediately as a scratch tape, specify a value of Y for this sysparm. A value of Y for this sysparm supersedes sysparm UCC1EXPR.</p> <p>This sysparm is only used when the user exit program name for sysparm TMSCTLEX is "ADSTH025."</p>
UNCATARCn	<p>This sysparm determines the catalog action to be taken for any archive tape dropped (deleted) in the index maintenance functions (merge/purge/delete) processing. The default value of N causes BrightStor CA-Disk for USS to leave the catalog status unaltered. A value of Y causes BrightStor CA-Disk for USS to uncatalog any tape that is deleted from the archives. This sysparm is intended for joint use with sysparm ARCTNAME.</p>

User Exits

Modules can be incorporated in BrightStor CA-Disk for USS to satisfy unique processing requirements of an installation. These modules are identified to BrightStor CA-Disk for USS by appropriate system parameter entries in member SYSPARMS of the parmlib data set. During execution of BrightStor CA-Disk for USS, the exit modules are dynamically loaded when they are called for the first time.

In order to provide compatibility for existing BrightStor CA-Disk for USS modules with 24-bit addressing, only the following AMODE/RMODE combinations are valid for user exits:

```
AMODE 24, RMODE 24
AMODE 31, RMODE 24
AMODE ANY, RMODE 24
```

Each of the user exits is discussed under the sysparm entry name that is necessary to define the respective module to BrightStor CA-Disk for USS for invocation.

Assembler language conventions are followed in passing parameters to the exit module. Register 1 points to a list of addresses, each of which points to a specific parameter as defined for the exit. In most cases, the exit module must return a halfword code through the designated parameter to indicate to BrightStor CA-Disk for USS what further actions are to be taken.

To implement this, add the following to your user exit (this call need only be executed after):

```
* Stump the function to retrieve during files dataset logging
#CALL ADSUI302,($ISRT,FUNCTION,CTYPE,K08,USRFUNC,UI302RC),VL
*
$ISRT   DC CL5'$ISRT'   Service = Insert
FUNCTION DC CL8'FUNCTION' Field name for retrieval
CTYPE   DC CL2'C'      Service type = character data
K08     DC H'8'        Length of field
USRFUNC DC CL8' '      Fill in user function name
UI302RC DC H'0'        Return code
```

User Exit Cross Reference Matrix

Following is a cross-reference matrix of all user exits:

User Exit Name	REPORTS	BACKUP	RESTORE	RECOVER	LISTU/V	IXMAINT	FILES MGT	User Exit Name	REPORTS	BACKUP	RESTORE	RECOVER	LISTU/V	IXMAINT	FILES MGT
ARCDSNEX		•						TAPCTLEX							•
ARCEODEX		•						TAPEDLEX							•
DSCLSCEX		•						TMSCTLEX							•
SYSOUTEX		•	•	•	•	•	•	TPMGMTX							•
TAOPPOEX		•						UNLOADX							
TAOPPREX		•						USERSENQ		•	•	•			

User Exit Descriptions

This section provides a detailed discussion of each user exit.

ARCDSNEX—Archive File Name Generation Exit

The module name supplied by this exit is invoked at archive data set name generation time. This exit allows you to change the BrightStor CA-Disk for USS archive data set name to meet special user requirements.

When this user exit is invoked, a unique data set name is created and passed as input. Make sure that the data set name created by the exit is unique (even across shared operating systems), and that the name selects valid archive pools.

Use sysparm ARCDNEX to specify the module name to be invoked for this exit. Following is a list of parameters passed to ARCDNEX:

Type	Size	Description
IN/OUT	CL44	UNIQUE ARCHIVE data set NAME
INPUT	CL8	JOBNAME
INPUT	CL4	SYSTEM ID as identified to SMF

ARCEODEX—Archive Tape End Of File (CLOSE) Exit

The module name supplied for this exit receives control after each archive tape is closed. It is used for informational purposes only.

The user is passed the ddname in order to determine if it is a close for the primary or the copy tape, and the JFCB in order to see the actual tape volumes used. The data set name and volume serials in the JFCB can be useful for interfacing to tape management systems that read the JFCB at job termination time only, rather than hooking directly into tape open and close routines. It can also be useful for generating tape labels, or for detecting when a multiple volume archive tape was created.

Use sysparm ARCEODEX to specify the module name to be invoked for this exit. Following is a list of parameters passed to ARCEODEX:

Type	Size	Description
INPUT	H	OPTION— H'1' Regular tape close Type of CLOSE call
		H'2' Final tape close -job is ending
INPUT	CL8	ddname CL8'ARCHIVE0' Close for primary tape CL8'ARCHIVEC' Close for copy tape
INPUT	XL176	JFCB— The JFCB for the named ddname after the tape close has been done
INPUT	XL110	ARCHVOLS—Image of the ARCHVOLS record as it is stored in the ARCHVOLS subfile

DSCLSCEX—DSCL Screening Exit

The module name supplied for this exit receives control when processing any DSCL command. Control is passed after the screening tests have been applied, and directly prior to processing the DSCL action command. The screening tests are based on the DSCL selection criteria, after the volume and file name filtering has been completed. The user can request that BrightStor CA-Disk for USS reject or process the request, regardless of what value was found in the result code parameter.

When the DS1CPOIT bit in the VTOC is set to 1 (on), the user exit DSCLSCEX allows the action commands ARCHIVE, MOVE, EXPIRE, RELEASE, or DELETE to take place only if an installation determined number of days has passed since the file was last referenced. The default value of 0 allows the file to be moved regardless of the value in the last used date. For related information, see the ARCDMEXT section of the *DFSMS/MVS V1R3 Installation Exits* manual.

Use system parameter DSCLSCEX to specify the module name to be invoked for this exit. Following is a list of parameters that can be passed to DSCLSCEX:

Type	Size	Description		
IN/OUT	HW	The value on input informs the exit what BrightStor CA-Disk for USS action takes place if unmodified by the exit. The value on output (if changed) instructs BrightStor CA-Disk for USS to take the new action.	H'0'	Continue Normal Processing
			H'8'	Terminate Processing For Request
INPUT	XL256	File Identifier. Format-1 DSCB and appendage, for file to be processed.		
INPUT	XL140	Volume Identifier. Format-4 DSCB for file. Volume serial is in position 1 to 6.		
INPUT	CL44	Catalog Name		
INPUT	CL8	Function Code	'BACKUP'	backup the file
			'REPORT'	report on the file
OUTPUT	F	This optional parameter indicates the number of days since last referenced for the file to be eligible for processing. The default value is '0'.		

Type	Size	Description
UT		Code
	H'4'	skip the print line

TAOPPOEX—Archive Tape Processing Exit (Post-Open)

The module name supplied for this exit receives control from BrightStor CA-Disk for USS functions that process archive tape data. It is invoked immediately after an archive tape volume has been opened.

The user exit allows the user to inspect the system control blocks related to the operand, change the system control blocks related to the open, or both. Use `sysparm TAOPPOEX` to specify the module name to be invoked for this exit. Following is a list of parameters that can be passed to TAOPPOEX:

Type	Size	Description
INPUT	CL8	Function—Indicates the BrightStor CA-Disk for USS function invoking the exit (that is, backup or restore)
INPUT	CL6	Type "input" indicating read Of Open
		"output" indicating write
INPUT	CL1	Type M indicates primary archive tape being Of processed Tape
		C indicates copy archive tape being processed
INPUT	XL	DCB—Data Control Block used for the open
INPUT	XL	JFCB—Job File Control Block used for the open

TAOPPREX—Archive Tape Processing Exit (Pre-Open)

The module name supplied for this exit receives control from BrightStor CA-Disk for USS functions that process archive tape data. It is invoked prior to opening an archive tape volume.

The user exit allows the user to inspect the system control blocks related to the open, change the system control blocks related to the open, or both. Use sysparm TAOPPREX to specify the module name to be invoked for this exit. Following is a list of parameters that can be passed to TAOPPREX:

Type	Size	Description		
INPUT	CL8	Function—Indicates the BrightStor CA-Disk for USS function invoking the exit (that is, backup or restore)		
INPUT	CL6	Type	“input”	Indicating read
		Of Open		
			“output”	Indicating write
INPUT	CL1	Type	M	Indicates primary archive tape being processed
		Of Tape		
			C	Indicates copy archive tape being processed
INPUT	XL	DCB—Data Control Block used for the open		
INPUT	XL	DCB—Data Control Block to be used when opening the tape volume. It is set up to open when the exit is called		

TAPCTLEX—Tape Management System Control Interface Exit

This exit is identical to TMSCTLEX. The module name supplied for this exit enables users to run more than one tape management system during a conversion process. For a full description and a complete list of input and output parameters available, see user exit TMSCTLEX.

TAPEDLEX—Archive Volume Being Deleted from Archives

The module name supplied for this exit receives control from any BrightStor CA-Disk for USS function that is deleting TAPE/CARTRIDGE archive volumes from the archive index. This exit is bypassed in SIMULATE mode.

The archive volume record passed to this exit is always sequence #1. Parm 2 indicates the total number of archvols records in the list and Parm 3 is the address of the list of arckey for this archive file. For example, if a multivolume archive file is being deleted, Parm 2 is GT H'1' and Parm 3 has more than one arckey in the list.

No return code is accepted by this exit. BrightStor CA-Disk for USS processing cannot be altered. Use sysparm TAPEDLEX to specify the module name to be invoked for this exit.

Type	Size	Description
INPUT	XL110	Archvols Record – This is the archive volume index record from the files file. ARCKEY is in positions 1 - 6. The true volser is in position 12 - 17.
INPUT	H	List Count – Number of archive keys in the list that follows (maximum of 35 for any given multivolume archive file).
INPUT	XL4	List Address – The address of the list of archive keys for the specified archive file.

Note: The number of keys in the list is specified in Parm 2.

TMSCTLEX—Tape Management System Control Interface Exit

The module name supplied for this exit receives control from any BrightStor CA-Disk for USS function that is updating the status of an existing archive tape volume. This exit can then make the appropriate updates in whatever tape management database is used to control the tapes. When BrightStor CA-Disk for USS changes the status of a tape volume, only the files file is updated. This exit must provide the interface to the tape management system.

Note: User Exit TAPCTLEX enables support for sites that use more than one tape management system.

There are two options that correspond to the two reasons for an update:

- BrightStor CA-Disk for USS no longer needs the tape. The tape can be returned to scratch status within the tape management system database. This option is invoked when BrightStor CA-Disk for USS is going to expire or delete an archive volume.
- The expiration date for a tape must be updated. This option is invoked when BrightStor CA-Disk for USS is going to alter the expiration date of an archive volume. When option 2 is invoked, a seventh parameter is passed. This parameter is the new expiration date that BrightStor CA-Disk for USS is going to reset for the archive volume. This parameter does not exist for option 1.

No messages are issued based on the return code passed back from this exit. The exit must provide these messages. The exit must also ensure the integrity of the tape management system database and ensure that tapes do not prematurely expire.

Use sysparm TMSCTLEX to specify the module name to be invoked for this exit. Following is a list of parameters that can be passed to TMSCTLEX:

Type	Size	Description
INPUT	H	OPTION: 1 Expire Tape
		2 Reset Expiration Date
INPUT	CL44	Tape DSN
INPUT	H	Number of tape volumes is always 1
INPUT	CL6	Tape volume
OUTPUT	H	Return Code: 1 Update successful
		2 or more update failed
OUTPUT	H	Reason Code: Not used at this time. Can be set to any HW value.
INPUT	PL4	New EXPDT – (option 2 only)

Although you can choose to write this exit yourself, three exit programs are supplied as part of the BrightStor CA-Disk for USS system.

- ADSTH014 – the interface to the BrightStor CA-1 External Data Manager (EDM) that is supported in Release 4.8 or higher of BrightStor CA-1. Using this program is the preferred method of interfacing with BrightStor CA-1. For implementation instructions, see Method 2 – Controlling Tapes Through the EDM in Appendix C “Tape Management.”
- ADSTH025 – a direct interface for Release 4.7 and below of BrightStor CA-1. This is documented below.
- ADSTH017 – supports the EDM interface developed for Release 5.0.4 of TLMS. This is documented in ADSTH017 in this chapter.

ADSTH025—Direct Interface for BrightStor CA-1

For expiration date control within a BrightStor CA-1 (Release 4.7 and below) environment, Computer Associates recommends that you install the direct interface that BrightStor CA-Disk for USS provides to BrightStor CA-1. Review the information presented below, as well as Method 2 – Controlling Tapes by Expiration Date in Appendix C “Tape Management” and then decide if the direct interface to BrightStor CA-1 is applicable and desirable in your installation. If it is not, you should change the expiration date being assigned through sysparm DYNEXPDT (for a description of this sysparm, see DYNEXPDTE99365) or your JCL to specify either a true expiration date or the value 99000, which can then be used to place the tapes under catalog control. For other applicable rules, see Assigning Tape Expiration Dates in Appendix C “Tape Management.”

Note: If BrightStor CA-1 Release 5.0 or above is installed, do not use the BrightStor CA-Disk for USS/BrightStor CA-1 direct interface. BrightStor CA-1 macro changes have made this interface incompatible.

The BrightStor CA-Disk for USS direct interface to BrightStor CA-1 uses a central module designed to handle the interchange and update of records in the Tape Management Catalog (TMC). All precautions are taken to ensure that updating is thorough and complete. However, if you use a BrightStor BrightStor CA-1 release that is higher than 4.7 and specify sysparm DYNEXPDT to a value other than the default, you should be aware of the following limitations when using this interface:

- Like BrightStor CA-1 Release 4.7 and below, this interface does not support expiration dates beyond the year 2000.
- This interface does no special processing for expiration dates that fall within the ranges of cycle or frequency control.
- This interface does not read or write exception or journal records.

When BrightStor CA-Disk for USS determines that an archive or backup tape can be returned to scratch status, it can inform BrightStor CA-1 in two different ways:

- BrightStor CA-Disk for USS can retrieve the TMC record for that tape and set its expiration date to the current date. The tape is made a scratch when the CA1 scratch and clean jobs are run.
- BrightStor CA-Disk for USS can take the same action as above, but also turn on the scratch bit indicator flag. Turning on this bit makes the tape an immediate scratch. This is particularly useful when also using the BrightStor CA-Disk for USS tape pool support.

Use of the direct interface to BrightStor CA-1 is not required. It simply provides the advantage of immediate updates to the BrightStor CA-1 tape management catalog when BrightStor CA-Disk for USS releases a tape volume from the archives or increases the expiration date of an archived file. It has no effect during file archival/backup processing since standard BrightStor CA-1 hooks in tape OPEN get the expiration date from system control blocks at that time.

Following are instructions on how to install the BrightStor CA-Disk for USS direct interface to BrightStor CA-1:

1. Assemble module ADSTH012. The module is located in the installation library, and is written in Assembler Language. Use a standard assembly procedure, ensuring that the BrightStor CA-Disk for USS macro library is concatenated before all other maclibs in the SYSLIB statements. Also, make sure that the BrightStor CA-1 macro library is in the SYSLIB statements but after the BrightStor CA-Disk for USS macro library.

If you have installed BrightStor CA-Disk for USS with SMP/E, this assembly should be done using SMP/E in the form of a usermod. Installation library member USERMODA can accomplish this for you. If you did not install BrightStor CA-Disk for USS with SMP/E, sample assemble/link JCL for assembling this module is located in installation library member INSTTMS.

We recommend using Assembler H to assemble BrightStor CA-Disk for USS modules. However, we try to allow source members in the installation library to be assembled with Assembler F. If you use Assembler F, you must add:

```
,AMODE=,SPLEVEL=
```

to the end of the PROC macro statement in your assemble source. If Assembler F cleanly assembles a BrightStor CA-Disk for USS module (condition code 0—no assembly errors) the module is correctly assembled. The problems with Assembler F are in the form of assembly errors (for example, “IF STATEMENT OUT OF ORDER...”) not experienced with Assembler H.

Note: Since the BrightStor CA-1 macros can change with each release of BrightStor CA-1, you must assemble and link the BrightStor CA-Disk for USS module ADSTH012 when you install a new release of BrightStor CA-1.

If a problem arises because some of the needed BrightStor CA-1 modules cannot be found, simply put a PRINT GEN statement in module ADSTH012 and reassemble it. The assembly listing with the expanded macros identifies the BrightStor CA-1 modules that must be found to complete a successful installation.

2. Verify that the module is in the BrightStor CA-Disk for USS load library and there are no unresolved references in the link-edit.

Following are instructions on how to activate the BrightStor CA-Disk for USS direct interface to BrightStor CA-1:

1. Make the BrightStor CA-1 load library available to BrightStor CA-Disk for USS. Either put the BrightStor CA-1 load modules in a LINKLIST library, or put the BrightStor CA-1 load library in the STEPLIB for all BrightStor CA-Disk for USS backup, archive, restore, and IXMAINT tasks.
2. Turn on the BrightStor CA-1 support by specifying sysparm TMSCTLEX with a value of ADSTH025.
3. Decide which of the two methods of scratching tapes BrightStor CA-Disk for USS is to use. BrightStor CA-Disk for USS defaults to just setting the expiration date to the current date. If you want BrightStor CA-Disk for USS to also set the scratch bit, causing BrightStor CA-1 to immediately treat it as a scratch tape, specify sysparm UCC1SCRT with a value of Y.

Following is a list of BrightStor CA-Disk for USS/BrightStor CA-1 interfacing considerations:

We recommend that you run the PATHDELETE command of IXMAINT (for details, see PATHDELETE in the chapter “Index Maintenance”) nightly before the BrightStor CA-1 scratch and clean functions are run. This helps to ensure that BrightStor CA-1 is always current with BrightStor CA-Disk for USS.

If a job abends while writing a tape and you are not using the EDM Interface as your BrightStor CA-Disk for USS interface, BrightStor CA-1 defaults to considering that output tape a scratch. To BrightStor CA-Disk for USS, however, the partial tape is a good tape and must be kept. You should take special precautions to prevent BrightStor CA-1 from marking BrightStor CA-Disk for USS output tapes as scratch tapes after an abend. This can be done manually, or you can make use of one of the exits within the BrightStor CA-1 system.

Note: A BrightStor CA-Disk for USS user has supplied a sample for this exit in member SLI035 of the user mod library. You should consult your local BrightStor CA-1 support staff or Computer Associates directly, if necessary, if you have further questions regarding their exit.

ADSTH017—Interface for TLMS

This module provides support for the EDM interface developed for Release 5.0.4 of TLMS. To enable the support, sysparms TMSCTLEX and DYNEXPDT must be set to the values ADSTH017 and E99365 respectively. The TLMS documentation supplied by Computer Associates must also be performed. Known requirements from Computer Associates for this support include, but are not be limited to, the following:

- The TLMSIPO member in the TLMS PPOPTION library needs to be updated with the following parameters:
 - EDM=DISK – This is the association to the tape owner
 - DSN=SAMSDISK.TAPE.NAME – This is the association to the tape dsname specified in sysparm ARC0NAME, ARCCNAME, MERP1NAM, and so on
 - JOBNAME=SAMDISK1 – This is the optional jobname that determines the eligibility to access the EDM
 - PGM=ADSMI002 – According to Computer Associates, this is the scratching program name, and IXMAINT runs under this program name
 - POOLID=DISKPOOL – This is an optional poolname that restricts this EDM to a TLMS range or a pool of tapes
- To update existing BrightStor CA-Disk for USS tapes so that they fall under the control of TLMS, use the UPV batch command as documented in the TLMS manual. An example of Release 5.0.4 is:


```
UPV NNNNNN,OWNER=DISK
```

where NNNNNN is the tape number being updated to EDM control and DISK is the value specified as the EDM ID in the TLMSIPO member.
- Replace the IBM ARCTVEXT exit with the TLMS ARCTVEXT exit.
- Set the INPUT=YES subparameter CHAIN=NO in the TLMSIPO member.

Note: This information is provided for your convenience. If there are any questions about implementing Computer Associates-TLMS parameters, customers should review their TLMS manual or contact Computer Associates support directly.

TPMGMTX—Tape Management Interface Exit

The module name supplied for this exit receives control from any BrightStor CA-Disk for USS function that is updating the status of an existing archive volume. Typical types of updates are if BrightStor CA-Disk for USS is going to expire or delete archive volumes or if BrightStor CA-Disk for USS is going to update the expiration date of an archive volume.

There are two options possible for this exit. Option 1 is invoked when BrightStor CA-Disk for USS is going to expire or delete an archive volume. Option 2 is invoked when BrightStor CA-Disk for USS is going to alter the expiration date of an archive volume. When option 2 is invoked, a seventh parameter is given. This parameter is the new expiration date that BrightStor CA-Disk for USS is going to reset for the archive volume. This parameter does not exist for option 1.

If you have sysparm TMSCTLEX specified with ADSTH014, (that is, you are using the External Data Manager facility of BrightStor CA-1 as your tape management interface), you should be aware that there is no facility available to change the expiration date within the BrightStor CA-1 database, therefore no processing is done by BrightStor CA-Disk for USS. This user exit is still called with this option, however, to support other tape management interfaces that can exist.

Use sysparm TPMGMTEX to specify the module name to be invoked for this exit. Following is a list of parameters that are passed to TPMGMTEX:

Type	Size	Description
INPUT	H	OPTION: 1 Expire Tape
		2 Reset Expiration Date
INPUT	CL44	Tape DSN
INPUT	H	Number of tape volumes. This number is always one.
INPUT	CL6	Tape volume
OUTPUT	H	Return Code 1 Okay to update
		2 or more Stop update
OUTPUT	H	Reason Code Not used at this time. Can be set to any HW value.
IN/OUT	PL4	New EXPDT – (option 2 only).

UNLOADEX—Process Unloaded Files File Record

The module name supplied for this exit receives control after a record has been selected for unload in the files unload process. The entire record is passed to the module and you can modify any field. If the length of any record is altered, you must ensure that the FILEDEFN table and files file entries conform to the change. The exit can also be used to selectively eliminate records by setting a return code of 2.

Note: When record length is referenced, it means the file-defined record length plus decimal 12; for example, if a UNXINDEX record is passed and its file-defined length is 256, the length passed is 268.

Following is a list of parameters that are passed to UNLOADEX:

Type	Size	Description
IN/OUT	CL8	Subfile Name
IN/OUT	CL256	Files Record (maximum length of 256 bytes)
IN/OUT	HW	Record Length (can be modified)
OUTPUT	HW	Return Code H'1' Unload the record
		H'2' Bypass the record

USERSENQ—ENQ/DEQ Exit

The module name supplied for this exit receives control for each ENQ, DEQ, RESERVE, or RELEASE to be issued. Its intended use is to allow the user to monitor, modify, or replace the normal BrightStor CA-Disk for USS ENQ processing. It does not affect processing specified by the BrightStor CA-Disk for USS sysparms SPFENQQN, SPFENQTY, or SPFENQWT.

A return code set by the exit module indicates what further action BrightStor CA-Disk for USS is to take. Use sysparm USERSENQ to specify the module name to be invoked for this exit. Following is a list of parameters that are passed to USERSENQ:

Type	Size	Description
IN/OUT	CL2	Type of ENQ Request
		CL2'E' ENQ Exclusive
		CL2'S' ENQ Shared
		CL2'C' Change Shared to Exclusive
		CL2'D' DEQ
		CL2'RE' RESERVE and ENQ Exclusive
		CL2'RS' RESERVE and ENQ Shared
		CL2'RC' RESERVE and Change Shared to Exclusive
		CL2'RD' RELEASE and DEQ
		CL2'TE' Tape ENQ
		CL2'TD' Tape DEQ
INPUT	CL44	DSNAME (the resource name)
INPUT	CL6	VOLUME (for RESERVE requests)

Type	Size	Description
OUTPUT	H	Return Code H'0' Skip normal ENQ processing. Do not perform ENQ/DEQ processing.
		H'4' Continue with normal BrightStor CA-Disk for USS ENQ/DEQ processing per the request type in parameter 1
		H'5' Reject the ENQ/DEQ request

This section describes the user-defined members of the BrightStor CA-Disk for USS parameter library (parmlib). You add members to parmlib when you want to modify or tailor the way a particular BrightStor CA-Disk for USS function works; for example, a list of files that you want to exempt from processing.

Purpose

All functions of BrightStor CA-Disk for USS use various control parameters during their execution. These parameters provide many different kinds of information. Some indicate how BrightStor CA-Disk for USS is to operate, some define the format of reports, some define files for which BrightStor CA-Disk for USS is to skip processing, and others provide user-dependent information to BrightStor CA-Disk for USS functions.

These parameter lists are specified as members of a partitioned data set known as the parameter library, or parmlib for short. The parmlib implementation allows BrightStor CA-Disk for USS to better tailor its operation to the requirements of each user. The contents of a parmlib member can easily be changed to reflect the options you want. After they are changed, the next execution of BrightStor CA-Disk for USS uses them.

The library is ready for use as soon as it is loaded from the tape during the installation process. The supplied members are regarded as parameter lists to be used for internal BrightStor CA-Disk for USS system functions and should not be changed. Parameter lists to contain user information are placed in new members of the library. The documentation for each function of the system explains what user-defined members from parmlib it uses and the purpose of each.

Most BrightStor CA-Disk for USS functions that require user-created members indicate specific names to be assigned when the member is created. Some functions allow this name to be overridden by parameters on the BrightStor CA-Disk for USS commands. This allows you to create several different parmlib members for use by the same function. Then you specify the member to be used at execution time.

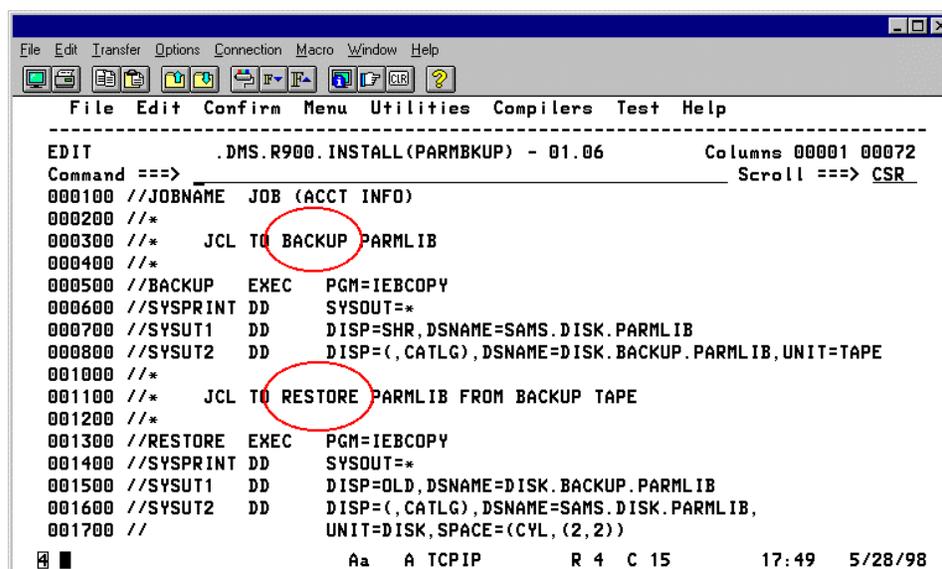
For explanations of common BrightStor CA-Disk for USS parameter lists and the purpose of each, see User PARMLIB Members in this appendix. The documentation for each function of BrightStor CA-Disk for USS further defines the parmlib members used. More information is found there.

Updating Member Data

Since the BrightStor CA-Disk for USS parameter library is a partitioned data set, it can be updated like any other PDS. BrightStor CA-Disk for USS does not provide special facilities to update it. Most installations have online editors to change members of PDSs. The standard IBM-supplied utility for maintenance of PDS members is the IEBUPDTE utility. If you are unsure how to update this library, see your systems programmer.

Backup Considerations

Since the parameter library is a conventional PDS, it can be managed by the same means. Note, however, that if the parameter library is lost, BrightStor CA-Disk for USS restores cannot be done. Therefore it is advisable to occasionally make a backup copy of this data set using the IEBCOPY utility. Following is sample JCL to back up the BrightStor CA-Disk for USS PARMLIB with the IBM IEBCOPY utility. A copy of this JCL is in the BrightStor CA-Disk for USS installation library in member name PARMBKUP.



```
File Edit Transfer Options Connection Macro Window Help
-----
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          .DMS.R900.INSTALL(PARMBKUP) - 01.06          Columns 00001 00072
Command ==>
000100 //JOBNAME  JOB (ACCT INFO)
000200 //*
000300 //*      JCL TO BACKUP PARMLIB
000400 //*
000500 //BACKUP  EXEC  PGM=IEBCOPY
000600 //SYSPRINT DD  SYSOUT=*
000700 //SYSUT1   DD   DISP=SHR,DSNAME=SAMS.DISK.PARMLIB
000800 //SYSUT2   DD   DISP=(,CATLG),DSNAME=DISK.BACKUP.PARMLIB,UNIT=TAPE
001000 //*
001100 //*      JCL TO RESTORE PARMLIB FROM BACKUP TAPE
001200 //*
001300 //RESTORE  EXEC  PGM=IEBCOPY
001400 //SYSPRINT DD  SYSOUT=*
001500 //SYSUT1   DD   DISP=OLD,DSNAME=DISK.BACKUP.PARMLIB
001600 //SYSUT2   DD   DISP=(,CATLG),DSNAME=SAMS.DISK.PARMLIB,
001700 //          UNIT=DISK,SPACE=(CYL,(2,2))
-----
Aa  A TCP/IP          R 4  C 15          17:49  5/28/98
```

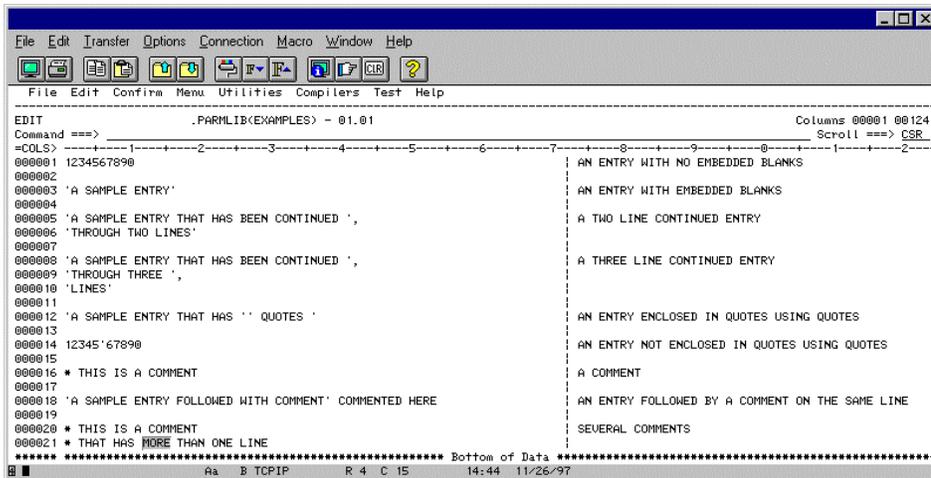
PARMLIB Specifications

Following are the syntax rules for entries in members of parmlib:

- A single entry can be from one to seventy-one characters long and can be continued if longer.
- An entry must be enclosed in quotes if it contains embedded blanks or is continued. Any entry can be enclosed in quotes.
- To continue an entry, add a comma after the ending quote.
- To have quotes within an entry enclosed in quotes, use two quotes.
- To include a comment, add an asterisk in column one. For a comment on a command line, it should follow one or more blanks after the entry.

Examples

Following is sample syntax for PARMLIB entries:



Special Rules for POOLDEFS

The POOLDEFS member defines up to 150 pool names to be used by the archive/backup functions by associating a poolname with the name of the archive data set being created. If the data set is being created on tape, the pool name defines a tapepool that must exist within the BrightStor CA-Disk for USS Files Data Set. If the data set is being created on disk (archival to disk), the pool name defines a diskpool that must exist as a member in the parmlib data set.

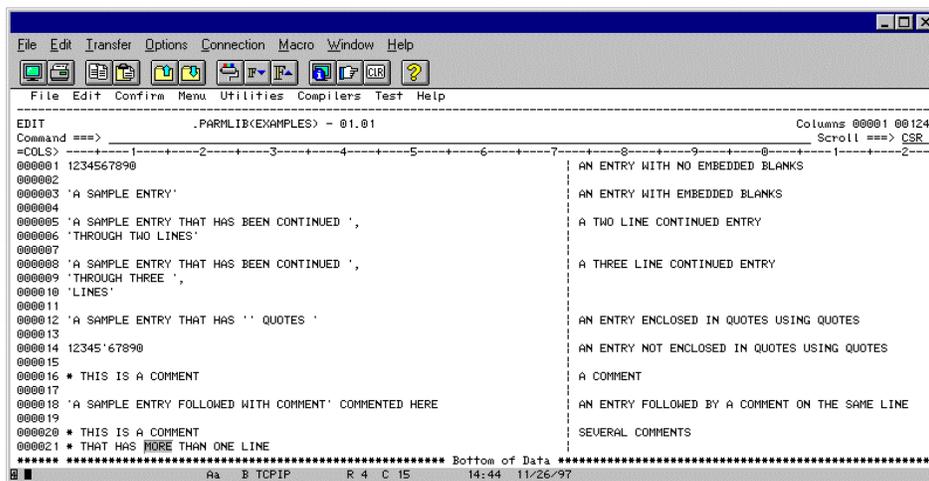
Each entry in POOLDEFS has the following format. Note that it must be enclosed in quotes because of the embedded blank between the two names.

```
'POOLNAME DATASETNAME'
```

POOLNAME is a pool name that must be eight characters in length and DATASETNAME is the file name or pattern, 1 to 44 characters in length, which is to be mapped to the specified pool.

PARMLIB Syntax Validation Utility

This utility validates the syntax of the specified members or all members in the parmlib data set. A copy of the execution JCL is in the installation library in member PARMVALD. Following is sample JCL of installation member PARMVALD:



User PARMLIB Members

This section provides a list of user PARMLIB members and their descriptions.

DASD Pool List

The member name of the DASD pool list is DASDPOOL. It is used by the RECOVER command and RECOVER function, and the RESTORE command and RESTORE function.

Description

Entries are listed in this member to define one or more pools of DASD volumes that share a common role. This member is used by the RESTORE function to obtain an alternate volume for allocation when the file being restored cannot be placed on the target volume. It is also used when the POOL parameter is specified on the RESTORE command.

The DASD pool list is a single member in the parmlib data set. Each entry contains the name of the volume to be associated with a DASD pool and the pool name. Multiple volumes can be associated with the same pool name by having multiple entries with the same pool name and different volume serials. Using different pool names on groups of entries can generate multiple DASD pools. Note that all entries comprising a pool must be grouped together.

You can define up to 60 different DASD pool names. Sysparm DASDVMAX specifies the total number of volumes that can be included in the defined pools (the sum of volumes in all different pools). This is also the maximum number of volumes that can be specified for a single pool name. The default value is 100; for example, one pool of 100 volumes, or two pools of 50 volumes, and so on.

Format

The format is 'VVVVVVPPPPPPPP' where:

- VVVVVV is the six character volume serial of a volume in a pool. If it is fewer than six characters in length, trailing blanks must be provided and the entry must be enclosed in quotes in positions 1 through 6.
- PPPPPPP is the one to eight character alphanumeric pool name. This name must be identical for all volume entries comprising a specific pool in positions 7 through 14.

Example

The following entries define two pools, a volume pool named RJEWORK and a volume pool named TSODATA:

```
'VOL003RJEWORK'  
'VOL004RJEWORK'  
'VOL006RJEWORK'  
'PACK01TSODATA'  
'PACK12TSODATA'  
'PACK15TSODATA'  
'PACK20TSODATA'
```

Exclusion Table for Data Compression

Following is a description and example of the exclusion table for data compression.

Description

Place the file names and patterns that are to be excluded from data compression during archive or back up into this member. A match causes data compression to be bypassed.

Format

The format is 'DDDD....' where:

- DDDD... is a 1- to 44-character file name or BrightStor CA-Disk for USS pattern file name located in positions 1 through 44.

Example

The following entries cause files with a prefix of "SYS1" and files with only three nodes where the third node is "PRINT" to be excluded from compression processing:

```
'SYS1/'  
'*.*.PRINT'
```

Inclusion Table for Data Compression

The member name of the inclusion table for data compressions is specified in sysparm CDEXCTBL_____. It is used by the ARCHIVE and BACKUP functions.

Description

Place the file names and patterns that are to be included for data compression during archive or backup into this member. A match compresses the file.

Format

The format is 'DDDD....' where:

- DDDD..... – A 1- to 44-character file name or BrightStor CA-Disk for USS pattern file name located in positions 1 through 44.

Example

The following entries compress files with a prefix of “SYS1” and files with only three nodes where the third node is “PRINT”:

```
'SYS1/'  
'*.*.PRINT'
```

Exemption Lists—Generalized

The member of parmlib is used by the EXCLUDE command (DSCL).

Description

Place the file names and patterns to be exempt from BrightStor CA-Disk for USS volume scan processing in this member of parmlib. When BrightStor CA-Disk for USS is searching DASD volumes for files to process, it ensures that the file name is not in this list before attempting processing for it. If the file name is in this list, it is exempted from processing.

Format

The format is 'DDDD.....' where:

- DDDD..... is a 1- to 44-character file name or BrightStor CA-Disk for USS pattern file name located in positions 1 through 44.

Example

The following entries cause files beginning with the five characters “SYS./”, and files with a third node level of “PRINT” to be excluded from BrightStor CA-Disk for USS processing:

```
'SYS /'  
'*.*.PRINT'
```

System Parameter List

The member name of the system parameter list is SYSPARMS. It is used by all BrightStor CA-Disk for USS functions.

Description

The BrightStor CA-Disk for USS system parameter list is a user-specified list of options used by BrightStor CA-Disk for USS during its operation. They indicate how the user wants BrightStor CA-Disk for USS to operate in their environment. For a complete explanation of these parameters, see System Parameters in Appendix E “Sysparms.”

Each entry in the SYSPARMS member specifies a separate option to BrightStor CA-Disk for USS. System parameters have eight-character names that identify them. Each entry contains the character name followed by the option specification. Option specification is variable, depending on the definition of the option.

Format

The format is 'XXXXXXXXVVVV....' where:

- XX.XX is the system parameter name located in positions 1 through 8.
- VVVV.... is the system parameter information located in positions 9 through nn.

Example

The following entries define two system parameters. This parameter indicates that the default retention periods for implicit archival processing is 15 days.

```
'RETRETPD0015'
```

User-Specified Condition Codes

The member name of user-specified condition codes is CCSET. It is used by all BrightStor CA-Disk for USS functions.

Description

This member overrides or does not override the condition code generated by BrightStor CA-Disk for USS. If the user chooses to override the BrightStor CA-Disk for USS-generated condition code, one of four possible alternative condition codes can be selected and assigned per message number. The first entry in this member is required. The format is as follows:

```
$FLG FORMAT
```

Format

The format is '\$FLGXyz' or 'nnnnbxy' where:

\$FLG	\$FLG		
X	BLANK		
y	BATCH FLAG	B	display message under BATCH
		BLANK	do not display message under BATCH
z	TSO FLAG	T	display message under TSO
		BLANK	do not display message under BATCH

or

nnnn 4-digit BrightStor CA-Disk for USS message code. Located in positions 1 through 4.

b blank, located in position 5.

x O override Use BrightStor CA-Disk for USS condition code. Located in position 6.

N no override

y override condition codes (located in position 7):

G 0 good

I 4 informational message

R 8 resource error

E 12 error

C 16 command error

Example

The following entry overrides the BrightStor CA-Disk for USS-generated condition code for message 3179 with a user-specified condition code of 16:
'3179 oc'

JCL DD Statements

This section provides an explanation of all DD statements within the PROCs. All DD statements (except STEPLIB) are arranged alphabetically. This allows overriding DD statements to be used much more easily, since all you need to remember is to provide them in alphabetical order. Following is a list of symbolic parameters and recommendations, where Rnn represents the release number that is installed. For example, Release 1.1 or 9.9 is represented as R11 or R99, respectively.

Symbolic PARMS	Recommended Defaults	Explanations
P	SAMS.UNIX.Rnn.INSTALL	Library for control statements
Q	SAMS.UNIX.Rnn	Qualifier for data sets
S	*	Sysout class
W	SYSDA	Work file default unit name

Note: These default values are recommended, but can be different because of your local naming conventions. Examine the actual procedures as installed and being used at your location and contact your local BrightStor CA-Disk for USS support personnel if you need assistance.

DD Statement Explanations

The next section provides a list of DD statements and their explanations.

//ARCHIVEC

Provide this DD statement only when you want duplicate backup copies of the archive tapes. It can be either omitted or dummied out if you want duplicate tapes suppressed.

DCB attributes are automatically copied from the ARCHIVE0 DD statement. It can also be dynamically allocated based upon the values of sysparms ARCCTYPE and ARCCNAME.

//ARCHIVES

This DD statement is used by BrightStor CA-Disk for USS to allocate a tape drive when read access of the archives is necessary. The RESTORE and RECOVER functions access the BrightStor CA-Disk for USS archives for read-only.

//ARCHIVE0

This DD statement references the output medium upon which archived files are to be stored. It can be provided in the JCL, or it can also be dynamically allocated based upon the values of sysparms ARC0TYPE and ARC0NAME. If the expiration date is not supplied by the label parameter or if its value is zero, the highest expiration date from all of the files to be written on it is placed in the label as the expiration date.

//COMMANDS

This is a temporary work file used by the RESTORE function to order the explicit RESTORE requests. This is done such that the archive tape is mounted only after, and processed only in the forward direction.

//CMDPRINT

This is a transaction listing of the input command stream and any command processing messages.

//COPYFILE

This is a data set containing a list of BrightStor CA-Disk for USS archive volumes that are to be copied.

//DISPHOLD

This is a temporary data set used to suspend data set disposition until a threshold value can be verified. This allows certain BrightStor CA-Disk for USS functions to operate in a “fail safe” mode.

//DISPWORK

A disposition record is written in this data set for subsequent reference in determining the re-catalog and scratch action after successful tape creation.

//DSNEXPDT

This file is required (live or simulate mode) only if:

- Files that satisfy the selection criteria specify ARCHIVE as one of the disposal actions.
- No expiration date is provided on the // ARCHIVE0 DD statement.

The file is used to determine the highest expiration date from all the files being archived. This date is then assigned as the expiration date for the entire archive volume.

The file is used much like an internally generated deferred request. All disposal records for any archives to be done are temporarily stored here. The highest expiration date is found, the archive volume is opened, and the disposal records are then processed to cause the actual archiving of the files.

//DSNIIN

This DD statement is used as a SORTIN data set.

//DSNIOUT

This DD statement is used as a SORTOUT data set.

//DSNIWK01 ... WK03

These DD statements are used as SORT work files.

//DSCBS

This DD statement is used as a SORTIN data set. It contains the format-1 DSCBs selected by the billing function.

//FILEIN

This DD statement is used as a SORTIN data set.

//FILEOUT

This DD statement is used as a SORTOUT data set.

//FILES

This DD statement allocates the BrightStor CA-Disk for USS Files Data Set. It contains subfiles for the UNXINDEX, ARCHVOLS, and other files.

//FILEWK01 ... WK03

These DD statements are used as SORT work files.

//FILPRINT

This DD statement defines a SYSOUT data set BrightStor CA-Disk for USS uses to produce a files status report.

//INDEXREC

A temporary work file used by the RECOVER function to order the UNXINDEX records. This is done such that an ARCHIVE tape is mounted only after, and processed only in the forward direction.

//MFILES

This DD statement allocates one or more BrightStor CA-Disk for USS Files Data Sets for auto restore, restore, and deferred restore processing. Each Files Data Set allocated in the concatenation is searched to find the first UNXINDEX record for a file to be restored. The FILES DD statement is not used if the MFILES DD statement is allocated.

//MSGPRINT

Execution time messages are directed to this data set.

//PARMLIB

This DD statement defines the BrightStor CA-Disk for USS parameter library. It contains internal system parameter lists as well as any user-defined parameters.

//REPORTS

This ddname is used to reference the data set that receives the detail records in any report-generating step, and the data set containing the sorted records in the print step.

//SEQFILES

This data set contains records from a FILES unload and is used for the FILES reload.

//SMFDATA

This ddname is used in the SMFRPT procedure to reference a sequential data set containing SMF type 14 and 15 records. These records are used to produce the Utilization by Date and Time report.

//SMSPRINT

BrightStor CA-Disk for USS generates diagnostic messages explaining the processing done for the BrightStor CA-Disk for USS function as it pertains to the IBM DFSMS. These messages are written out to this data set.

//STEPCAT

The STEPCAT DD statement should be used only to override the normal catalog search techniques.

//SYSPARMS

The SYSPARMS DD statement can be used to supply overriding sysparm values at execution time.

//SYSPRINT

A formatted listing of all actions taken during the run is directed to this data set.

//TAPEPULL

This data set contains a copy of the generated "tape pull" list and restricts BrightStor CA-Disk for USS to only those tapes.

Index

A

abends
 codes, 1-11
 tape status, Appendix D-14

allocation considerations, Appendix C-1

ARCHIVE, 5-2, 5-3
 graceful shutdown, 2-3

ARCHIVE command example, 5-5

B

Backup
 abends, Appendix D-14
 assigning expiration dates, Appendix D-7
 cataloging the tapes, Appendix C-2
 commands available, 5-3
 considerations, Appendix D-3
 execution examples, 5-4
 expiration dates, Appendix C-2
 integrity, Appendix D-12
 media used, Appendix D-4
 medium used, Appendix D-3
 naming the archive, Appendix D-5
 overwriting valid tapes, Appendix D-13
 primary volumes, 6-2
 producing duplicate tapes concurrently, Appendix C-2
 QUIESCE parameter, 5-4
 sample report, 5-5
 security, Appendix D-12
 system parameters, Appendix D-2
 tape protection, Appendix D-12

BACKUP, 5-1, 5-3
 graceful shutdown, 2-3

BACKUP command, 5-3

BACKUP command and parameters, 5-2

BACKUPCC
 graceful shutdown, 2-3

basic conventions and requirements, 1-3

C

catalog status controlled tapes, Appendix C-10

cataloging the tape archives, Appendix C-2

CCSET, Appendix G-8

Commands
 BACKUP, 5-3
 FIND, 2-13
 LISTU, 8-10
 LISTV, 8-13
 MERGE, 7-11
 PATHDELETE, 8-1
 PURGE, 8-5
 REBUILD, 8-7
 RECOVER, 6-12
 RESTORE, 6-3
 UREPORT, 4-1
 VOLDELETE, 8-4

condition codes
 CCSET member of PARMLIB, Appendix G-8
 DSCL, 2-4
 FMS, 6-7
 MERGE, 7-11
 RESTORE, 6-2
 user specified, Appendix D-15

configuration options
 setting ISPF, 9-16

considerations

- allocation type considerations, Appendix C-1
- backing up the PARMLIB dataset, Appendix G-2
- backup processing, Appendix D-3
- files dataset, 1-2
- ISPF general, 9-2
- OS/390 UNIX, 1-2
- pattern matching, 1-2
- tapepool processing, Appendix C-3

control statements, 1-4

controlling tapes

- catalog status, Appendix C-10
- EDM interface, Appendix C-7
- expiration date, Appendix C-9
- TLMS/EDM, Appendix C-11

conventions and requirements, 1-3

COPY

- graceful shutdown, 2-3

copy archive volumes, 6-2

D

DASDPOOL, Appendix G-4

data set

- name validation, 1-6
- names and patterns, 1-5

date formats, 1-7

DD statement explanations, Appendix H-1

DDNAMES, Appendix C-1

DELETE command, 2-3

DISP, 5-4

display

- ISPF explicit file record, 9-18

displaying

- ISPF path list, 9-24

DSCL

- condition codes, 2-4
- EXCLUDE rules, 2-10
- FIND rules, 2-8
- overview, 2-1
- processing considerations, 2-13
- SCAN rules, 2-8
- SELECT rules, 2-10

E

EDM interface, Appendix C-7

EXCLUDE parameter, 2-15, 6-8

EXCLUDE rules, 2-10

exclusion table for data compression, Appendix G-6

execution

- for backup, 5-4
- LISTU, 8-11
- LISTV, 8-14
- PATHDELETE, 8-3
- PURGE, 8-7
- VOLDELETE, 8-5
- for recover, 6-13
- for restore, 6-5

exemption lists – generalized, Appendix G-7

expiration controlled tapes, Appendix C-9

expiration dates

- in archive datasets, Appendix D-7
- in tapes, Appendix C-2

EXPIRE command

- graceful shutdown, 2-3

explicit file record display

- ISPF, 9-18

explicit file restore

- ISPF, 9-20

explicit path recovery

- ISPF, 9-22

explicit restore, 6-1

F

file list panel

- ISPF, 9-10

file record display

- ISPF explicit, 9-18

file restore

- ISPF explicit, 9-20

Files Data Set, 1-2
 formatting of, Appendix B-4
 management, Appendix B-4
 performance, Appendix D-2
 reload utility, Appendix B-8
 RELOAD utility, Appendix B-11
 restrictions, Appendix B-2
 status report, Appendix B-5
 subfiles, Appendix B-2
 unload utility, Appendix B-8, Appendix B-9

FIND command, 2-13
 SYNTAX, 2-14

FIND rules, 2-8

flag fields, 8-13

FMS
 condition codes, 6-7

full backup, 5-1

G

general considerations
 ISPF, 9-2

general information
 explicit restore, 6-1
 FMS recover, 6-7
 maintenance commands, 8-1
 parmlib dataset, Appendix G-1
 reporting, 8-10

General Information
 MERGE Command, 7-1

general panel overview
 ISPF, 9-3

gregorian date formats, 1-7

I

implicit restore, 6-7

inclusion table for data compression, Appendix G-6

incremental backup, 5-1

index maintenance
 execution examples, 8-3, 8-5, 8-7, 8-11, 8-14
 flags, 8-13
 LISTU command, 8-10
 LISTV command, 8-13
 PATHDELETE command, 8-1
 PURGE command, 8-5
 REBUILD command, 8-7
 reporting commands, 8-9
 VOLDELETE command, 8-4

integrity in the archives, Appendix D-12

ISPF
 setting configuration options, 9-16
 explicit file record display, 9-18
 explicit file restore, 9-20
 explicit path recovery, 9-22
 file list panel, 9-10
 general considerations, 9-2
 general panel overview, 9-3
 opening panel, 9-3
 path list displaying, 9-24
 path list panel, 9-6
 recover paths, 9-28
 restore from list, 9-32

J

JCL requirements, 1-11

Julian date formats, 1-8

L

learning to use BrighStor CA-Disk for USS, 1-3

list
 ISPF restore from, 9-32

list displaying
 ISPF path, 9-24

list panel
 ISPF file, 9-10
 ISPF path, 9-6

LISTU command, 8-10

LISTV command, 8-13

LISTV flags, 8-13

M

maintenance commands, 8-1

managing the files dataset, Appendix B-4

media used in backup, Appendix D-4

medium used in backup, Appendix D-3

memory requirements, 1-11

MERGE

- Abend/Restart Considerations, 7-7
- Command and Parameters, 7-11
- condition codes, 7-11
- Implementation Concerns, 7-4
- Operation concerns, 7-5
- Overview, 7-1
- Processing Logic, 7-9
- Report, 7-15
- Simulate mode considerations, 7-8
- Specifying Merge Output Options, 7-4
- Specifying What to Merge, 7-2
- Syntax, 7-12
- Tapepull processing, 7-6

messages, 1-10

midnight, 5-2

MOVE

- graceful shutdown, 2-3

multiple files datasets, 6-1

N

naming the backup tapes, Appendix D-5

O

opening panel

- ISPF, 9-3

options

- setting ISPF configuration, 9-16

Overriding Sysparms Instream, Appendix E-2

overview

- ISPF general panel, 9-3
- MERGE Command, 7-1
- OS/390 UNIX, 1-1
- parmlib dataset, Appendix G-1

P

panel

- ISPF file list, 9-10
- ISPF opening, 9-3
- ISPF path list, 9-6

panel overview

- ISPF general, 9-3

panels

- using the, 9-15

parameters

- EXCLUDE, 2-15, 6-8
- QUIESCE, 5-4
- SCAN UNIX, 2-12
- SCAN UNXINDEX, 6-8
- SELECT, 2-15, 6-8
- SET MODE, 2-11, 6-7
- UATT, 4-2
- USPC, 4-2

PARMLIB

- backup considerations, Appendix G-2
- CCSET member, Appendix G-8
- DASDPOOL, Appendix G-4
- exclusion table for data compression, Appendix G-6
- exemption lists, Appendix G-7
- inclusion table for data compression, Appendix G-6
- POOLDEFS, Appendix G-3
- purpose of, Appendix G-1
- specifications, Appendix G-3
- syntax validation utility, Appendix G-4
- SYSPARMS member, Appendix G-7
- updating member data, Appendix G-2
- user members, Appendix G-4

path list displaying

- ISPF, 9-24

path list panel

- ISPF, 9-6

path recovery

- ISPF explicit, 9-22

PATHDELETE command, 8-1

paths

ISPF recover, 9-28

pattern matching, 1-2

performance tips, Appendix D-2

files dataset, Appendix D-2

POOLDEFS, Appendix G-3

primary archive volumes, 6-2

PURGE command, 8-5

Q

QUIESCE parameter, 5-4

R

REBUILD command, 8-7

record display

ISPF explicit file, 9-18

recover

EXCLUDE parameter, 6-8

execution examples, 6-13

SCAN UNXINDEX parameter, 6-8

SELECT parameter, 6-8

SET MODE parameter, 6-7

RECOVER command, 6-12

recover paths

ISPF, 9-28

recovery

ISPF explicit path, 9-22

RELEASE

graceful shutdown, 2-3

Reports

EXCLUDE parameter, 2-15

MERGE Command, 7-15

SCAN UNIX, 2-12

SELECT parameter, 2-15

SET MODE parameter, 2-11

UATT parameter, 4-2

USPC parameter, 4-2

requirements, 1-3, 1-11

restore

commands available, 6-3, 6-12

condition codes, 6-2

execution examples, 6-5

explicit mode, 6-1

FMS recover, 6-7

implicit mode, 6-7

ISPF explicit file, 9-20

multiple files datasets, 6-1

RESTORE command, 6-3

restore from list

ISPF, 9-32

restrictions

Files Data Set, Appendix B-2

S

sample BACKUP report, 5-5

SCAN rules, 2-8

SCAN UNIX parameter, 2-12

SCAN UNXINDEX parameter, 6-8

security

archive tape protection, Appendix D-12

overwriting valid tapes, Appendix D-13

security in the archives, Appendix D-12

SELECT command, 2-15

SELECT parameter, 2-15, 6-8

SELECT rules, 2-10

separator pages, 1-10

SET MODE parameter, 2-11, 6-7

setting ISPF configuration options, 9-16

storage (memory) requirements, 1-11

subfiles of the files dataset, Appendix B-2

syntax, 5-4

SYNTAX

FIND command, 2-14

MERGE, 7-12

SELECT command, 2-15

SYSPARMS, Appendix G-7

SYSPARMS Member in PARMLIB, Appendix E-1

system parameter definitions, Appendix E-5

system parameter names reserved for users, Appendix E-2

system parameters, Appendix E-3

- backup, Appendix D-2
- definitions, Appendix E-5
- lines per page, Appendix D-2
- list of sysparms available, Appendix E-3
- overriding, Appendix E-2
- separator pages, Appendix D-2
- SYSPARMS member of PARMLIB, Appendix G-7

T

tailoring to user needs, 1-7

tape

- status after abends, Appendix D-14

tapepool processing

- considerations, Appendix C-3
- MERGE, 7-6

terminology, 1-4, 5-1

TLMS/EDM, Appendix C-11

U

UATT parameter, 4-2

UNIX file system, 1-2

UREPORT, 4-1

user abend codes, 1-11

user exits, Appendix F-2

using the panels, 9-15

USPC parameter, 4-2

utilities

- PARMLIB syntax validation, Appendix G-4
- RELOAD utility, Appendix B-8, Appendix B-11
- UNLOAD utility, Appendix B-8, Appendix B-9

V

validation

- of dataset name, 1-6

VBACKUP

- graceful shutdown, 2-3

VOLDELETE command, 8-4

Glossary

Archival

Provision for the storage of data at a lower cost, where it is not directly accessible to using programs. It includes the facility for restoring the data to the original medium on a demand basis.

Archvols

The tape volumes or disk data sets that contain the data sets that have been archived, and the index to them that is kept in the Files Data Set.

Auto-Restore

A BrightStor CA-Disk for USS process by which data sets that have been archived are restored automatically if they are required by an application at a later time. The data set can be restored by an online job or a batch job. This allows the DASD manager to keep free space available, yet gives the user access to data sets that are referenced infrequently. Data sets that have been archived to disk rather than to tape can be auto-restored with no operator intervention.

Back Up

A process by which a copy of a DASD data set is made, usually on tape, for the purpose of allowing the data set to be restored in the event of loss or damage to the original data set. Back up differs from the archive process, in that the data sets are copied, but not scratched or uncataloged.

BrightStor CA-Disk for USS backup processing provides for incremental back up, which limits copying to those data sets that have been changed since the last back up was taken. As each changed data set is identified, the backup copy is created and indexed, and then the change bit is turned off. To protect the data sets that are not changed, they are backed up regularly, such as after a week. For more information, see Full Pack Back Up.

Compress

The term “compress” has three different meanings in BrightStor CA-Disk for USS:

- In PDS Compress, the term “compress” describes the process of eliminating “dead space” in a PDS, or shrinking the size of the data set or its directory, by writing the PDS to a work file, calculating reallocation requirements, then copying it back.
- The second usage applies to archive or backup processing. When data sets are copied to an archive medium, only active and used data blocks are read, thus resulting in PDS, ISAM, and VSAM data sets being copied to the archive medium in compressed format. Subsequent restore or volume recovery results in the consolidation of several extents into one, a process sometimes referred to as “defragmentation,” and also the compression/reorganization of the data set.
- The third usage also applies to archive or backup processing, and is sometimes referred to as “compaction.” When data is being copied to an archive media, the data is analyzed and encoded (encrypted) into a new format that occupies less space. When the data set is restored, the data is automatically expanded (or de-compressed) into its original format.

Data set index—DSNINDEX

A file containing a record for each data set that has been archived and has not yet expired. Separate records are created for each version of a data set that has been archived more than once. The existence of the data set index makes it possible to restore an archived data set by referencing only its name.

UNIX data set index—UNXINDEX

Same as data set index above, only for UNIX files.

Deferred processing

A means by which requests for certain actions are recorded and saved for later processing (queued processing). This is usually done to allow common requests to be grouped and processed in bulk in a more efficient manner, or to provide greater integrity by ensuring that certain actions are complete before other actions begin.

Differential Backup (DIFF)

This is a term instructing BrightStor CA-Disk for USS to back up all members that have changed since the last FULL backup, including any member previously backed up with the INCR option.

Dispose processing

The actual execution of those requests that at some prior time were deferred (queued); that is, deferred actions are disposed of at a later time.

DSCB—data set control block

A record maintained by the operating system in a special DASD data set for each data set stored on a DASD volume. It contains attributes such as record format, block size, and record length, as well as information about space occupied.

DSCL—Data storage command language

A set of BrightStor CA-Disk for USS commands that provide a comprehensive, efficient, common means to process VSAM and non-VSAM data sets. These commands are preprocessed by BrightStor CA-Disk for USS to group all overlapping requests to allow all actions to be performed during a single pass of VTOCs or catalogs.

Explicit Processing

Any BrightStor CA-Disk for USS processing in which the data sets to be processed are not found by scanning either VTOCs or catalogs, but by processing a list of explicit names that either the user has provided, or have been extracted from the catalog. For more information, see "Implicit Processing."

File (first use)

A data set that has been archived and now exists as one of numerous internal files to the larger archive data set. Such a file (representing a single archived data set) must be restored by a BrightStor CA-Disk for USS utility function before an application program can reference it.

File (second use)

A set of logically related records maintained by BrightStor CA-Disk for USS as a subfile of a single BDAM data set. The entire BDAM data set is referred to as the Files Data Set, while each subfile is usually referenced by its specific file name, such as the data set index file. Each file (subfile) is defined by an entry in the FILEDEFN member of parmlib.

Fingerprint

A fingerprint is a BrightStor CA-Disk for USS hash count of a PO data set. It is stored in a KSDS cluster known as the Site Master.

Full Pack Back Up

BrightStor CA-Disk for USS uses the volume-level back up method and is invoked by the VBACKUP command, making use of DSCL. Use the VRECOVER command to restore data.

Implicit Processing

Any BrightStor CA-Disk for USS processing in which the data sets selected for (or excluded from) processing are determined by scanning the input source, such as VTOCs or catalogs, from beginning to end, and applying user-specified criteria to determine what is or is not to be processed. The criteria can include factors such as last-use date, expiration date, create date, catalog status, block size, or being listed in a table. BrightStor CA-Disk for USS permits multiple selection or exclusion criteria to be specified.

Incremental Backup (INCR)

This is a SAMS>Select term instructing BrightStor CA-Disk for USS to back up all members that have changed since the last FULL, DIFF, or INCR backup.

Merge

The redistribution of archived data sets residing within several archive data sets (on tape or disk) to a smaller number of new archive data sets. Expired data sets are deleted, and unexpired data sets are sorted and grouped to use the archive media more efficiently.

Migration

The movement of data from one type of storage medium to another without restricting accessibility to other programs. Reasons for migration include obtaining a lower-cost storage medium and changing the grouping of data.

MLA

Multi-Level Alias

PARMLIB—Parameter Library

A standard partitioned data set containing both system-supplied members and user-supplied members that control various aspects of BrightStor CA-Disk for USS processing.

Prefix or Group

A set of data sets or members that is identified with a common name prefix. A group name is indicated by placement of a slash (/) immediately following the last common character in a data set or member name. All data sets or members that have names that match the number of characters preceded by the slash (/) are considered part of that group.

Restore

The process by which an individual data set is restored (that is, copied from a backup or archive copy) to a DASD device. BrightStor CA-Disk for USS permits restore processing without preallocation for ISAM, VSAM, PS, PO, and DA data sets as well as for OS CVOL catalogs. Individual members of a PDS can also be restored. For more information, see "Recover."

Recover

The process by which a group of data sets, an entire volume, or a group of volumes is recovered (that is, copied from a backup or archive copy) to DASD. BrightStor CA-Disk for USS recover processing provides for the same capabilities as restore processing, plus permits additional controls to be specified for the greater flexibility necessary for volume recovery.

Retention Control

An automated means by which BrightStor CA-Disk for USS determines which data sets should be retained on disk and which ones should be removed.

Stand-Alone Restore

Stand-Alone Restore is a BrightStor CA-Disk for USS program (known as DMSSAR), provided as a Selectable Unit that is designed to restore an entire DASD volume without the presence of the MVS operating system. It runs under the IBM/370, IBM/XA, and IBM/ESA instruction sets and runs under VM as a guest operating system. The Stand-Alone Restore Selectable Unit is documented in the *Stand-Alone Restore Guide*.

- In order to provide the ability to perform a stand-alone restore, BrightStor CA-Disk for USS has developed volume-level processing (as opposed to its normal data set-level processing mode) for back up and recover functions.

SYSPARM—System Parameter

A means by which BrightStor CA-Disk for USS can be tailored and implemented to meet the unique processing needs of each installation. Sysparms control various aspects of each BrightStor CA-Disk for USS function. The default values are in effect unless they are overridden either temporarily (through overriding DD statements) or permanently (through specification in the SYSPARM member of the parmlib data set).

Systems Managed Storage

The ability of an installation to specify the storage management rules and methodologies needed for their unique installation, and to then automate these within the system itself. Systems Managed Storage not only removes most, if not all, of the clerical effort and intervention that has historically been required, but also provides the installation with control and flexibility in specifying its own rules and methodologies. If this second quality is overlooked, Systems Mismanaged Storage can result. The ability to manage and understand the automated system is of equal, if not greater, importance.

Volume-Level Processing

BrightStor CA-Disk for USS normally processes at the data set level. However, in order to provide the capability to perform Stand-Alone Restore, BrightStor CA-Disk for USS can perform backup and recover processing at the volume level.

- The volume-level back up function (VBACKUP) is invoked through DSCL. It produces a backup tape or tapes containing an entire DASD volume, starting at cylinder 0, track 0, and continuing through all used cylinders.
- The volume-level recover function is invoked under the RECOVER procedure, but through the VRECOVER command. When specified at recovery time, it overlays the entire volume in extent track order (that is, from cylinder 0 to the end of the used cylinders).

VTOC—Volume Table of Contents

The data set on each DASD volume that contains a record called a Data Set Control Block (DSCB) for each data set stored on that volume. It also carries records that are used to account for free space.