Guardian Disk and Tape Utilities Reference Manual

Abstract

This publication describes the disk and tape utilities BACKCOPY, BACKUP, DCOM, DSAP, PAK/UNPAK, and RESTORE for the HP NonStop™ operating system.

Product Version

Utilities G06

Supported Release Version Updates (RVUs)

This publication supports J06.03 and all subsequent J-series RVUs, H06.03 and all subsequent H-series RVUs, and G06.00 and all subsequent G-series RVUs, until otherwise indicated by its replacement publications.

Part Number	Published
523325-012	March 2010

Document History

Part Number	Product Version	Published
523325-008	Utilities G06	June 2005
523325-009	Utilities G06	October 2005
523325-010	Utilities G06	May 2008
523325-011	Utilities G06	December 2009
523325-012	Utilities G06	March 2010

Legal Notices

Printed in the US

Guardian Disk and Tape Utilities Reference Manual

Index | Figures | Tables

2. BACKCOPY (continued)

3. BACKUP

3. BACKUP (continued)

4. Disk Compression (DCOM)

5. Disk Space Analysis Program (DSAP)

6. PAK/UNPAK

7. RESTORE

7. RESTORE (continued)

7. RESTORE (continued)

Figures

Tables

What's New in This Manual

Manual Information

Abstract

This publication describes the disk and tape utilities BACKCOPY, BACKUP, DCOM, DSAP, PAK/UNPAK, and RESTORE for the HP NonStop™ operating system.

Product Version

Utilities G06

Supported Release Version Updates (RVUs)

This publication supports J06.03 and all subsequent J-series RVUs, H06.03 and all subsequent H-series RVUs, and G06.00 and all subsequent G-series RVUs, until otherwise indicated by its replacement publications.

Part Number	Published
523325-012	March 2010

Document History

Part Number	Product Version	Published
523325-008	Utilities G06	June 2005
523325-009	Utilities G06	October 2005
523325-010	Utilities G06	May 2008
523325-011	Utilities G06	December 2009
523325-012	Utilities G06	March 2010

New and Changed Information

Changes to the 523325-012 manual:

- Added the following SQL/MX manuals in the Related Documentation on page xiv:
  - *SQL/MX Installation and Management Guide*
  - *SQL/MX Programming Manual for C and COBOL*
- Added a new guideline for the PARTONLY option on page 7-30.

Changes to the 523325-011 Manual

- Updated CATALOGFILES on page 3-16.

- Updated the guideline to run DCOM on page 4-7.
- Updated the file limit supported by DCOM and DSAP on a single disk volume on page 4-8 and page 5-16, respectively.
- Update the note on page 5-1.
- Updated Figure: DDL Format of the Permanent Work File on page 5-35.
- Added the `-pri PAK Run` option in Table 6-1 on page 6-2 and on page 6-5.
- Updated the Considerations section on page 6-7.
- Added a new RESTORE option, NOPURGEUNTIL, on page 7-5 and its description on page 7-22.
- Updated Restoring Enscribe Files on page 7-45.
- Added an example for the NOPURGEUNTIL option on page 7-47.
- Updated error 7836 on page A-68.
- Updated an error message on page C-7.
- Added a new error on page C-15.
- Updated the BACKCOPY Syntax on page D-1.
- Updated the BACKUP File-Mode Syntax on page D-1.

Changes to the H06.14/J06.03 Manual

- Added error 6011 on page A-3.
- References to Release Version Updates (RVUs) throughout this manual have been updated to include references to J-series RVUs, where appropriate.

Changes to the 523325-009 Manual

- Added TRUSTED, TRUSTME, and TRUSTSHARED attributes to Table 3-1, File Attributes for the WHERE Expression, on page 3-11.
- Updated the description of CODE code on page 3-24.
- Added an example to Using Qualified File-Set Lists on page 3-48.
- Updated the RESTORE program listing on page 7-13.
- Added one new condition to Restoring Enscribe Files on page 7-46.
- Updated message 7147 on page A-31.
- Updated message 8170 on page A-107.
- Added TRUSTED, TRUSTME, and TRUSTSHARED attributes to *file-attributes* on page D-4.

- Updated the TAPE DEFINE attribute VOLUME { volume-id | SCRATCH } on page E-6.
- Added these messages:
  - 7046 on page A-17
  - 7078 on page A-22
  - 7080 on page A-22
  - 7103 on page A-25
  - 7105 on page A-25
  - 7130 on page A-28
  - 7138 on page A-29
  - 7274 on page A-36

Changes to the G06.24 Manual

- DCOM does not permit multiple volumes in the command line.
- You cannot change the default workfile because DCOM/DSAP has been a native object since the G06.16 RVU.
- DSAP `$volume`, including `$*`, with a permanent workfile option displays reports for the selected volumes as long as the collective workfile size is less than or equal to 500MB minus the SQL Buffer size.

About This Manual

This manual describes how to use these disk and tape utilities for the NonStop operating system:

BACKCOPY	Disk Space Analysis Program (DSAP)
BACKUP (T9074)	PAK/UNPAK
Disk Compression (DCOM)	RESTORE

The G06 product version of the utilities cannot be executed on G-series preceding the G06.00 RVU. The D46 product version of the utilities cannot be executed on D-series preceding the D46.00 RVU.

Audience

This manual is written for system operators, system managers, and all other users of the disk and tape utilities. You should be familiar with:

- The user interface for the NonStop operating system
- The user interface for the HP Tandem Advanced Command Language (TACL) command interpreter
- Basic NonStop system operations

Organization

Section/Appendix	Description
Section 1, Disk and Tape Utilities Overview	Introduces each of the disk and tape utilities and provides general information about mounting tapes, tape formats, and entering BACKUP and RESTORE commands
Section 2, BACKCOPY	Describes BACKCOPY, a utility that duplicates backup tapes
Section 3, BACKUP	Describes BACKUP, a utility that backs up disk files to tape
Section 4, Disk Compression (DCOM)	Describes Disk Compression (DCOM), a utility that compresses disks (by moving extents) to save disk space
Section 5, Disk Space Analysis Program (DSAP)	Describes the Disk Space Analysis Program (DSAP), a utility that analyzes how disk space is being used on a specific volume
Section 6, PAK/UNPAK	Describes PAK and UNPAK, a pair of utilities used to compress and decompress files on a NonStop system
Section 7, RESTORE	Describes RESTORE, a utility that restores tapes to disk

Section/Appendix	Description
Appendix A, BACKCOPY, BACKUP, and RESTORE Messages	Describes the messages returned by the BACKCOPY, BACKUP, and RESTORE utilities
Appendix B, DCOM/DSAP Messages	Describes the messages returned by the DCOM and DSAP utilities
Appendix C, PAK/UNPAK Messages	Describes the messages returned by the PAK and UNPAK utilities
Appendix D, Syntax Summaries	Is a syntax summary for all of the disk and tape utilities
Appendix E, CLASS TAPE DEFINEs	Describes the attributes of the CLASS TAPE DEFINEs that relate to the disk and tape utilities
Appendix F, Tape Label Formats	Shows ANSI and IBM tape label format standards

Related Documentation

These manuals provide additional information that relates to this manual:

- *File Utility Program (FUP) Reference Manual*
- *Guardian Procedure Errors and Messages Manual*
- *Guardian User's Guide*

For information about other products referenced in this manual:

- *Backup and Restore 2.0 Manual*
- *Binder Manual*
- *DSM/Tape Catalog Operator Interface (MEDIACOM) Manual*
- *DSM/Tape Catalog User's Guide*
- *TMF Planning and Configuration Guide*
- *TMF Operations and Recovery Guide*
- *SQL/MP Installation and Management Guide*
- *SQL/MP Programming Manual for C*
- *SQL/MP Programming Manual for COBOL*
- *SQL/MX Installation and Mangement Guide*
- *SQL/MX Programming Manual for C and COBOL*
- *TACL Reference Manual*

Notation Conventions

Hypertext Links

Blue underline is used to indicate a hypertext link within text. By clicking a passage of text with a blue underline, you are taken to the location described. For example:

> This requirement is described under Backup DAM Volumes and Physical Disk Drives on page 3-2.

General Syntax Notation

This list summarizes the notation conventions for syntax presentation in this manual.

UPPERCASE LETTERS. Uppercase letters indicate keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

```
MAXATTACH
```

lowercase italic letters. Lowercase italic letters indicate variable items that you supply. Items not enclosed in brackets are required. For example:

```
file-name
```

computer type. `Computer type` letters within text indicate C and Open System Services (OSS) keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

```
myfile.c
```

italic computer type. *`Italic computer type`* letters within text indicate C and Open System Services (OSS) variable items that you supply. Items not enclosed in brackets are required. For example:

```
pathname
```

[ ] Brackets. Brackets enclose optional syntax items. For example:

```
TERM [\system-name.]$terminal-name
```

```
INT[ERRUPTS]
```

A group of items enclosed in brackets is a list from which you can choose one item or none. The items in the list can be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
FC [ num  ]
   [ -num ]
   [ text ]
```

```
K [ X | D ] address
```

{ } Braces. A group of items enclosed in braces is a list from which you are required to choose one item. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
LISTOPENS PROCESS { $appl-mgr-name }
                  { $process-name  }

ALLOWSU { ON | OFF }
```

| Vertical Line. A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
INSPECT { OFF | ON | SAVEABEND }
```

… Ellipsis. An ellipsis immediately following a pair of brackets or braces indicates that you can repeat the enclosed sequence of syntax items any number of times. For example:

```
M address [ , new-value ]...

[ - ] {0|1|2|3|4|5|6|7|8|9}...
```

An ellipsis immediately following a single syntax item indicates that you can repeat that syntax item any number of times. For example:

```
"s-char..."
```

Punctuation. Parentheses, commas, semicolons, and other symbols not previously described must be typed as shown. For example:

```
error := NEXTFILENAME ( file-name ) ;

LISTOPENS SU $process-name.#su-name
```

Quotation marks around a symbol such as a bracket or brace indicate the symbol is a required character that you must type as shown. For example:

```
"[" repetition-constant-list "]"
```

Item Spacing. Spaces shown between items are required unless one of the items is a punctuation symbol such as a parenthesis or a comma. For example:

```
CALL STEPMOM ( process-id ) ;
```

If there is no space between two items, spaces are not permitted. In this example, no spaces are permitted between the period and any other items:

```
$process-name.#su-name
```

Line Spacing. If the syntax of a command is too long to fit on a single line, each continuation line is indented three spaces and is separated from the preceding line by

a blank line. This spacing distinguishes items in a continuation line from items in a vertical list of selections. For example:

```
ALTER [ / OUT file-spec / ] LINE

   [ , attribute-spec ]...
```

!i and !o. In procedure calls, the !i notation follows an input parameter (one that passes data to the called procedure); the !o notation follows an output parameter (one that returns data to the calling program). For example:

```
CALL CHECKRESIZESEGMENT (  segment-id                    !i
                         , error        ) ;              !o
```

!i,o. In procedure calls, the !i,o notation follows an input/output parameter (one that both passes data to the called procedure and returns data to the calling program). For example:

```
error := COMPRESSEDIT ( filenum ) ;                      !i,o
```

!i:i. In procedure calls, the !i:i notation follows an input string parameter that has a corresponding parameter specifying the length of the string in bytes. For example:

```
error := FILENAME_COMPARE_ (  filename1:length           !i:i
                            , filename2:length ) ;       !i:i
```

!o:i. In procedure calls, the !o:i notation follows an output buffer parameter that has a corresponding input parameter specifying the maximum length of the output buffer in bytes. For example:

```
error := FILE_GETINFO_ (  filenum                        !i
                        , [ filename:maxlen ] ) ;        !o:i
```

Notation for Messages

This list summarizes the notation conventions for the presentation of displayed messages in this manual.

Bold Text. Bold text in an example indicates user input typed at the terminal. For example:

```
ENTER RUN CODE

?123

CODE RECEIVED:      123.00
```

The user must press the Return key after typing the input.

Nonitalic text. Nonitalic letters, numbers, and punctuation indicate text that is displayed or returned exactly as shown. For example:

```
Backup Up.
```

lowercase italic letters. Lowercase italic letters indicate variable items whose values are displayed or returned. For example:

```
p-register

process-name
```

[ ] Brackets. Brackets enclose items that are sometimes, but not always, displayed. For example:

```
Event number = number [ Subject = first-subject-value ]
```

A group of items enclosed in brackets is a list of all possible items that can be displayed, of which one or none might actually be displayed. The items in the list can be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
proc-name trapped [ in SQL | in SQL file system ]
```

{ } Braces. A group of items enclosed in braces is a list of all possible items that can be displayed, of which one is actually displayed. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
obj-type obj-name state changed to state, caused by
{ Object | Operator | Service }

process-name State changed from old-objstate to objstate
{ Operator Request. }
{ Unknown.          }
```

| Vertical Line. A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
Transfer status: { OK | Failed }
```

% Percent Sign. A percent sign precedes a number that is not in decimal notation. The % notation precedes an octal number. The %B notation precedes a binary number. The %H notation precedes a hexadecimal number. For example:

```
%005400

%B101111

%H2F

P=%p-register E=%e-register
```

Notation for Management Programming Interfaces

This list summarizes the notation conventions used in the boxed descriptions of programmatic commands, event messages, and error lists in this manual.

UPPERCASE LETTERS. Uppercase letters indicate names from definition files. Type these names exactly as shown. For example:

```
ZCOM-TKN-SUBJ-SERV
```

lowercase letters. Words in lowercase letters are words that are part of the notation, including Data Definition Language (DDL) keywords. For example:

```
token-type
```

!r. The !r notation following a token or field name indicates that the token or field is required. For example:

```
ZCOM-TKN-OBJNAME      token-type ZSPI-TYP-STRING.          !r
```

!o. The !o notation following a token or field name indicates that the token or field is optional. For example:

```
ZSPI-TKN-MANAGER      token-type ZSPI-TYP-FNAME32.         !o
```

Change Bar Notation

Change bars are used to indicate substantive differences between this manual and its preceding version. Change bars are vertical rules placed in the right margin of changed portions of text, figures, tables, examples, and so on. Change bars highlight new or revised information. For example:

> The message types specified in the REPORT clause are different in the COBOL environment and the Common Run-Time Environment (CRE).
>
> The CRE has many new message types and some new message type codes for old message types. In the CRE, the message type SYSTEM includes all messages except LOGICAL-CLOSE and LOGICAL-OPEN.

1 Disk and Tape Utilities Overview

This section briefly describes the disk and tape utilities of the NonStop operating system. The usage scenarios of most of these utilities are defined clearly by their basic functions. For the BACKUP, RESTORE, and BACKCOPY utilities, which have overlapping uses, this section also explains the similarities and differences in their usage:

Throughout this manual, the term *disk* refers to both magnetic and optical disk for each utility, except for DCOM and DSAP, which work only on magnetic disks. The term *tape* refers to magnetic tape.

The two formats for disk files are:

- A format 1 file is any file created on a system running an RVU preceding G06.00 or D46.00 or a file created on any more recent RVU that is smaller than 2 GB minus 1 MB.
- A format 2 file is either a large format file or a file that can contain larger partitions than a file created on RVUs preceding G06.00 or D46.00. A format 2 file can exceed the 2 GB minus 1 MB size limit of a format 1 file.

For more information about handling format 1 and format 2 files, see the *File Utility Program (FUP) Reference Manual*.

Utility Descriptions

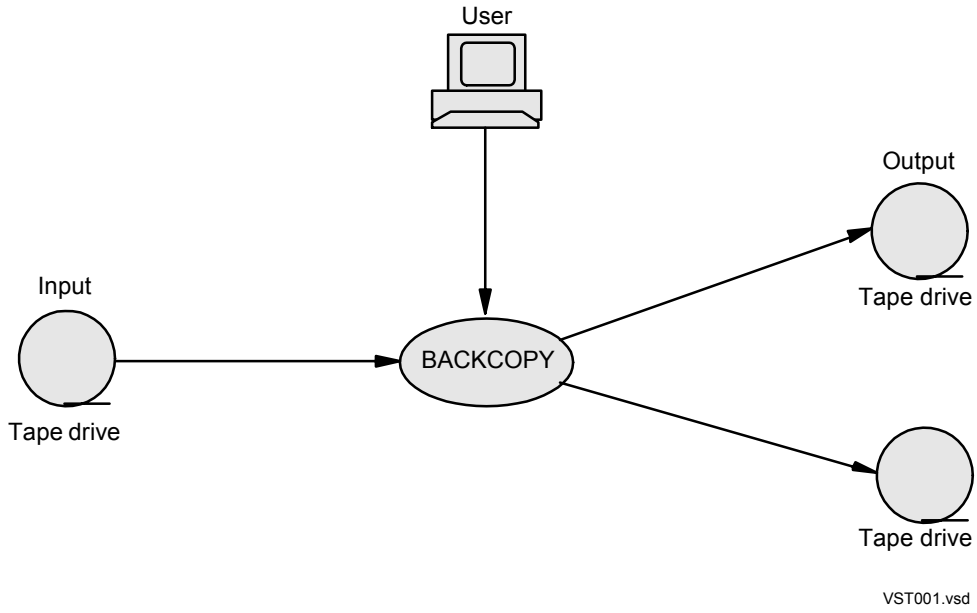
Table 1-1. Disk and Tape Utilities and Their Associated Tasks

Task	Utility
Duplicate backup tapes	BACKCOPY
Back up files on disk to tape	BACKUP
Move file extents to gain more usable disk space	DCOM
Analyze and generate reports on disk space usage	DSAP
Compress or decompress files	PAK/UNPAK
Copy files on tape to disk	RESTORE

BACKCOPY

The BACKCOPY utility makes duplicates of any tape created using the file-mode feature of the BACKUP utility. The BACKCOPY utility can create up to two duplicate tapes for archiving, distribution, or disaster recovery. BACKCOPY can be used for migrating backups from or to appendable pools. This also facilitates migrating backups to older RVUs.

Figure 1-1. BACKCOPY Utility



BACKUP

The BACKUP utility copies files from disk to magnetic tape. Using BACKUP minimizes the risk of losing valuable data by disk failure or human error. Information stored using BACKUP can be copied back onto your system using the RESTORE utility.

BRCOM (T2721), the interface to Backup and Restore 2.0, coexists with the Backup and Restore utilities (T9074).

△ **Caution.** The syntax for Backup and Restore 2.0 is different from the syntax for the Backup and Restore utilities (T9074). If you enter the syntax incorrectly for each file type at the BRCOM prompt, BRCOM might not be able to determine if the command needs to be forwarded to the Backup and Restore utilities (T9074).

You must use the syntax documented in this manual for Enscribe and SQL/MP files. For Open System Services (OSS) and SQL/MX files, you must use the syntax documented in the *Backup and Restore 2.0 Manual*.

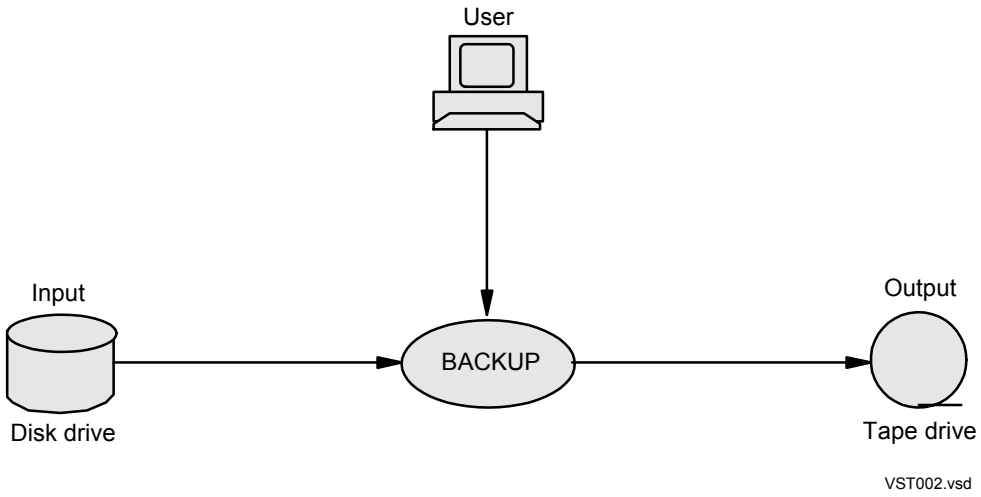
Note. BRCOM does not support the same tape format as the Backup and Restore utilities (T9074). You cannot mix Enscribe and SQL/MP files on the same tape with OSS and SQL/MX files.

BACKUP has two modes of operation:

- File mode backs up individual disk files. Any user can run BACKUP in file mode.
- Volume mode backs up all files on a disk volume. Only the super ID (usually the system manager or operator with a user ID of 255,255) can use volume-mode BACKUP.

Note. The RESTORE utility also uses file and volume mode. If a tape is created by BACKUP in one mode (file or volume), the RESTORE utility must use the same mode to process it.

Figure 1-2. BACKUP Utility

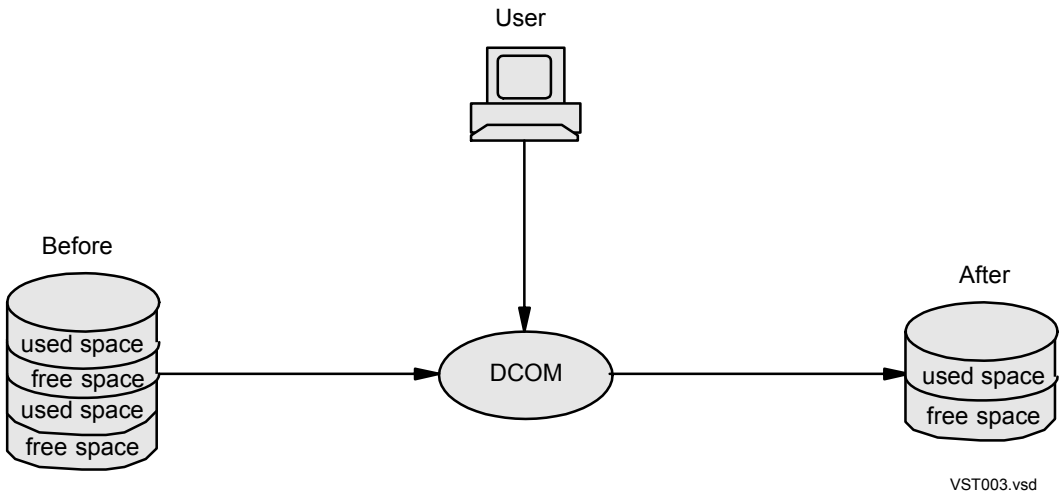


DCOM

The Disk Compression (DCOM) utility compresses the space used by disk files. DCOM moves allocated file extents (areas reserved for the growth of designated files) and free-space extents (unallocated areas) to different locations on a disk volume. DCOM consolidates the free-space extents, making larger extents available. After using DCOM to compress disk space, you can allocate new files with larger extent sizes.

DCOM is for use by programmers, system operators, and system managers.

Figure 1-3. DCOM Utility



DSAP

The Disk Space Analysis Program (DSAP) analyzes how the space on a disk volume is being used. You can use DSAP to determine how many free-space pages, allocated pages, deallocatable extent pages, and unused pages are on a disk. A page is a 2048-byte block of disk space.

DSAP/DCOM have a DSAPCSTM file to customize the DSAP environment. You can specify any DSAP options in the DSAPCSTM file. DSAP options specified in the DSAPCSTM file are appended to the DSAP command that you enter at the TACL prompt. For more information on these features, see DSAPCSTM File on page 5-37.

You can receive DSAP output in several different report formats that analyze the disk differently.

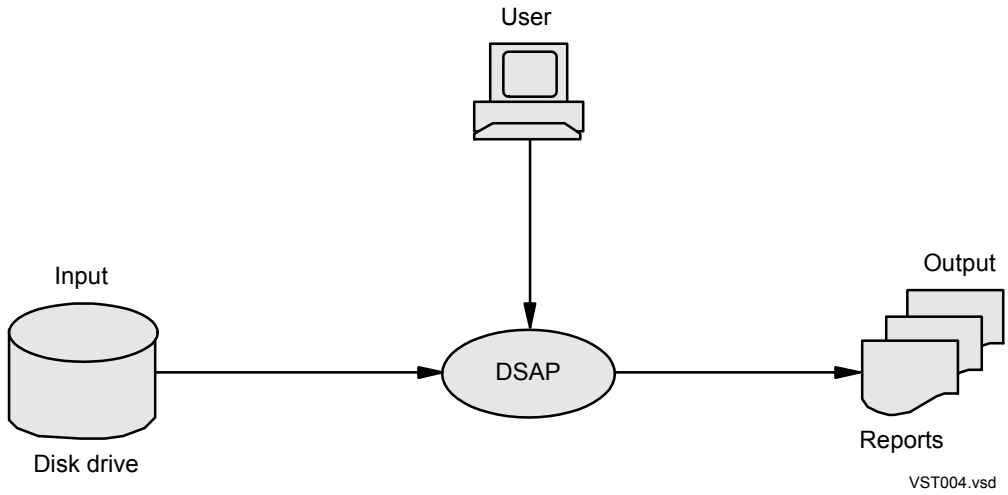
Table 1-2. DSAP Report Samples (page 1 of 2)

Report	Description
Subvol Summary Report	Analyzes the space used by each subvolume on a disk.
User Summary Report	Analyzes the space used by each user who owns files on the disk.
Detail Report	Lists the file name and space used by each file on the disk.

Table 1-2. DSAP Report Samples (page 2 of 2)

Report	Description
Summary of Space Use Report	Provides a general summary of how space is used on the disk.
Short Report	Shows the name, mirrored volume, volume capacity, volume free space, free-space fragments, and largest free-space fragment for each volume selected.
Tabular Report	Formats the output for different DSAP options by arranging the data in columns. The report can be directly ported to any popular spreadsheet program.

Figure 1-4. DSAP Utility



PAK/UNPAK

The PAK utility compresses NonStop files. PAK compresses Guardian files into a single unstructured archive file on any Expand-connected NonStop system. You can create a self-extracting archive file or use UNPAK to decompress the archive file when needed.

PAK efficiently and compactly collects data, which is useful in situations such as transferring large numbers of files to the Global Customer Support Center (GCSC) for analysis.

PAK and UNPAK work directly with BACKUP and RESTORE:

- PAK uses BACKUP to read files so its syntax is identical to BACKUP syntax. For example, you can use multiple file sets, wild cards, WHERE conditions and most BACKUP options.
- UNPAK uses RESTORE to decompress files from the archive file.

- PAK starts BACKUP, and UNPAK starts RESTORE.
- To run PAK or UNPAK, the BACKUP and RESTORE programs must be available (that is, they must exist and be secured so you can execute them).

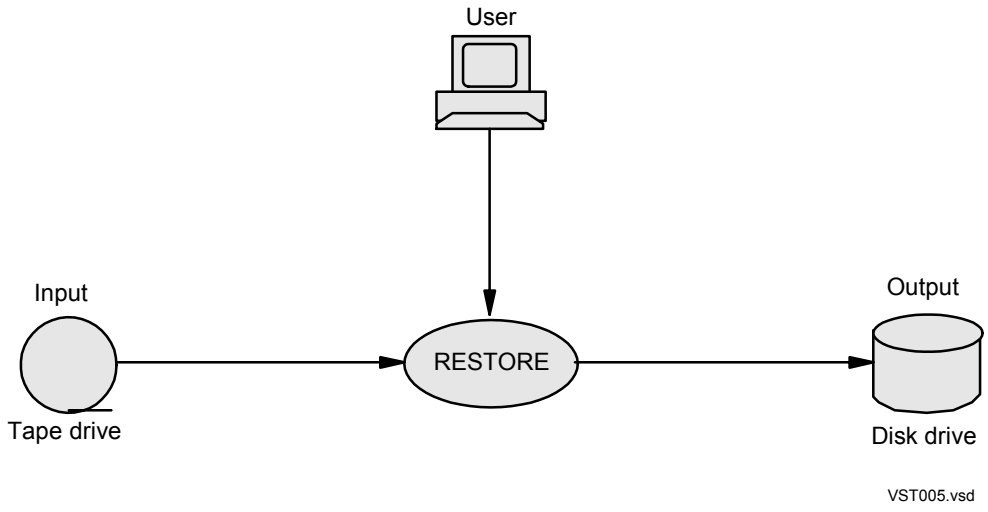
Because PAK and UNPAK use BACKUP and RESTORE, you need to be familiar with these utilities.

RESTORE

The RESTORE utility copies files from a backup tape to disk, and displays tape file information. This utility is essential for complete system management, especially after a disk failure or human error causes disk data to be lost. The RESTORE utility has three modes of operation:

- LISTONLY mode generates information about files on a backup tape without restoring the files to disk.
- File mode copies individual files to disk from a tape created by file-mode BACKUP.
- Volume mode re-creates a whole disk volume from a tape that was created by volume-mode BACKUP. Volume-mode RESTORE can be used only by the super ID (255,255), usually the system manager or operator.

Figure 1-5. RESTORE Utility



Using BACKCOPY, BACKUP, and RESTORE

The BACKCOPY, BACKUP, and RESTORE utilities are related programs that share common procedures:

- Commands used by BACKUP and RESTORE are entered the same way, and both use qualified file-set lists.
- All three utilities share a common procedure for mounting tapes on the tape drive.
- The utilities can use the CLASS TAPE DEFINE to trigger $ZSVR to process a labeled-tape request.

Note. BACKUP and RESTORE can also use the CLASS TAPECATALOG DEFINE to trigger $ZSVR to process a labeled-tape request. For information about the CLASS TAPECATALOG DEFINE, see the *DSM/Tape Catalog User's Guide*. The appendable tape feature is only available using CLASS TAPECATALOG DEFINE.

Entering BACKUP and RESTORE Commands

To enter commands in BACKUP and RESTORE, either:

- Type a complete command at the command interpreter prompt:

  ```
  1> BACKUP /HIGHPIN OFF/ $TAPE, \CSYS.$MYVOL.MYSUBVOL.*,&
  1>& LISTALL
  ```

  This example shows how to use the HIGHPIN OFF option to back up files on D-series or G-series by using the TACL RUN command. By default, HIGHPIN OFF runs at a low PIN and is needed only if BACKUP was manually set to run at a high PIN. For information on creating and running a high-PIN process, see the *Guardian Application Conversion Guide*.

- Specify a command file as an IN file in your initial command to start the program. The IN file contains the device name, qualified file-set list, and any utility options. An IN file is useful if the BACKUP command is long and inconvenient to type at the command interpreter line. This is a complete command:

  ```
  6> BACKUP / IN $RECDS.BAKERY.BCOMM /
  ```

  $RECDS.BAKERY.BCOMM is an IN file that contains these command elements:

  ```
  $TAPE,
  ($PIES.APPLE.*,
  $PIES.PEACH.*,
  $PIES.PECAN.*,
  $PIES.CHERRY.*,
  $ROLLS.KAISER.*,
  $ROLLS.DINNER.*),
  LISTALL
  ```

  IN files cannot contain an ampersand (&) at the end of each line as can commands entered from the command interpreter prompt. The command interpreter reads the

ampersand as a continuation of the command, but the disk and tape utilities do not.

Mounting a Tape

To mount an unlabeled tape for a BACKCOPY, BACKUP, or RESTORE process:

1. Check the input and output media write settings:
   - For a BACKUP or BACKCOPY process, ensure the output tape media is write-enabled. For tape reels be sure the write ring is inserted.
   - For a RESTORE or BACKCOPY ensure the input tape media is not write-enabled. For tape reels be sure the write ring is removed.

△ **Caution.** Removing the write-enable rings prevents the possibility of any inadvertent backups (and subsequent overwrites) to the tapes. This recommended action is only a precaution. RESTORE operates with or without the presence of the write-enable rings on the tape reels.

2. Enter a complete BACKCOPY, BACKUP, or RESTORE command.
3. If you used the NOPROMPT option in a BACKUP or RESTORE command (which is acceptable except in some cases when you use the NOREWIND option), go to Step 8.
4. Wait until the program displays:

   ```
   $tape : Not ready?
   ```

   Then press the Return key and wait for the same message again. The *tape* is the name of the tape drive specified in the command.
5. Mount the tape and press the Return key to start the process on this tape. If you press the Return key before you mount the tape, the process on this tape begins when the system detects that the tape is mounted and ready.
6. Each subsequent reel causes one of these prompts to appear:

   ```
   Mount correct tape #n?
   Mount next tape #n?
   Mount previous tape #n?
   ```
7. For each subsequent prompt, go to Step 5. When no more tape prompts appear, go to Step 9.
8. If you used the NOPROMPT option, this noninteractive message appears:

   ```
   $tape: device not ready -- beginning to poll --
   ```

   Mount the first (or the next) tape. For each reel, the program continues when it detects that the drive is ready.
9. When there are no more tape prompts, the process is complete.

Exiting the Program

To stop the program when the “Not ready?” message appears, enter STOP or press Ctrl-y.

To stop the program at any other time, press the Break key and enter the STOP command at the TACL prompt.

Dismounting a Tape

When you dismount a tape from a tape drive, $ZSVR issues this status message:

```
STATUS 1512: vid TAPE DISMOUNTED FROM DRIVE drive
```

vid

is the ID of the volume on which the tape was mounted.

drive

is the drive on which the volume was mounted.

$ZSVR is cannot to detect when you have manually dismounted a tape from a tape drive until you mount another tape on the drive (or until you open the drive). In these situations, $ZSVR issues the status message after a new tape is mounted or the tape drive is opened.

If you manually dismount an unlabeled tape and mount another unlabeled tape on the same drive, $ZSVR does not detect a tape dismount and does not issue the status message.

The volume ID printed in the status message (*vid*) is incorrect if the dismount is caused by any of these commands:

- CLOSE command of a LABELS BYPASS OPEN
- MEDIACOM LABEL (IBM) command without the NOUNLOAD option
- UNLABEL command without the NOUNLOAD option
- RELABEL (IBM) command without the NOUNLOAD option
- SCRATCH command without the NOUNLOAD option
- DUMPLABELS command without the NOUNLOAD option

The volume ID printed in the status message corresponds to the volume that was online when the LABELS BYPASS OPEN command was issued (or during the labeling operation). If a volume was not online (or the volume online was an unlabeled volume), the volume ID is blank in the status message.

Disk Formats

Beginning with the G06.00 and D46.00 RVUs, a new format for disk files, format 2, is used to describe large format files.

Format 1 describes files created on:

- RVUs G06.00, D46.00, or later that are smaller than 2 GB minus 1 MB.
- RVUs preceding G06.00 or D46.00.

A format 2 file is either a large format file, or a file that can contain larger partitions than a file created on RVUs preceding G06.00 or D46.00. A format 2 file has the potential of exceeding the 2 GB - 1 MB size limit of a format 1 file.

For more information about handling format 1 and format 2 files, see the *File Utility Program (FUP) Reference Manual*.

Tape Formats

A tape used with the BACKCOPY, BACKUP, or RESTORE utilities can have one of five different tape formats. (See Table 1-3.) The format includes the tape format number, the type of files stored on a tape with that particular format, the BACKUP option that can cause that type of files to be stored on the tape, and the version of the RESTORE utility that recognizes that particular tape format. Multiple tape formats exist because of continuing improvements in disk and tape processing.

Table 1-3. Tape Formats

Tape Format	Files	BACKUP Option	RESTORE Version
0	DP1 files only	OLDFORMAT (no longer supported)	A01 to B40
1	DP1 files only	DP1FORMAT	A04 to B40
2	DP2 files, or file sets with both DP2 and DP1 files	DP2FORMAT	B00 or later
3	Archive tape format for parallel copies, labeled tapes, SQL objects, and DP2 files	ARCHIVEFORMAT	C00 or later

Tape Format History

The history of the various tape formats and their associated attributes are summarized in Table 1-4, which identifies the NonStop software version in which these attributes were introduced.

Table 1-4. History and Attributes of Tape Formats

Associated Attributes	Tape Format 0	Tape Format 1	Tape Format 2	Tape Format 3
Software version in which tape format was introduced	A01 NonStop 1+	A04 NonStop 1+	B00 NonStop II	C00 NonStop II
Disk process	DP1	DP1 only	DP1* DP2**	DP2
Record size	2048 bytes	Up to 30 KB	Up to 30 KB***	Up to 30 KB ***
Checksums and sequence numbers per record	No	Yes	Yes	Yes
Compatible RESTORE versions	A01 to B40	A04 to B40	B00 or later	C00 or later
Compatible between NonStop and NonStop 1+	Yes	Yes, only if BLOCKSIZE= 2	No	No

* If the input files are DP1 files or the DP1FORMAT option is specified, tape format 1 is used.

** If the input files include DP2 files or the DP2FORMAT option is specified, tape format 2 is used.

*** Record size (BLOCKSIZE) for DP2 tapes can be an even value in the range 2 through 30, but BACKUP rounds the given BLOCKSIZE down to a multiple of 4 (unless BLOCKSIZE is 2). For details, see BLOCKSIZE on page 3-15.

Reason for introducing each tape format:

Tape format 0	Carried over from NonStop 1+ (unchanged)
Tape format 1	To accommodate large block size and checksums
Tape format 2	To accommodate DP2 files
Tape format 3	To facilitate labeled tapes, and parallel copies, and SQL objects

The default tape format for each version of the NonStop operating system is displayed in Table 1-5. Starting with the B00 software release, the default tape version is conditional. The table shows each BACKUP option that overrides the default and the resulting tape format when these parameters are specified.

Table 1-5. Default Tape Formats and Tape Format Options

NonStop operating system Version	Default Tape Format and Backup Parameters That Override the Default (File-Mode Operations Only)	
A03—A04	Tape format 0 only (carried over from NonStop 1+)	
A04—A30	Tape format 1 introduced	
	Default:	Tape format 1
	Override:	Tape format 0 if OLDFORMAT option is specified
B00—B30	Tape format 2 introduced. Conditional default:	
	a. Default:	Tape format 1 if only DP1 files are in the file-set list and no tape formatting parameter is specified
	b. Default:	Tape format 2 if any DP2 file is in the file-set list and no tape formatting parameter is specified
	c. Override:	Tape format 1 if DP1FORMAT BACKUP option is specified
	d. Override:	Tape format 2 if DP2FORMAT BACKUP option is specified
B40	Default:	Tape format 2; B00 rules c and d still apply
C00—C3x, Dxx	Tape format 3 introduced. Conditional default:	
	a. Default:	Tape format 2 if no tape format 3 attributes are present and no tape formatting parameter is specified
	b. Default:	Tape format 3 if labeled tapes are used, parallel copies are specified, or system is configured with SQL
	c. Override:	Tape format 1 if DP1FORMAT BACKUP option is specified
	d. Override:	Tape format 2 if DP2FORMAT BACKUP option is specified
	e. Override:	Tape format 3 if ARCHIVEFORMAT BACKUP or BACKCOPY option is specified
D32 (ADL)—D38, D42 (ADE)—D48, Gxx	a. Default	Tape format 3
	b. Override	Tape format 1 if DP1FORMAT BACKUP option is specified
	c. Override	Tape format 2 if DP2FORMAT BACKUP option is specified

Using BACKUP and RESTORE With DSM/Tape Catalog

The Distributed Systems Management/Tape Catalog (DSM/TC) is an online database containing information about tape volumes, tape files, and disk files. By using DSM/TC during backup, you can keep track of what files are on a tape volume without printing out a listing or restoring the files. You can also protect backup volumes against accidental overwrites.

BACKUP and RESTORE communicate to DSM/TC through a specific define class, TAPECATALOG.

For information on the use of BACKUP and RESTORE with DSM/TC, and on the attributes of DEFINE (CLASS TAPECATALOG), see the *DSM/Tape Catalog User's Guide*. For information on DSM/TC and its operator interface (MEDIACOM), see the *DSM/Tape Catalog Operator Interface (MEDIACOM) Manual*.

2 BACKCOPY

The BACKCOPY utility duplicates tapes that are made from a file-mode operation in BACKUP. It cannot to duplicate tapes that are made from a volume-mode BACKUP operation. For more information about BACKUP, see Section 3, BACKUP.

BACKCOPY can create one or two duplicate tapes for archive storage, distribution, or disaster recovery. It can also create one or two labeled (or unlabeled) tape sets from a labeled or unlabeled tape set.

Although BACKCOPY generates a tape in the same tape format as the original when it makes one duplicate copy of a tape, it generates tapes in tape format 3 (archive format) when it makes two duplicate copies of a tape.

Security

BACKCOPY is not a privileged program. Each customer determines which users are allowed to run it. You can control access to BACKCOPY in several ways. For example:

- Limit access to a subset of users by either setting file security attributes or using a Safeguard access-control list.
- Give the super-group user (255, *n*) EXECUTE access to a PROGID copy of the BACKCOPY program with PROGID set to the super ID (255, 255).

Check the security policy established by your organization to set the appropriate level of access to BACKCOPY and all system programs. For a complete discussion on securing information on a NonStop system, see the *Security Management Guide*.

BACKCOPY Syntax

The syntax for BACKCOPY is as follows. For a concise statement of BACKCOPY command syntax, see Appendix D, Syntax Summaries.

```
[[[\node.]$volume.]subvolume.]BACKCOPY

[ / run-option [ , run-option ] ... /                ]

   source-tape, dest-tape, *.*.*

   [ , ARCHIVEFORMAT                                 ]
   [ , CATALOGFILES                                  ]
   [ , { DENSITY density | TAPEMODE tapemode }       ]
   [ , LISTALL                                       ]
   [ , { NOREWINDIN | NOUNLOADIN }                   ]
   [ , { NOREWINDOUT | NOUNLOADOUT }                 ]
   [ , PAGELENGTH number                             ]
   [ , VERIFYREEL                                    ]
```

node.volume.subvolume

indicates the node, volume, and subvolume where BACKCOPY runs. It defaults to the current node, volume, and subvolume if these parameters are omitted.

run-option

is any option for the command interpreter RUN command. The two most common run options are:

```
IN filename
OUT listfile
```

The IN option specifies input files, and the OUT option specifies the output files. These files usually override the home terminal as the input or output device. An IN file is a text file containing the remainder of the command line, including parameters and options. For a complete list of *run-options* and a description of the command interpreter RUN command, see the *TACL Reference Manual*.

source-tape

is the name of the tape drive that reads the backup tape. You can specify a DEFINE of CLASS TAPE with the attribute LABELS BACKUP, LABELS IBMBACKUP, or LABELS BYPASS to read a labeled backup tape, or specify a DEFINE of CLASS TAPECATALOG to copy tapes and catalog disk file entries.

The value of *source-tape* is one of:

```
{ [ \node.]$device }
{ [ \node.]$ldev   }
{ [ define-name ]  }
```

node

is the name of the local or remote node (system) where the tape drive resides.

device

is the logical name of the magnetic tape unit, such as $TAPE1.

ldev

is the device number of the magnetic tape unit, such as $17.

define-name

specifies a DEFINE name of CLASS TAPE for copying a labeled backup tape. For a list of all the CLASS TAPE DEFINE attributes, see Appendix E, CLASS TAPE DEFINEs.

dest-tape

is the name of the tape drive or drives to which duplicate tapes are written. You can use a DEFINE of CLASS TAPE with the attribute LABELS BACKUP, LABELS IBMBACKUP, or LABELS BYPASS to write a labeled tape.

The value of *dest-tape* is one of:

```
{ [ \node.]$device                                                    }
{ ( [ \node.]$device, [ \node.]$device ) }
{ [ \node.]$ldev                                                      }
{ [ \node.]( $ldev, $ldev )                                           }
{ [ define-name ]                                                     }
```

node

is the name of the local or remote node (system) where the tape drive resides.

device

is the logical name of the magnetic tape unit, such as $TAPE1.

ldev

is the device number of the magnetic tape unit, such as $17.

define-name

specifies a DEFINE name of CLASS TAPE for copying a labeled backup tape. For a list of all CLASS TAPE DEFINE attributes, see Appendix E, CLASS TAPE DEFINEs.

..*

specifies that all files on *source-tape* are copied to *dest-tape*. You must use this syntax (*.*.*). You cannot specify individual files.

ARCHIVEFORMAT

specifies tape format 3 for *dest-tape*. If you specify two tape drives for *dest-tape* (to make two copies of the backup tape), ARCHIVEFORMAT is automatically selected. If you specify one tape drive and do not specify ARCHIVEFORMAT, the tape format is the same as the format on the original backup tape.

CATALOGFILES

specifies catalog information about disk files. For more information and examples, see the *DSM/Tape Catalog User's Guide*.

DENSITY *density*

sets the recording density of the destination tape drive in bits per inch (bpi). The default is the physical setting of the destination drive.

Density Specified	Recording Density (in bpi)
GCR or 6250	6250
PE or 1600	1600
NRZI or 800	800

The tape drives supported on D-series do not include an 800-bpi density.

LISTALL

lists on the terminal (or in the output file) the names of all of the files that were copied successfully, and the names of files that caused errors.

NOREWINDIN

directs the BACKUP utility to leave the last read tape positioned at its current location and leave it online when the BACKUP process is completed. This option lets the tape be labeled for the next BACKUP without having to search for the end of tape. This option is available starting with the G06 version of BACKUP.

NOREWINDOUT

directs the BACKUP utility to leave the last written tape positioned at its current location and leave it online when the BACKUP process is completed. This option lets the tape be labeled for the next BACKUP without having to search for the end of tape. This option is available starting with the G06 version of BACKUP.

Note. If you specify NOREWINDIN or NOREWINDOUT for an unlabeled tape, the tape is not usable until it is manually repositioned to its load point.

NOUNLOADIN

causes the final tape to remain rewound and left online after the RESTORE process is completed. If neither NOUNLOAD or NOREWIND is specified, the last tape is rewound and unloaded when the process is completed.

NOUNLOADOUT

directs BACKUP to rewind the final tape and leave it online when the BACKUP process is completed.

PAGELENGTH *number*

specifies the number of lines to be generated on each page of BACKCOPY output. The value of *number* must be an integer in the range 20 through 100. When BACKCOPY output is sent to a printer, a form feed (or page eject) is generated after the number of lines specified in the PAGELENGTH option. If PAGELENGTH is not specified, BACKCOPY defaults to 60 lines per page.

TAPEMODE *tapemode*

specifies the tape recording mode for tape units that support streaming.

STARTSTOP | STREAM

specifies the tape mode of the destination drive. It applies only to tape units that support streaming.

STARTSTOP

Tape speed is 50 ips (inches per second). Tape density is 12000 bpi (bits per inch). This mode writes one record to the tape at a time.

STREAM

Tape speed is 75 ips. Tape density is 12000 bpi. This mode writes one block to tape at a time.

Note. You cannot use the DENSITY and TAPEMODE in the same BACKCOPY command.

VERIFYREEL

directs BACKCOPY to verify each reel for data integrity after the reel is written.

Guidelines

Consider these guidelines when using BACKCOPY:

- BACKCOPY copies all files. If BACKCOPY cannot read a file because of a parity or checksum error, it displays an error message and the tape duplication fails.
- BACKCOPY automatically generates tapes in tape format 3 if any of these are true:
  - The duplicate tape is a labeled tape (*dest-tape* is a DEFINE name).
  - You are making two copies (by specifying two *dest-tape* names).
  - You specify the ARCHIVEFORMAT option.

  Otherwise, BACKCOPY creates a duplicate tape in the same format as the original.
- BACKCOPY can create tapes in formats 0, 1, 2, or 3.
- Only versions C00 (and later) of RESTORE can read tapes generated in tape format 3 (archive format).
- BACKCOPY supports labeled-tape processing in the same way as BACKUP. For more information, see Backing Up With Labeled Tapes on page 3-51.
- A duplicate tape generated with BACKCOPY is not the same as parallel tape copies generated with BACKUP:
  - When parallel copies are made with BACKUP, both copies contain exactly the same amount of information on each reel. You need to mount new tape reels for both copies at the same time. BACKUP makes identical copies so that each reel of tape is interchangeable with its corresponding copy during a RESTORE process.
  - When two copies are made with BACKCOPY, the amount of information on each tape reel can vary between the copies. You can mount a new tape whenever a drive reaches the end of reel; you do not necessarily change tape reels for both copies at the same time. The individual reels are not interchangeable. Only the whole duplicate tape set is interchangeable with the original set.
  - When you make a single copy of a tape using BACKCOPY, the BACKCOPY tape might contain a different amount of data from the BACKUP source tape. When you make two copies using BACKCOPY, these two copies contain the same amount of data and are therefore interchangeable. However, these copies might contain a different amount of data from the BACKUP source tape.

Examples: Different Ways to Use BACKCOPY

- To read an unlabeled backup tape on the device $TAPE1 and duplicate it on \REMOTE.$TAPE2:

```
1> BACKCOPY $TAPE1, \REMOTE.$TAPE2, *.*.*, LISTALL
```

- To read a labeled BACKUP tape and create an unlabeled duplicate of it on $TAPE2:

```
2> ADD DEFINE =Input, CLASS TAPE, LABELS BACKUP, &
2> &VOLUME (FOX082,FOX004)
3> BACKCOPY =Input, $TAPE2, *.*.*
```

- To read an unlabeled BACKUP tape on $TAPE1 and create a labeled duplicate of it:

```
4> ADD DEFINE =Output, CLASS TAPE, LABELS BACKUP, &
4> &VOLUME SCRATCH
5> BACKCOPY $TAPE1, =Output, *.*.*
```

- To read an unlabeled BACKUP tape on $TAPE1 and create two identical copies of it using tape format 3 (the default format when two copies are made):

```
6> BACKCOPY $TAPE1,($TAPE2, $TAPE3), *.*.*
```

- To copy a tape from an appendable pool to a nonappendable pool:

```
7> add define =DEF1, class tapecatalog, pool weekly, volcat
\CAUNI3.SILO1_VOLCAT, gen 1, version 0, device $tape0,
filecat
\CAUNI3.SILO1_FILECAT,FILEID W00032,USE IN

8> add define =DEF2, class tapecatalog, pool
weekly_appendable, volcat
\CAUNI3.SILO1_VOLCAT, device $tape2, filecat
\CAUNI3.SILO1_FILECAT,FILEID W00007,USE OUT

9>backcopy =DEF1, =DEF2,  *.*.*, LISTALL, NOREWINDOUT
```

Note. You can also use the process in this example for importing or exporting to an appendable pool.

Completion Information

BACKCOPY returns a completion code to the process that started it. The completion code indicates how successfully BACKCOPY ran. Its primary purpose is to support batch processing by allowing conditional execution of subsequent processes based on the success of the BACKCOPY operation.

If TACL started BACKCOPY, the completion code is stored in a TACL variable (:_COMPLETION), where you can examine it by using TACL functions. TACL also displays the completion codes at the home terminal unless BACKCOPY completes without any warnings or errors. For more information about completion codes and batch processing, see the *TACL Programming Guide*.

BACKCOPY also returns a subsystem identifier (TANDEM.76.*version*), where *version* is the release number (for example, D46). The subsystem identifier is also stored in the TACL variable (:_COMPLETION).

Table 2-1. Completion Codes Returned by BACKCOPY and the Operating System

Code	Description
0	A normal, voluntary termination. The process finished normally with no errors or warnings.
1	A normal, voluntary termination with warnings. All files were copied, but warnings were issued for one or more files. (Only file-specific warnings cause this code.)
2	An abnormal, voluntary termination with errors. BACKCOPY could not copy one or more files because of errors on the files. BACKCOPY terminates without finishing the copy.
3	A premature, voluntary termination with fatal errors. BACKCOPY terminated because of a fatal error or an invalid user command, and the copy generated might not be complete.
4	BACKCOPY never started. This completion code is generated by the TACL process executing BACKCOPY. The *terminationinfo* field holds the error code returned by the process creation procedure.

3 BACKUP

The BACKUP utility copies files from disk to magnetic tape. The RESTORE utility, described in Section 7, RESTORE, performs the complementary function of copying files from magnetic tape back to disk.

△ **Caution.** The syntax for Backup and Restore 2.0 is different from the syntax for the Backup and Restore utilities (T9074). If you enter the syntax incorrectly for each file type at the BRCOM prompt, BRCOM might not be able to determine if the command needs to be forwarded to the Backup and Restore utilities (T9074).

You must use the syntax documented in this manual for Enscribe and SQL/MP files. For OSS and SQL/MX files, you must use the syntax documented in the *Backup and Restore 2.0 Manual*.

Note. BRCOM does not support the same tape format as the Backup and Restore utilities (T9074). You cannot mix Enscribe and SQL/MP files on the same tape with OSS and SQL/MX files.

You can use BACKUP to:

- Store files on tape for recovery purposes if they are ever lost or damaged on the disk
- Free disk space by archiving files that are used infrequently
- Use labeled tapes
- Move files from one system to another
- Convert files from one type of disk process to another (from DP1 to DP2)

Note. Updated operating system software uses the DP2 standard and does not always support the DP1 standard. A conversion might be a one-time maintenance operation (or an infrequent necessity) when an old backup tape is restored.

Security

BACKUP is a privileged program. Each customer determines the users who are allowed to run it. You can control access to BACKUP in several ways, including:

- Do not license the program file for general use. Only the super ID (255, 255) can run the program without licensing. If BACKUP is not licensed, SQL tables cannot be backed up by anyone (including the super ID).
- License the program file and limit access to a subset of users by either setting file security attributes or using a Safeguard access-control list.

  Note. DSM/SCM automatically licenses the BACKUP program file. If you do not use DSM/SCM to install BACKUP, the BACKUP program file is not licensed. The super ID can use the FUP LICENSE command to license the BACKUP program file at any time.

- Give super-group users (255, *n*) EXECUTE access to a PROGID copy of the BACKUP utility, and set PROGID to the super ID (255, 255).
- Create a special user ID that is used only for running BACKUP. Use the Safeguard product to give this ID read-only access to all files, and give the password to a user who is responsible for backups. For information about configuring the Safeguard product, see the *Safeguard Administrator's Manual*.

To set the appropriate access level for BACKUP and all other system programs, see your organization's security policy. For a description of securing information on a NonStop system, see the *Security Management Guide*.

BACKUP and RESTORE Interaction

The BACKUP command options you use to create a tape directly affect how the tape is restored. After creating a backup, you cannot view all the options you used when you ran BACKUP, so note the options you use each time you run BACKUP.

BACKUP and RESTORE have two modes of operation:

- File mode, in which BACKUP copies one file at a time to create a tape that is a sequence of individual files.
- Volume mode, in which BACKUP transfers an entire disk volume to tape. Only the super ID (255,255) can use volume-mode BACKUP.

A tape created by BACKUP using one mode cannot be restored using the other mode.

You should be familiar with these subsections that apply to both utilities:

- Entering BACKUP and RESTORE Commands on page 1-7
- Mounting a Tape on page 1-8
- Tape Formats on page 1-10
- File Sets, File-Set Lists, and Qualified File-Set Lists on page 3-5.

BACKUP Syntax

BACKUP has different syntax options in file mode and volume mode. For concise statements of both modes of BACKUP syntax, see Appendix D, Syntax Summaries.

The syntax for the file mode of BACKUP is:

```
[\node.][$volume.][subvolume.]BACKUP

[ / run-option [, run-option ] ... / ]

   { tape-device-name           }
   { ( tape-device-name1, tape-device-name2, ... ) }

   , backup-files

   [ , file-mode-backup-option ] ...
```

The syntax for the volume mode of BACKUP is:

```
[\node.][$volume.][subvolume.]BACKUP

[ / run-option [, run-option ] ... / ]

   { tape-device-name           }
   { ( tape-device-name1, tape-device-name2, ... ) }

   , VOLUMEMODE, { $volume | $ldev } [ -P | -M ]

   [ , volume-mode-backup-option ] ...
```

node.volume.subvolume

indicates the node, volume, and subvolume where BACKUP runs. It defaults to the current node, volume, and subvolume if these parameters are omitted.

Note. The current node during a backup is the node where BACKUP is running. It is recommended that you run BACKUP on the node where your TACL process is running.

---

run-option

is any option for the TACL RUN command. The two most common run options are:

```
IN filename
OUT listfile
```

The IN option specifies input files, and the OUT option specifies the output files. These files usually override the home terminal as the input or output device. An IN file is a text file containing the remainder of the command line, including parameters and options. For a complete list of the run options and a description of the TACL RUN command, see the *TACL Reference Manual.*

tape-device-name

is the name of a tape drive to use for an unlabeled-tape BACKUP operation or a DEFINE name for a labeled-tape BACKUP operation. To create a single backup tape, specify one *tape-device-name*. To create two identical backup tapes on two separate drives, specify two names (enclosed in parentheses and separated by a comma). For unlabeled-tape operations, to specify up to four tape drives, use the MULTIDRIVE option on page 3-25. The *tape-device-name* is one of:

```
[ \node.]$device
[ \node.]$ldev
define-name
```

node

is the name of the node (system) where the tape drive resides.

device

is the name of the magnetic tape unit, such as $TAPE1.

ldev

is the device number of the magnetic tape unit, such as $17.

define-name

specifies a CLASS TAPE DEFINE name for backup to labeled tape. This DEFINE sends a request to $ZSVR (the labeled-tape server process) for labeled-tape processing.

backup-files

designates the files to be backed up in one of these formats:

```
fileset
fileset-list
qualified-fileset-list
```

as described in File Sets, File-Set Lists, and Qualified File-Set Lists on page 3-5.

file-mode-backup-option

specifies one or more conditions for the file-mode BACKUP operation (see File-Mode BACKUP Options on page 3-11).

VOLUMEMODE

instructs BACKUP to run in volume mode.

$volume | *$ldev*

specifies the disk volume name or logical device number for a volume-mode backup.

`-P`
`-M`

specifies the primary or mirror half of a volume pair to be backed up. If neither one is specified, the primary and mirror halves must be valid and matching, and the logical pair is backed up.

`volume-mode-backup-option`

specifies one or more conditions for the BACKUP operation (see Volume-Mode BACKUP Options on page 3-12).

File Sets, File-Set Lists, and Qualified File-Set Lists

You can specify precisely which files to back up in one of these formats:

`Fileset`	A set of files on a local or remote node
`File-set list`	One or more file sets
`Qualified file-set list`	A file set list that can contain conditions to include or exclude the backup of certain files

Note. This subsection applies to the syntax for both BACKUP and RESTORE.

Wild-Card Characters in a File Set

A file set can contain these wild-card characters in the *`volume`*, *`subvolume`*, and *`file-id`* fields:

* (asterisk)	Matches from 0 through 8 characters in the position where it appears
? (question mark)	Matches one character in the position where it appears

For example, the volume name $SB?? matches all four character volume names that begin with SB. The file name *CH? matches all file names that end with CH followed by any single character.

Fileset Syntax

```
fileset
```

is a set of files to back up from the current or another node (system) specified as:

`[[[ \`*`node`*`.]$`*`volume`*`.]`*`subvolume`*`.]`*`file-id`*

When you back up files from a node different from the one on which you are running BACKUP, the operation is known as a remote back up.

Files in a *`fileset`* are processed in alphabetical order (by volume) of their fully qualified Guardian names. A *`fileset`* can contain wild-card characters. If you omit *`node`*, *`volume`*, or *`subvolume`*, BACKUP assumes the current volume and subvolume.

For example, this *fileset* specifies all files ending with X on the PAYROLL subvolume of the current volume:

```
PAYROLL.*X
```

If you specify *volume*, you must also specify *subvolume*. Also, if you omit *subvolume*, you must also omit *node* and *volume*.

Note. You can specify only one node name with the BACKUP syntax. You can only back up files from one node during a single BACKUP operation. Run BACKUP once for each node that you want to back up.

```
fileset-list
```

is one or more *filesets* in the form:

```
( fileset [ , fileset ] ... )
```

For example, this *fileset-list* specifies all files on the subvolume PAYROLL (that are on the volume $DATA), and all the files on the volume $ACCOUNT that begin with B:

```
($DATA.PAYROLL.*, $ACCOUNT.*.B*)
```

```
qualified-fileset-list is:

{ list-element                                }
{ ( list-element [ , list-element ] ... )    }
```

A *qualified fileset-list* allows further exclusions from a *fileset-list*.

list-element is:

fileset-list [*qualifier* [*qualifier*] ...]

qualifier

specifies qualifying criteria for including files or objects in *fileset*. A qualifier is one of:

```
[ EXCLUDE fileset-list]
[ FROM CATALOG[S] catalog-list]
[ START filename]
[ WHERE expression]
```

Each qualifier can be specified only once for a given *fileset-list*. You can specify qualifiers in any order.

An example of a *list-element* of a *qualified-fileset-list* is:

```
$DATA.PAYROLL.*X WHERE (ENSCRIBE AND INDEX)
```

File-Set List Qualifiers

For examples of BACKUP commands that use qualified file set lists, see Using Qualified File-Set Lists on page 3-47.

`EXCLUDE` *`fileset-list`*

specifies files to exclude from the operation. The EXCLUDE qualifier works the same as the BACKUP NOT option.

`FROM CATALOG[S]` *`catalog-list`*

specifies backup of SQL objects in *`catalog-list`* that match the *`fileset-list`*. This qualifier specifies one or more SQL catalogs that contain descriptions of the SQL objects (tables, indexes, views, or SQL programs) to be backed up. The SQL catalogs are specified by volume and subvolume (or by a CLASS CATALOG DEFINE name).

`catalog-list` is any of:

```
[$volume.]subvolume
( [$volume.]subvolume [ , [$volume.]subvolume ] ... )
define-name
( define-name [ , define-name ] ... )
```

`START` *`filename`*

specifies a *`filename`* within the *`fileset`* or *`fileset-list`* at which to start the BACKUP. This is useful for restarting a backup operation that was interrupted. The only wild-card character allowed in a *`fileset`* qualified by the START qualifier is an asterisk (*) in place of the entire subvolume (or file ID) of a Guardian file name. The START qualifier works the same as the BACKUP START option.

`WHERE` *`expression`*

specifies criteria for further qualifying the *`backup-files`*. The *`expression`* qualifier is defined as any of:

```
expression
NOT expression
(expression AND expression ... )
(expression OR expression ... )
```

The use of parentheses within a WHERE *`expression`* is optional. The order of precedence in expression evaluation is: parentheses, NOT, AND, OR.

Possible expressions are:

```
OWNER = user-id
timestamp-field  conditional  time-value
FILECODE conditional number
EOF conditional number
file-attribute
```

`OWNER = user-id`

specifies a NonStop user ID in one of these forms:

```
{ group-name.user-name       }
{ group-name.*               }
{ group-number, user-number  }
{ group-number,*             }
```

timestamp-field conditional time-value

selects a file based on when it was created, last modified, or last opened; or when it expires and can be purged.

timestamp-field is any of:

```
CREATIONTIME
EXPIRATIONTIME
LASTOPENTIME
MODTIME
```

For information on using the WHERE MODTIME qualifier, see PARTIAL on page 3-32.

conditional is any of:

```
<
BEFORE
>
AFTER
```

time-value is:

day [*time*] | [*day*] *time*

day is specified as:

dd mmm yyyy | *mmm dd yyyy*

where *dd* (day) is an integer in the range 1 through 31; *mmm* (month) is one of:

```
JAN, FEB, MAR, APR, MAY, JUN,
JUL, AUG, SEP, OCT, NOV, DEC,
```

and *yyyy* (year) is a 4-digit integer in the range 1900 through 2999.

The default for *day* is today's date.

time is specified as:

hh:mm[:ss]

where *hh* (hour) is an integer in the range 0 through 23; *mm* (minute) and *ss* (second) are 2-digit integers in the range 00 through 59.

Valid *time-value* examples are:

```
1 JAN 2001 06:30
JAN 1 2001 06:30
02 JUL 2001 08:25:30
```

The default *time-value* is 00:00:00 (midnight) of today's date.

`FILECODE` *conditional number*

selects a file based on its file code. The *conditional* qualifier is any of:

```
<
>
<=
=
>=
<> (not equal to)
```

number is a file code such as 101.

`EOF` *conditional number*

selects a file based on its *number* of bytes. The *conditional* qualifier is any of:

```
<
>
<=
=
>=
<> (not equal to)
```

file-attribute

is covered in the next subsection.

File Attributes

To specify file attributes, use a WHERE *expression* qualifier in a file-set list (see Table 3-1 on page 3-10). You can use these attributes with, without, or instead of BACKUP options. Although some file attributes (such as AUDITED, INDEX, PARTITION, and SAFEGUARD) perform the same function as some BACKUP options, you can use additional qualifiers (NOT, AND, and OR) with the file attributes for the WHERE *expression*.

If you specify a file attribute and a similar BACKUP option, the result is identical. For example, you get the same result when you specify the EXCLUDE attribute or the NOT option—or when you specify both. For descriptions of each BACKUP option, see BACKUP Options on page 3-12.

Table 3-1. File Attributes for the WHERE Expression (page 1 of 2)

Attribute	Specifies...
AUDITED	Files audited by the HP NonStop Transaction Management Facility (TMF).
BROKEN	Files marked broken (files that need media recovery because an I/O or a consistency check failure occurred the last time it was open). A likely use is WHERE NOT BROKEN.
CORRUPT	CORRUPT files (files whose contents are in question). For example, FUP marks the destination file corrupt while a DUP or LOAD operation is performed, and leaves it marked if the operation fails. A likely use is WHERE NOT CORRUPT.
CRASHOPEN	Files marked crash-open (files not closed normally by the disk process). A likely use is WHERE NOT CRASHOPEN.
ENSCRIBE	Enscribe files.
ENTRYSEQUENCED	Entry-sequenced files.
FILECODE	A file based on its file code.
FORMAT 1	Files created using an RVU prior to G06 or D46; or non-large-format files created using G06, D46, or later RVUs.
FORMAT 2	Large-format files (created using RVU G06, RVU D46, or subsequent RVUs).
INDEX	SQL indexes.
KEYSEQUENCED	Key-sequenced files.
LICENSED	Object files that contain privileged code and are licensed to be run by user IDs other than 255,255.
OPEN	Files in the open state. To back up open files, you must use the BACKUP OPEN option in addition to this OPEN file attribute. A likely use of OPEN is WHERE NOT OPEN.
[PRIMARY | SECONDARY] PARTITION	The primary partition of partitioned files or secondary partitions of partitioned files. Without this attribute, all partitions of partitioned files are backed up.
PROGID	Files whose PROGID flag is set.
RELATIVE	Relative files.
ROLLFORWARDNEEDED	Files that require a TMF rollforward operation. A likely use is WHERE NOT ROLLFORWARDNEEDED.
SAFEGUARD	Files protected by the Safeguard product.
[SHORTHAND | PROTECTION] VIEW	Shorthand SQL views or protection SQL views. Without this attribute, all SQL views are backed up.
SQL	All types of SQL files, but not SQL program files.
SQLPROGRAM	SQL object program files.
TABLE	SQL tables.

Table 3-1. File Attributes for the WHERE Expression (page 2 of 2)

Attribute	Specifies...
TRUSTED	Files where the FLTrustFlags flag is set to ME or SHARED.
TRUSTME	Files where the FLTrustFlags flag is set to ME.
TRUSTSHARED	Files where the FLTrustFlags flag is set to SHARED.
UNSTRUCTURED	Unstructured disk files.

Note. TRUSTED, TRUSTME, and TRUSTSHARED attributes are supported only on systems running H-series RVUs or J-series RVUs.

File-Mode BACKUP Options

In file mode, the BACKUP options fall into five categories:

- Standard BACKUP options that also can be used with the other four categories:

ARCHIVEFORMAT	MSGONLOCK	NOT	START
AUDITED	MULTIDRIVE	NOUNLOAD	TAPEMODE
BLOCKSIZE	NEEDBOTH	OPEN	VERIFYREEL
DENSITY	NOMYID	PAGELENGTH	VERIFYTAPE
IGNORE	NOPROMPT	PARTONLY	
LISTALL	NOREWIND	REMOTEIOSIZE	

- File conversion BACKUP options for converting files from one tape format to another:

DP1FORMAT	DSLACK	ISLACK
DP2FORMAT	EXT	SCRATCHVOL

- Enscribe file BACKUP options for backing up Enscribe files:

ALTFILE	PART	VOL
NOSAFEGUARD	PARTIAL	

- SQL file BACKUP options for backing up SQL files:

INDEXES	NOSQLDATA	SQLCATALOGS

- DSM/TC BACKUP option, used to catalog information about disk files:

  CATALOGFILES

Note. File mode BACKUP filters out but does not back up HP NonStop Storage Management Foundation (SMF) physical files (files named ZYS*.* or ZYT*.*). These files are not visible to BACKUP. If you attempt a backup for $P.ZYS*.*, BACKUP responds indicating zero files were backed up.

Volume-Mode BACKUP Options

These BACKUP options are available in volume mode:

BLOCKSIZE	NEEDBOTH	TAPEMODE
DENSITY	NOPROMPT	VERIFYTAPE
LISTALL	NOREWIND	VOLUMEMODE
MULTIDRIVE	NOUNLOAD	WHOLEDISC

Note. Any volume-mode BACKUP/RESTORE request to a SMF virtual disk results in the following error message: “Volume Mode BACKUP/RESTORE is incompatible with DSM/SM virtual disks, use File Mode.”

BACKUP Options

This subsection describes file-mode and volume-mode BACKUP options in detail. The options are arranged alphabetically by option name.

ALTFILE

The ALTFILE option changes the name of an alternate-key file in the file label of the primary-key file. This option applies only to Enscribe files.

```
ALTFILE ( key-file-number ,
        [$volume.][subvolume.]file-id )
```

key-file-number

is an integer in the range 0 through 255, inclusive, that identifies the alternate-key file you are naming.

volume.subvolume.file-id

is the new name of the alternate-key file.

Guidelines

- Because the ALTFILE option affects all backup files that have alternate keys, use the ALTFILE option only when you are backing up a single primary-key file.

- You cannot use wild-card characters with this option.

Example

To change the name of an alternate-key file in the file label of the primary-key file AFILE:

```
1> BACKUP $TAPE, $MYVOL.BASE.AFILE, ALTFILE (2, MYFILE6)
```

ALTFILE 2 must already be a defined attribute of $MYVOL.BASE.AFILE.

ARCHIVEFORMAT

The ARCHIVEFORMAT option directs BACKUP to use tape format 3, which supports SQL files, labeled tapes, and parallel copies.

```
ARCHIVEFORMAT
```

Guidelines

BACKUP automatically uses tape format 3 unless DP1FORMAT or DP2FORMAT is specified

For more information, see Tape Formats on page 1-10.

Example

To back up files in the archive format:

```
1> BACKUP $TAPE, ($OLD1.*.*, $OLD2.*.*), LISTALL, ARCHIVEFORMAT
```

This command produces these results:

- All disk files on volumes `$OLD1` and `$OLD2` are backed up on the tape that is mounted to the device `$TAPE`.
- The terminal displays the names of all files backed up, and the names and error messages associated with the files that are requested but not backed up.
- The ARCHIVEFORMAT option causes BACKUP to create the backup tape in tape format 3.

AUDITED

The AUDITED option directs BACKUP to back up files audited by TMF. This option applies to both Enscribe and SQL files.

```
AUDITED
```

Guidelines

- If you omit AUDITED, audited files are not backed up.
- The AUDITED option is often needed to back up SQL tables, which are usually audited.
- The AUDITED option works the same as the AUDITED file attribute in a qualified fileset.
- TMF has its own recovery mechanisms for audited files, but you might want to use BACKUP and RESTORE to:
  - Transport audited files to another system
  - Archive files and retrieve files that are used infrequently
  - Keep old versions of files
- No entries are made in the TMF catalog for audited files that are backed up.

Table 3-2. BACKUP and TMF Interaction While Backing Up Audited Files

BACKUP Options Used	State of File	What BACKUP Does
No AUDITED option	Open or closed	File is not written to tape. Message `Audited File Skipped` is sent to output file.
AUDITED	Closed, or open for shared or protected read	File is written to tape.
AUDITED	Open for write or open exclusive	File is not written to tape. Message `Error 12 - File Aborted` is sent to output file.
AUDITED and OPEN	Open for shared or protected read/write	File is written to tape.
AUDITED and OPEN	Open exclusive	File is not written to tape. Message `Error 12 - File Aborted` is sent to output file.

- BACKUP and RESTORE can be used with the TMFCOM DUMP command for disks containing a combination of audited and nonaudited files or for disks that contain only audited files:

  1. Ensure the safe and complete recovery of files when media recovery is needed. At a SCF prompt, type:

```
1> CONTROL DISK $DATA, REFRESH
2> BACKUP $TAPE, $DATA.*.*,OPEN,AUDITED, &
2> &PARTONLY ON ,LISTALL
3> TMFCOM DUMP FILES $DATA.*.*
```

The SCF CONTROL DISK, REFRESH command updates the file labels and writes all changed buffers to the disk.

The backup includes nonaudited and audited files.

The online dump performed by TMFCOM DUMP includes only audited files, and ensures that there is an accessible copy of these files if TMF fails.

If the disk (in this case $DATA) becomes damaged or unusable, recover the files on a new disk that is properly labeled and initialized using SCF.

2. RESTORE is used to restore the nonaudited files, and TMFCOM is used to recover the audited files. For example:

```
 4> RESTORE $TAPE, $DATA.*.*,OPEN,PARTONLY ON,LISTALL
 5> TMFCOM RECOVER FILES $DATA.*.*
```

The AUDITED option is omitted in the RESTORE command so that only the nonaudited files are restored. After the RESTORE operation, the audited files are recovered using the TMFCOM utility.

For more information on backing up open files, see OPEN on page 3-29.

BLOCKSIZE

The BLOCKSIZE option specifies the size of each tape record (block) written to the backup tape.

```
BLOCKSIZE data-record-size
```

data-record-size

is the number of 1024-byte increments (blocks) in each record. Specify *data-record-size* as 2, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, or 52. The actual blocks written to tape include sequence numbers, checksums, and address information, so they are slightly larger than 1024 * *data-record-size*.

Note. Data transfer sizes larger than 28 KB are only supported by the Napa 3215 and Marathon 3216 controllers. All other transfers must be in the range of 2 KB through 28 KB.

Guidelines

- BLOCKSIZE can be an even value in the range 2 through 52, but BACKUP rounds it down to a multiple of 4 (unless BLOCKSIZE is 2). Rounding is performed because when DP2 tapes are to be generated (tape format 2 or greater) and BLOCKSIZE is 4 or greater, BACKUP and the DP2 disk process perform integrity checking by reading whole-structured file blocks of length BLOCKSIZE * 1024. The longest structured file block length is 4096. BLOCKSIZE must be a multiple of 4 so that the long reads will contain an integral number of whole-structured blocks.

- The default *data-record-size* on NonStop operating systems is 8 (8192 bytes); the default on NonStop 1+ systems is 2 (2048 bytes).
- You can back up files on a NonStop 1+ system and restore those files on a NonStop operating system with no special command options. RESTORE detects the difference and adjusts for the smaller size. However, you must specify BLOCKSIZE 2 and include the DP1FORMAT option when you create a backup tape on a NonStop operating system to be restored later on a NonStop 1+ system.
- A tape backed up with a 52 KB BLOCKSIZE can only be restored with a D30 (or later) version of RESTORE on a tape drive that supports 56 KB transfers.
- Optical drives do not support transfers using the 56 KB BLOCKSIZE option.
- If you want to use the 52 KB BLOCKSIZE option in a multireel environment, each tape drive must support 56 KB transfers.
- The EXPAND environment supports up to a 56 KB BLOCKSIZE. To use this feature, the source and destination nodes must have the enhancement installed. Check your EXPAND environment for BLOCKSIZE support.

CATALOGFILES

The CATALOGFILES option enables you to catalog information about disk files.

```
CATALOGFILES every-n-files
```

every-n-files

is the number of disk files in the range of 1 through 20000. The default value is 20000. After *every-n-files* is processed, the disk file entries will be made for each disk file written on to the tape.

△ **Caution.** If the value of *every-n-files* is below the specified range when backing up large number of files with small size, *every-n-files* will hit numerous catalog updates in a short period of time, which could affect the performance of the BACKUP utility.

This option is recommended for use during a file-mode BACKUP process. For examples and more information, see the *DSM/Tape Catalog User's Guide*. Do not use this option during a volume-mode BACKUP process.

Guideline

BACKUP sends this message to your user terminal while DSM/TC is being updated:

```
Please wait -- Backup is sending updates to DSM/TC
```

When processing is complete, this message is sent to your terminal:

```
DSM/TC processing has completed
```

The successful messages is displayed either when you specify the value for *every-n-files* as 20000 or no value is specified for *every-n-files*.

DENSITY

The DENSITY option specifies the tape recording density.

```
DENSITY density
```

density

is the tape recording density:

Density Specified	Recording Density in Bits per Inch
GCR or 6250	6250
PE or 1600	1600
NRZI or 800	800

Tape drives supported on D-series and G-series do not include an 800-bpi density.

Guidelines

- You cannot use the DENSITY and TAPEMODE options in the same BACKUP command.
- Generally, the higher the density specified, the shorter the backup time and the less tape used to complete the backup. However, you might need to specify a lower density value in the backup to offset the possible use of a tape drive (during RESTORE) that does not support a high-density value.
- If you enter the DENSITY option for a tape drive that does not support programmatic density selection, this warning is displayed and the BACKUP procedure continues:

```
*WARNING* Tape Drive does not support DENSITY
selection.
```

- For information on the densities that your tape drive supports, see the manual for that tape drive.
- If DENSITY is specified and *tape-device-name* is a CLASS TAPE DEFINE name, the DENSITY attribute of the DEFINE must either be unspecified or must match *density*.
- The default density is the physical setting of the tape drive.

DP1FORMAT

The DP1FORMAT option directs the BACKUP utility to write all files to tape in DP1 format (tape format 1).

```
DP1FORMAT
```

Guidelines

- Any DP2 files encountered are converted on disk into temporary DP1 files and then written to tape in DP1 format.
- Because SQL files can be backed up only with tape format 3, any SQL files named in *backup-files* are skipped.
- When performing file conversion, BACKUP creates temporary disk files on the current subvolume. These files, which are in the format of the destination tape, have names that begin with ZZCV. If you terminate the BACKUP procedure early, your current subvolume might contain some of these temporary files. Purge any ZZCV files to regain disk space.
- You can back up files on a NonStop 1+ system and restore those files on a NonStop operating system with no special command options. However, when creating a backup tape on a NonStop operating system to be restored on a NonStop 1+ system, you must specify DP1FORMAT and BLOCKSIZE 2.

DP2FORMAT

The DP2FORMAT option directs BACKUP to write all files to tape in DP2 format (tape format 2).

```
DP2FORMAT
```

Guidelines

- Any DP1 files encountered are converted on disk into temporary DP2 files and then written to tape.
- Because SQL objects can be backed up only with tape format 3, any SQL files named in the *backup-files* are skipped.
- For DP2 key-sequenced files, index blocks must be the same size as data blocks; this is not true for DP1 files. When you use the BACKUP utility to convert key-sequenced files from DP1 to DP2 format, the index-block size is changed if it is not equal to the data-block size. This block change does not apply to relative or entry-sequenced files.
- You can use BACKUP to copy an entry-sequenced DP1 file to a DP2 file on tape. However, if the DP1 file does not have a DP2 block size, the block size is rounded

up to a size compatible with DP2 (that is, 2048 or 4096 bytes). Block sizes not compatible with DP2 include 1536, 2560, 3072, and 3584 bytes. This operation changes record addresses in the file. If BACKUP converts a primary DP1 entry-sequenced file to DP2 format, the program displays a warning instructing you to perform a FUP LOADALTFILE operation on all the alternate-key files of this primary file. For instructions on using the FUP LOADALTFILE command, see the *File Utility Program (FUP) Reference Manual*.

- Although you can specify the PARTONLY option, BACKUP backs up entire partitioned files that have secondary partitions on remote systems (if you include the DP2FORMAT option).
- For more information, see Backing Up or Converting DP1 and DP2 Files on page 3-49.

DSLACK

The DSLACK option sets the minimum percentage of slack space in data blocks. It is used only when converting key-sequenced Enscribe files between DP1 and DP2 format.

```
DSLACK percentage
```

percentage

is an integer in the range 0 through 99. The default value is 0.

EXT

The EXT option sets the extent sizes of the copy to be made on tape. Used only when converting files between DP1 and DP2 format.

```
EXT { extent-size                          }
    { ( pri-extent-size, sec-extent-size ) }
```

Note. The use of extent sizes over 65535 requires format 2.

You can specify these values for *extent-size*, *pri-extent-size* (primary extent size), and *sec-extent-size* (secondary extent size):

```
0:33,554,432 [ PAGE[S] ]
```

specifies the extent size in pages (2048-byte units). The minimum extent size is one page, so specifying 0 pages allocates one page (2048 bytes). The PAGE is the default unit of measurement for the EXT option.

`0:68,719,476,735 BYTE[S]`

specifies the extent size in bytes. BACKUP rounds up to the next full page (2048-byte units). For example, if you specify 2047 bytes, BACKUP allocates one page; for 2049, BACKUP allocates two pages, and so on.

`0:68,719,476,735 REC[S]`

specifies the extent size based on the current settings for record length, data-block length, index-block length, key-field lengths, and compression settings. BACKUP rounds up to the next full page.

Guidelines

- You cannot include the EXT option with the PARTONLY option.
- If BACKUP returns file-system error 45 (file is full) for a key-sequenced file, back up that file separately with the EXT option to specify larger extent sizes. Error 45 can appear when the specified slack values cause significant file size increases in converted key-sequenced files.
- If BACKUP returns error 45 for a key-sequenced and partitioned file (because the default extent size of the temporary disk file used by BACKUP is too small), use the EXT option to specify larger extent sizes.

Examples

- To convert DP1 files to DP2 format and set primary and secondary extent sizes to two and three pages (respectively) for the files in $DP1.CONV:

```
1> BACKUP $TAPE, $DP1.CONV.*, DP2FORMAT, EXT (2,3)
```

- To set the data slack and index slack values for a DP1 to DP2 file conversion and set primary and secondary file extents to three:

```
 2> BACKUP $TAPE1, *.*.*, DP2FORMAT, DSLACK 20, ISLACK
 10,&
 2> &EXT 3
```

IGNORE

The IGNORE option directs BACKUP to ignore certain data errors on disk. BACKUP writes the invalid data to tape (if possible); otherwise, zeros are written in place of invalid data.

```
IGNORE
```

Guidelines

- Specify the IGNORE option to back up a file that is marked as corrupt.
- If you omit IGNORE and a data error occurs, BACKUP skips the file and starts processing the next file.
- BACKUP ignores errors in DP1 files by sectors (a sector is 512 bytes) and sends an error message to the output file for each defective sector. For DP2 files, BACKUP ignores errors on a block-by-block basis. For structured files, the length of this block is the data-block length. For unstructured files, this block is the buffer length.

INDEXES

The INDEXES option specifies whether the indexes defined for SQL tables are automatically backed up when the tables are backed up.

```
INDEXES [ IMPLICIT | EXPLICIT ]
```

`IMPLICIT`

specifies that indexes defined for a table are backed up automatically when the table is backed up. This is the default when you specify INDEXES, and when you omit INDEXES from the BACKUP command when combined with PARTONLY OFF.

`EXPLICIT`

specifies that indexes defined for a table are not backed up automatically when the table is backed up. Only those indexes explicitly named in the backup files are backed up. EXPLICIT is the default and only allowed value when PARTONLY ON is specified in BACKUP.

Examples

- To back up all files and SQL tables (except those that are open exclusively) on the volume $FIN:

  ```
  1> BACKUP $TAPE1, $FIN.*.*, INDEXES IMPLICIT, AUDITED
  ```

  For SQL tables that are backed up, all indexes are also backed up, no matter what volume the indexes are on. Indexes on $FIN that are defined for base tables that are not on $FIN are not backed up.

- To back up all files and SQL tables (except those that are open) on the volume $FIN:

  ```
  2> BACKUP $TAPE1, $FIN.*.*, INDEXES EXPLICIT, AUDITED
  ```

  For SQL tables that are backed up, only the indexes on $FIN are backed up.

ISLACK

The ISLACK option sets the minimum percentage of slack space in index blocks. Used only when converting key-sequenced Enscribe files between DP1 format and DP2 format. The default value is 0.

```
ISLACK percentage
```

percentage

is an integer in the range 0 through 99. The default value is 0.

LISTALL

The LISTALL option lists the names of all backed up files, and the names of the files that cause errors. If you omit LISTALL, BACKUP lists only the file names associated with error messages.

```
LISTALL
```

Guidelines

If the BACKUP command includes the LISTALL option, the output listing includes information about all backup files.

Figure 3-1. Listing Format

```
System:  \<nodename>
Tape:  $<tape>    Operating System:  <vsn>   Tape Version:  <v>
Backup options:  <params>
[            *WARNING*  <comment>                          ]
                  .       .
                  .       .
[Partial Time:  <date> <time> ]
Backup time:  <date> <time>                                          Page:  <n>

Tape:  <n>      Code     EOF     Last modif     Owner     RWEP     Type     Rec Bl

$<vol>.<subvol>
 <filename>    <code>   <eof>   <date> <time>   <g,u>    <rwep>   <type>   <recbl>
      .           .       .        .     .        .         .        .         .
      .           .       .        .     .        .         .        .         .
Summary Information

Files dumped = <n>   Files not dumped = <n>
```

Figure 3-2. Listing Format Sample Output

```
backup $tape, $ski.rbstestb.*, listall
File Mode BACKUP Program - T9074D46 (07SEP98)
Copyright Tandem Computers Incorporated 1981-1998
Drives: ($TAPE)
System: \LAB1  Operating System: G06  Tape Version: 3
Backup options:  NO AUDITED, BLOCKSIZE 8, NO IGNORE, NO OPEN, PARTONLY OFF,
                 INDEXES IMPLICIT
*WARNING-7033*  This tape can only be restored with TNS/II RESTORE (B41, C00 or
                 later).
Backup time: 10Dec1998 14:31                                          Page: 1
Tape: 1       Code          EOF        Last modif   Owner RWEP   Type   Rec Bl

$AEP.DANCE
ECSWNG                       0    03Dec1998 11:22    -1   NUNU   E      200  4
WCSWNG       101             0    03Dec1998 11:22    -1   NUNU

$AEP.DANCE
TANGO                        0    03Dec1998 11:22    -1   NUNU   PK      80  2
$AEP.MUSIC
LATIN                        0    03Dec1998 11:22    -1   NUNU  XPK      80  2

$AEP.DANCE
RUMBA                        0    03Dec1998 11:22    -1   NUNU    K      18  2
SAMBA                        0    03Dec1998 11:22    -1   NUNU    K      18  2
$AEP.DANCE
FXTRT            +           0    03Dec1998 11:22    -1   NUNU   PK      80  2
$AEP.MUSIC
BGBND            +           0    03Dec1998 11:22    -1   NUNU  XPK      80  2

Summary Information

Files dumped = 6  Files not dumped = 0
```

The BACKUP listing generated when LISTALL is used will include whatever of this information is relevant to the specific backup operation:

`Tape: [\`*node*`.]$`*tape*	Name of the tape drive used.
`Operating System:` *vsn*	Version of the NonStop host.
`Tape Version:` *v*	Tape format in use with this operating system.
`Backup options:` *params*	Command options that might affect the restoration of these files.
`Partial Time:` *date*`/`*time*	Date and time specified with the PARTIAL option.
`Backup time:` *date*`/`*time*	Date and time of the backup.
`Page:` *n*	Page number of the listing.
`Reel:` *n*	The number of the magnetic tape reel or tape cartridge where the file was written.
`$`*vol*`.`*subvol*	Volume and subvolume of files.
filename	Files backed up. If the letter C follows the file ID, the file is corrupt. If B follows the file ID, the file is broken and needs media recovery.

`CODE` *code*	File code of the listed file. If the letter A follows the code, the file is audited by TMF. If L follows the code, the file is licensed. If P follows the code, the file has the PROGID attribute set. If M follows the code, the state of the FLTrustFlags is ME. If S follows the code, the state of the FLTrustFlags is SHARED.
`EOF` *eof*	End-of-file value (the size of the file in bytes).
`Last modif` *date/time*	Date and time that the listed file was last modified.
`Owner` *g,u*	Group and user ID of the owner of the listed file.
`RWEP` *rwep*	Security of the listed file for read, write, execute, and purge access. Four asterisks (****) indicate that the listed file is protected by Safeguard.
`Type` *type*	File type of the listed file. The *type* can be any of: blank = unstructured R = relative E = entry-sequenced K = key-sequenced Ta = SQL table In = SQL index SVi = SQL shorthand view PVi = SQL protection view A = (suffix) file with alternate key P = (prefix) partitioned file XP = (prefix) file is an extra (secondary) partition
`Rec Block` *recbl*	A file's logical record and block lengths (in kilobytes).
`*ERROR*` *comment*	Any error that prevents normal back up of a file.
`*WARNING*` *comment*	Warning about the file being backed up.
`No. of volumes skipped =` *n*	Number of volumes that were skipped; appears only if relevant to the specific backup operation.
`No. of file sets not matched =` *n*	Number of file sets in the specified *qualified fileset-list* that contained no files; appears only if relevant to the specific backup operation.
`Tape related warnings =` *n*	Warnings related to the tape device; appears only if relevant to the specific backup operation.

Note. The FLTrustFlags attribute is supported only on systems running H-series RVUs or J-series RVUs.

MSGONLOCK

The MSGONLOCK option lets you back up or skip a file that is locked by responding to a message.

```
MSGONLOCK
```

Guidelines

- When MSGONLOCK is specified, BACKUP displays this message at your terminal (if a file you request to be backed up is locked during the BACKUP procedure):

```
File or record lock encountered in file: filename
Enter SKIP to skip the file or carriage return to wait for
unlock?
```

- Enter SKIP to skip the locked file and continue immediately with the backup. Any part of the file that was written before BACKUP detected the locked condition is then erased from the tape. For example, if the first file being backed up is locked and you type SKIP, BACKUP skips the file and continues with the second file.
- MSGONLOCK does not apply to open files that are not locked.
- Because bulk I/O is used for DP2 files, record locks do not cause this message to be displayed when you are backing up DP2 files.

MULTIDRIVE

The MULTIDRIVE option lets you have up to four tape drives queued for unlabeled-tape BACKUP operations. The sequence of *tape-device-names* specified in the BACKUP command determines the order in which BACKUP writes to the tapes.

```
MULTIDRIVE
```

Guideline

If a tape drive is unavailable or off-line, or if a tape is not mounted when the BACKUP utility tries to write to that drive, BACKUP polls the drive and displays this message:

```
   $tape: device not ready -- beginning to poll --
```

The BACKUP process continues in this state until it detects that the drive is ready, with a tape mounted.

Example

To back up a four-reel tape set using two tape drives:

```
1> BACKUP ($TAPE1, $TAPE2, $TAPE3, $TAPE4), *.*.*,
MULTIDRIVE
```

BACKUP writes to alternate tape drives until the tape set is finished. In this example, BACKUP is set to write to tape #1 on $TAPE1, tape #2 on $TAPE2, tape #3 on $TAPE1, and tape #4 on $TAPE2.

NEEDBOTH

The NEEDBOTH option lets you terminate a parallel BACKUP if an error occurs on one of the tapes.

```
NEEDBOTH
```

Guideline

Without the NEEDBOTH option, you cannot stop a parallel BACKUP if one of the tape operations fails. The parallel BACKUP continues with one drive producing one copy instead of the two that were requested.

Example

To terminate the parallel BACKUP if an error occurs on one of the tapes:

```
1> BACKUP ($TAPE1, $TAPE2, NEEDBOTH), *.*.*, LISTALL
```

BACKUP writes to alternate tape drives until the tape set is finished. In this example, BACKUP is set to write to tape #1 on $TAPE1 and tape #2 on $TAPE2.

NOMYID

The NOMYID option disallows the use of the MYID option from RESTORE, which lets you restore files that originally belonged to another user onto your user ID.

```
NOMYID
```

If the NOMYID option is specified, backup tapes are marked so that the MYID option cannot be used during a RESTORE procedure.

△ **Caution.** Although the NOMYID option provides greater security, it is not a substitute for a careful security plan with restricted physical access to confidential disk and tape information.

NOPROMPT

The NOPROMPT option instructs BACKUP not to prompt the user before beginning to write on each tape, but to begin when it detects the tape drive is ready.

```
NOPROMPT
```

Example: Mounting a Tape for a BACKUP using NOPROMPT

1. Before starting a BACKUP process, insert a write-enable ring on each tape reel.
2. Enter a complete BACKUP command, including the NOPROMPT option.

   This noninteractive message is displayed:

```
$tape: device not ready -- beginning to poll --
```

3. Mount the first (or next) tape. For each reel, the program continues when it detects that the drive is ready.
4. When there are no more device not ready prompts, the process is complete.

NOREWIND

The NOREWIND option directs the BACKUP utility to leave the tape positioned at its current location and leave it online when the BACKUP process is completed. This option lets the tape be labeled for the next BACKUP without having to search for the end of tape. This option is only available starting with the G06 version of BACKUP.

```
NOREWIND
```

NOREWIND is mutually exclusive with NOUNLOAD. If neither is specified the default is to unload the tape.

If NOREWIND is specified for an unlabeled tape and an attempt is made to read (RESTORE/BACKCOPY) or write (BACKUP/BACKCOPY) to the tape without first manually repositioning the tape, the utility abends with Error 8103.

NOSAFEGUARD

The NOSAFEGUARD option excludes Safeguard security information in a BACKUP process. This option applies only to Enscribe files.

```
NOSAFEGUARD
```

Guidelines

- If you use the NOSAFEGUARD option, files with Safeguard security information are backed up but do not retain Safeguard protection.

- When you back up files that have Safeguard protection and do not use the NOSAFEGUARD option, the files retain that protection.

NOSQLDATA

The NOSQLDATA option makes BACKUP record only the SQL file label for all SQL files in the qualified file set. It does not skip the SQL files entirely, it just skips the data transfer portion of the BACKUP process. NOSQLDATA sets the EOF in the file labels for all SQL files on the tape to 0, so the resulting BACKUP tape contains the DDL information necessary to re-create an empty version of the SQL object, without any of the system's SQL data.

```
NOSQLDATA
```

Guidelines

- Use this option to create a BACKUP tape that can be used for emergency recovery of SQL metadata for objects that are too large for data archiving and where the data itself is not mission critical.
- This option puts ENSCRIBE files included in the qualified file set on the BACKUP tape with all the data. Only SQL data is omitted.
- An SQL object can effectively be copied using a NOSQLDATA BACKUP tape and the RESTORE MAP NAMES and CATALOG(S) options. This provides a create-like function for multiple partition objects.
- Since a NOSQLDATA BACKUP contains no sensitive customer data, it is more manageable for development debugging of DDL and file label problems than a full data tape.
- This option cannot be used in the same BACKUP command as the SQLCATALOGS option.
- When you use this option, this warning message is displayed at BACKUP time:

```
*WARNING-7157*  The NOSQLDATA option is active. No SQL data
can be recovered using this BACKUP tape.
```

NOT

The NOT option excludes the files named in the *not-fileset-list* from the BACKUP process. That is, all of the files specified in *backup-files*, except those files specified in *not-fileset-list*, are backed up.

```
NOT not-fileset-list
```

not-fileset-list

is specified the same as a file set or file-set list. See File Sets, File-Set Lists, and Qualified File-Set Lists on page 3-5.

Guideline

The NOT option is equivalent to the EXCLUDE qualifier in the qualified file set syntax. See File-Set List Qualifiers on page 3-7.

Example

To back up all the files in the system (except the files on the volume $SYSTEM):

```
11> BACKUP $TAPE, *.*.*, NOT $SYSTEM.*.*, LISTALL
```

NOUNLOAD

The NOUNLOAD option directs BACKUP to rewind the final tape and leave it online when the BACKUP process is completed.

```
NOUNLOAD
```

△ **Caution.** Using BACKUP with the NOUNLOAD option leaves the tape online and write-enabled. Subsequent BACKUP commands (either from you or another user) could write over the data that was just backed up. If you are backing up critical data, do not use the NOUNLOAD option unless there is no possibility that the tape can be inadvertently overwritten. The use of the NOREWIND option will prevent another process from using the drive until the drive is manually reset. The use of labeled tapes with a TAPECATALOG DEFINE can also prevent other users from overwriting the tape.

NOUNLOAD is mutually exclusive with NOREWIND. If neither is specified the default is to unload the tape.

OPEN

The OPEN option directs BACKUP to back up files even if they are currently open with write or read/write access unless the files are also open with exclusive access.

If you do not include the OPEN option, the BACKUP process skips the files that are currently open with write or read/write access.

```
OPEN
```

△ **Caution.** If you restore a file that was modified or open for write access while being backed up, file-system error 59 (file is bad) can occur, and data can be lost. Whenever possible, close all files before running BACKUP. If audited files are backed up in the WRITE OPEN state, they could be corrupt. Restoring such files and using them in audited mode can cause TMF to leave the whole volume in an inconsistent state. If audited files are restored, they should be used in nonaudited mode. Use TMF to dump audited files to tape.

Guidelines

- BACKUP normally tries to open files with protected read access, meaning that other processes can read the files but cannot write to them. If the OPEN option is in use and BACKUP fails to open a file with protected access, BACKUP then attempts to open the file with shared read access.
- For additional information about backing up files that are open and audited, see AUDITED on page 3-13.
- Before backing up files with the OPEN option, issue an SCF CONTROL DISK, REFRESH command to update file labels on disk. Then be certain the files being backed up are not modified while BACKUP is running.

Example

To back up all files on the current volume, excluding audited files and including open files:

```
6> BACKUP $TAPE, *.*.* WHERE NOT AUDITED, OPEN, LISTALL
```

PAGELENGTH

The PAGELENGTH option specifies the number of lines per page of output from the BACKUP procedure.

```
PAGELENGTH number
```

number

is an integer in the range 20 through 100 that specifies the number of lines per page of output from the BACKUP procedure.

Guidelines

- When the BACKUP output is sent to a printer, a form feed or page eject is generated after the number of lines specified in the PAGELENGTH option.
- If PAGELENGTH is not specified, the output from the BACKUP procedure defaults to 60 lines per page.

Example

To direct the output from the BACKUP procedure to a printer, and to print 50 lines per page:

```
1> BACKUP /OUT $S.#BOOK/ $TAPE, $MYVOL.*.*, PAGELENGTH 50
```

PART

The PART option specifies a new node or volume name (or both) for a secondary partition of a partitioned file. This option causes the secondary partition name to be inserted in the file label of the primary partition, and it applies only to Enscribe files.

```
PART ( sec-partition-num , [ \node.][$volume]
    [ , pri-extent-size , [ sec-extent-size ] ] )
```

sec-partition-num

is an integer in the range 1 through 15, inclusive, that designates the secondary partition. This number was assigned to the partition when the partitioned file was created.

node

is the name of the node (system) where the secondary partition is to reside. If you omit the node name, BACKUP uses the original node name of the secondary partition.

volume

is the name of the volume where the secondary partition is to reside. If you omit the volume name, BACKUP uses the original volume name of the secondary partition.

pri-extent-size
sec-extent-size

defines the primary and secondary extent sizes, respectively. The default value is 1 page (2048 bytes). You can specify the following values for *pri-extent-size* and *sec-extent-size*:

`0:33554432 [ PAGE[S] ]`

specifies the extent size in pages (2048-byte units). The minimum extent size is one page, so specifying 0 pages allocates one page (2048 bytes). The PAGE is the default unit of measurement.

`0:68719476735 BYTE[S]`

specifies the extent size in bytes. BACKUP rounds up to the next full page. For example, if you specify 2047 bytes, BACKUP allocates one page; for 2049, it allocates two pages, and so on.

`0:68719476735 REC[S]`

specifies the extent size based on the current settings for record length, data-block length, index-block length, key-field lengths, and compression settings. BACKUP rounds up to the next full page.

△ **Caution.** The PART option affects all partitioned files in the *backup-files*. There might be more than one partition with the specified *sec-partition-num*, so use the PART option carefully if you back up more than one partitioned file.

Guidelines

- When you specify the PART option, the original name of the partition indicated by *sec-partition-num* is replaced by the new partition name you specify. The *sec-partition-num* must already exist in the source file.
- You can use the PART option to specify the destination partition extent sizes when converting files from one disk-process type to another.
- You cannot specify extent sizes with the PART option if your BACKUP command also contains the PARTONLY ON option.
- If you include the PART and PARTONLY ON options, BACKUP changes the name of the secondary partition in the file label of the primary partition. If you include the PART option without PARTONLY ON, BACKUP changes both the secondary-partition name in the primary-partition file label and the actual file name of the secondary partition.
- You cannot use wild-card characters with this option.

Example

To back up all partitions of the file $PUBS.BOOKS.SECT1:

```
1> BACKUP $TAPE, $PUBS.BOOKS.SECT1, PART (2, $SAVE)
```

When this file is restored, partition 2 is restored to the volume $SAVE instead of the volume on which it currently resides.

PARTIAL

The PARTIAL option directs BACKUP to back up only the *backup-files* that have been modified since *partial-dump-date*. This option applies only to Enscribe files. All SQL objects in the *backup-files* are backed up even if they have not been

modified since the *partial-dump-date*. All file labels in the *backup-files* are also backed up when you use the PARTIAL option.

```
PARTIAL partial-dump-date
```

partial-dump-date

is either one of:

```
{ day month } year , hour:minute
{ month day }
```

day

is a one- or two-digit day of the month, such as 1, 02, 15, or 31.

month

is the first three letters of the month name, such as JAN, FEB, JUN, JUL, DEC.

year

is a four-digit year, such as 1986, 1991.

hour

is a one-digit or two-digit number, such as 0 for midnight, 1 for 1 a.m., 22 for 10 p.m.

minute

is a two-digit number, such as 00 for the hour, 20 for twenty minutes after the hour, and 55 for fifty-five minutes after the hour.

Guidelines

- The PARTIAL option cannot be used in a BACKUP command in a DP1/DP2 file conversion.
- Whenever the PARTIAL option is used, some modified partitions are likely backed up while unmodified partitions of the same file are not. Read the listing to see which partitions were backed up to tape and carefully RESTORE the ones you want with PARTONLY ON.
- The PARTIAL option copies to tape all SQL tables and indexes, regardless of their modification times. This can affect the backup time and number of tape reels required.
- Only the files that have been modified since the date/time specified in your command are backed up; however, BACKUP uses the last CREATIONTIME values for files as the qualifier. You can have a file with a more recent CREATIONTIME than MODTIME (for example, FUP DUP A, B, SOURCEDATE).

In this example, file B retains the last MODTIME value from file A, but it also has a new CREATIONTIME. A partial backup that targets file B would have to precede its MODTIME (not its CREATIONTIME).

- The PARTIAL option copies to tape the file label for each file that is not backed up. That is, in addition to the files modified since the *partial-dump-date*, a partial backup tape contains file label information for files that have not been modified.
- Use the WHERE MODTIME and CREATION-TIME qualifiers instead of the PARTIAL option to create a tape that contains only the files created or modified after a specific date. For syntax instructions, see File-Set List Qualifiers on page 3-7. Using these qualifiers can decrease the backup time and number of tape reels required for a partial backup.
- Use the RESTORE REBUILD option to restore a file set that was backed up using the PARTIAL option. For more information, see Section 7, RESTORE. A file set that was backed up using the WHERE MODTIME qualifier cannot be restored with the REBUILD option.

  Restore the last partial tape set before you try to restore a file set that was backed up using the WHERE MODTIME qualifier. Then restore the previous tape set using the KEEP option. Continue restoring tape sets using the KEEP option until the last full backup tape set is restored.
- Using the PARTIAL and CATALOGFILES options together causes an entry to be made for each file label that is backed up to tape in the DSM/TC database. This makes the DSM/TC database too large and prevents entries to the database from being recovered automatically. Use the WHERE MODTIME or CREATION-TIME qualifiers if the CATALOGFILES option is used to prevent overcrowding in the DSM/TC database.

Example

To selectively back up all of the files in all of the subvolumes on the volume $STORE1 that were modified since 3 a.m. on August 24, 2001:

```
1> VOLUME $STORE1
2> BACKUP $TAPE, *.*, PARTIAL 24 AUG 2001, 3:00, &
2> &LISTALL
```

PARTONLY

The PARTONLY option specifies whether all partitions of a partitioned file are backed up. This option applies to both Enscribe and SQL files.

```
PARTONLY [ ON | OFF ]
```

ON

specifies that only the partitions of a file explicitly named in *backup-files* are backed up. ON is the default if you specify PARTONLY in the BACKUP command.

OFF

specifies that if the primary partition is specified in *backup-files*, then both the primary and secondary partitions are backed up. OFF is the same as omitting PARTONLY from the BACKUP command.

Guidelines

- If you specify PARTONLY ON, you cannot specify INDEXES IMPLICIT.
- If you specify PARTONLY ON in your BACKUP command, PARTONLY ON is automatically applied during the RESTORE operation for those files.
- If you name the primary partition of a file in *backup-files*, PARTONLY ON specifies that only that particular partition (in this case, the primary partition) is backed up. The secondary partitions are not backed up unless they are explicitly named in *backup-files*.
- If you name the primary partition of a file in *backup-files*, PARTONLY OFF specifies that all the secondary partitions of the file are backed up. That is, if you name the primary partition in *backup-files*, the entire partitioned file is backed up even though the secondary partitions are not named in the *backup-files*.
- If you name a secondary partition of a file in *backup-files*, but not the primary partition, PARTONLY OFF specifies that the secondary partition is not backed up.
- When converting files from one disk-process type to the other, using BACKUP with the PARTONLY ON option skips all relative files and any DP1 entry-sequenced files that do not have DP2 block sizes. For example, blocks of 1536, 2560, 3072, or 3584 bytes are not compatible with DP2. An error message gives the names of the skipped files. You must back up these files without the PARTONLY option.
- If you specify PARTONLY ON, BACKUP backs up entire partitioned files that have some secondary partitions on remote nodes if you include the DP2FORMAT option. BACKUP creates the backup tape in format 2 and converts any DP1 files it encounters to DP2 format.
- You cannot specify extent sizes with the PART option if your BACKUP command also contains the PARTONLY ON option.
- If you include the PART and PARTONLY ON options, BACKUP changes the name of the secondary partition in the file label of the primary partition. If you include the PART option without the PARTONLY option, BACKUP changes both the secondary-partition name in the primary-partition file label and the actual file name of the secondary partition. (This case applies only to Enscribe files because the PART option cannot be used with SQL files.)
- Do not use this option if you plan to use the RESTORE PARTOF option when you recover the files. If you use the PARTONLY ON option during BACKUP and then use the PARTOF option during RESTORE, RESTORE fails.

- When you later use RESTORE, you can specify PARTONLY with MAP NAMES or CATALOG[S] if you also specify KEEP.

Examples

- To back up all files on the volume $BOOKS (including primary and secondary partitions on $BOOKS, but not partitions on other volumes):

```
1> BACKUP $TAPE1, $BOOKS.*.*, PARTONLY ON, AUDITED, LISTALL
```

- To back up all files on the subvolume MORE on the volume $WORDS:

```
2> BACKUP $TAPE2, $WORDS.MORE.*, PARTONLY OFF, LISTALL
```

  Primary partitions on $WORDS.MORE and secondary partitions on other volumes whose primary partition is on $WORDS.MORE are backed up. Secondary partitions on $WORDS.MORE (whose primary partitions reside elsewhere) are not backed up.

  This command without the PARTONLY OFF option would give the same results.

REMOTEIOSIZE

The REMOTEIOSIZE option specifies the maximum size of each data block transferred between systems during a BACKUP process. This option is needed mainly for SQL files where the small default remote I/O transfer size can create a performance problem.

```
REMOTEIOSIZE data-block-size
```

data-block-size

is the maximum number of 1024-byte increments (blocks) in each remote I/O transfer, specified as 2, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, or 52.

Guidelines

- REMOTEIOSIZE values above four (4096 bytes) cause RESTORE to use BulkIO logic.
- The default SQL remote I/O transfer size on the NonStop operating system is 4 KB (4096 bytes). Local I/O transfer size for SQL objects is 28 KB.
- Use of the BLOCKSIZE option when creating the BACKUP limits the available REMOTEIOSIZE values.

  For example, specifying a BLOCKSIZE of 28 in the BACKUP command limits the remote I/O read operation transfers to 28 KB blocks even if you specify a REMOTEIOSIZE of 52 in your RESTORE command. When restoring, REMOTEIOSIZE cannot increase the remote I/O write operation transfer size beyond what BACKUP put on the tape. Therefore, a REMOTEIOSIZE greater than

the BLOCKSIZE is reduced internally. If the REMOTEIOSIZE is greater than BLOCKSIZE the warning message 7158 is sent.

- BLOCKSIZE values larger than 28 KB are only supported by the 3215 and 3216 controllers. All other transfers must be in the range of two to 28 KB. BLOCKSIZE can only be used if both the backup and restore are done on the same type of drive.
- The 52 KB limit is due to current tape BLOCKSIZE limitations. Expand can handle data transfers larger than 52 KB.
- Older Expand environments might not handle 52 KB transfer sizes. Different qualified file sets might involve systems with different *Expand* limitations, and the systems at RESTORE time might be different than those that created the BACKUP tape. For these reasons, individual limitations and automatically limited remote I/O size cannot be identified. If the REMOTEIOSIZE exceeds current hardware or software limitations, RESTORE returns this error:

  ```
  TABLENAME    *ERROR*  Guardian error: 21 (BulkWrite)
  ```

  You must repeat the operation using a smaller REMOTEIOSIZE.
- The default ENSCRIBE I/O transfer size is equal to BLOCKSIZE, but you can use REMOTEIOSIZE to reduce it if you need to because of Expand limitations.

SCRATCHVOL

The SCRATCHVOL option specifies a disk volume where BACKUP is to create the temporary files used in the DP1-to-DP2 conversion process. Use this option when converting nonpartitioned files or individual partitions of partitioned files.

```
SCRATCHVOL $volume
```

$volume

is the name of a disk volume.

Guidelines

- If you omit this option, BACKUP creates temporary files (with prefix ZZCV) on the current default volume and subvolume.
- Use this option if disk space is inadequate during a file conversion (if BACKUP returns error 43).
- You cannot use wild-card characters with this option.

SHAREOPEN

The SHAREOPEN option causes files to be backed up with shared-read access instead of protected-read access. This gives other processes read/write access to the backup files during BACKUP procedures.

```
SHAREOPEN
```

△ **Caution.** If you restore a file that was modified while being backed up, file-system error 59 (file is bad) can occur, and data can be lost. File inconsistencies are not detected during BACKUP or RESTORE but might be discovered when the file is subsequently accessed.

Note. Backup always opens the ZTMFAT subvolume with SHAREOPEN option, irrespective of whether SHAREOPEN has been specified or not.

Guidelines

- Because use of the SHAREOPEN option presents some risk to data integrity, this message appears when the option is used:

  ```
  *WARNING-7139* The backup files may be opened for write
  because the SHAREOPEN option was used for Backup. This may
  result in inconsistent file data on the backup tape.
  ```

  This warning message also appears before a BACKCOPY or RESTORE that uses a BACKUP tape made by using the SHAREOPEN option.

- You cannot use OPEN and SHAREOPEN in the same BACKUP command.
- Backup always opens TMF audit trail files with SHAREOPEN access irrespective of whether SHAREOPEN access has been specified or not.

Example

To back up all files on the current volume (excluding key-sequenced files) and allow shared-read access to the backup files:

```
1> BACKUP $TAPE, *.*.* WHERE NOT KEYSEQUENCED, SHAREOPEN, &
1> &LISTALL
```

SQLCATALOGS

The SQLCATALOGS option lets you specify whether SQL catalog tables are to be backed up.

```
SQLCATALOGS [ ON | OFF ]
```

ON

lets you back up a table that is part of a catalog. ON is the default if you specify the SQLCATALOGS option in the BACKUP command.

OFF

specifies that SQL catalogs are not backed up. OFF is the same as omitting the SQLCATALOGS option from the BACKUP command.

Guidelines

- Use SQLCATALOGS ON to dump all SQL catalogs when reporting a problem to HP support. Otherwise, do not use this option to back up SQL objects.
- During a RESTORE procedure, a catalog table can be restored only as an ordinary (noncatalog) table with a file code of 0.
- Changes made in a SQL catalog after it is backed up are not preserved if the catalog on the BACKUP tape is restored to the same catalog name.
- This option cannot be used in the same BACKUP command as the NOSQLDATA option.
- For more information, see Backing Up SQL Files on page 3-48 and SQLCATALOGS on page 7-36.

START

The START option specifies where in *backup-files* the BACKUP procedure is to begin. Files on each subvolume, subvolumes on each volume, and volumes on a node are sorted in alphabetical order.

```
START [$volume.][subvolume.]file-id
```

volume.subvolume.file-id

directs BACKUP to begin with a specific volume, subvolume, and file, rather than beginning with the first file referenced by *backup-files* in the BACKUP command.

Guidelines

- The file specified with the START option must be referenced in *backup-files*.
- Specify the file set like any other file set in a BACKUP command. You can use only the asterisk (*) wild card to represent a whole subvolume name or file ID:

  *$volume.subvolume.**

  or

  $volume..**

  but not

  *$volume.subvolume.*AA*

- The START option works the same as the START qualifier in a qualified file set.

Example

If $DATA.AAA.AA through $DATA.CCC.PZZZZZZ are already backed up, and you do not want to back them up again, to start another BACKUP procedure where the previous one ended:

```
1> BACKUP $TAPE2, $DATA.*.*, START CCC.QA, LISTALL
```

TAPEMODE

The TAPEMODE option specifies the tape recording mode for tape units that support streaming. Streaming mode is a faster mode of transmission that locks the drive into a constant carrier signal. You can back up large files considerably faster if you specify the STREAM mode rather than the STARTSTOP mode.

```
TAPEMODE [ STARTSTOP | STREAM ]
```

STARTSTOP

indicates tape speed is 50 ips (inches per second) and tape density is 12000 bpi (bits per inch). This mode writes one record to the tape at a time.

STREAM

indicates tape speed is 75 ips and tape density is 12000 bpi. This mode writes one block to tape at a time.

Guidelines

- You cannot specify the TAPEMODE and DENSITY options in the same BACKUP command.
- The default recording mode is the physical setting of the tape drive if you do not specify the TAPEMODE option.

VERIFYREEL

The VERIFYREEL option, for use in file mode only, directs BACKUP to examine each reel for data integrity after the tape is written.

```
VERIFYREEL
```

Guidelines

- Using the VERIFYREEL option rewinds the tape when it reaches the end and then reads and examines the tape. Verification involves checking tape-record sequence numbers and reading the volume labels, file labels, data records, and checksums.
- The VERIFYREEL option does not examine a file until it rewinds the tape. Use the VERIFYTAPE option (for format 2 tapes) to examine a file immediately after it is written. Format 3 tapes cannot be verified after each file is written.

VERIFYTAPE

The VERIFYTAPE option, for use in file or volume mode, directs BACKUP to examine each tape reel for data integrity. BACKUP examines the tape after each file is written (or after the entire tape is written) depending on the tape format in use.

```
VERIFYTAPE
```

Guidelines

- While creating a tape in format 2, BACKUP with the VERIFYTAPE option examines each file after it is written to tape.
- After creating a tape set in format 3 (archive format), using the VERIFYTAPE option rewinds the tape when it reaches the end, then reads and examines the tape.
- In both cases, verification involves checking tape-record sequence numbers, and reading the volume labels, file labels, data records, and checksums.

VOL

The VOL option gives a new volume or subvolume name to the files copied to tape. This option applies only to Enscribe files. For SQL files, use the MAP NAMES option in RESTORE to restore your files with new volume and subvolume names.

```
VOL [ $new-vol.]new-subvol
```

$new-vol

is the new volume name.

new-subvol

is the new subvolume name.

Guidelines

- If you specify a new volume name with the VOL option, all the files you are backing up must reside on the same volume.
- If you specify a new subvolume name with the VOL option, all the files you are backing up must reside on the same subvolume.
- The new volume or subvolume you specify with the VOL option must be on the node you are backing up.
- You cannot use wild-card characters with this option.

VOLUMEMODE

The VOLUMEMODE option directs BACKUP to operate in volume mode, rather than file mode.

```
VOLUMEMODE , { $volume | $ldev }  [ -P | -M ]
```

Note. Any volume-mode BACKUP/RESTORE request to an SMF virtual disk results in the error message: “Volume Mode BACKUP/RESTORE is incompatible with SMF virtual disks, use File Mode.”

$volume

is the volume name of the disk volume to back up, such as $MIS.

$ldev

is the logical device number of the disk volume to back up, such as $24.

-P
-M

specifies the primary or mirror half of a volume pair to be backed up. If neither one is specified, the primary and mirror halves must be valid and matching, and the logical pair is backed up.

Guidelines

- Use -P or -M when you only know the primary or mirror half of the volume pair is valid.
- The disk volume you specify to be backed up must be on the current node (system); that is, you cannot do a remote volume-mode backup.

- If you attempt a volume-mode backup from a disk with a capacity greater than 2 GB, you will be unable to restore it to an operating system version prior to D30, and this message is generated:

  ```
  The source disk has a capacity of more than 2 GB, which
  cannot be restored to a pre-D30 system.
  ```

- For more information, see Backing Up in Volume Mode on page 3-54 and Example on page 3-55.

WHOLEDISC

Use the WHOLEDISC option to specify backup of an entire DP2 disk image, including an intact free space table (FST). This option applies only to volume-mode backups for DP2 disks.

```
WHOLEDISC
```

Guidelines

- The WHOLEDISC option instructs the BACKUP utility to copy the disk image and the FST exactly as they exist on the disk (compression is not done). Without specifying the WHOLEDISC option, the BACKUP utility compresses the FST to optimize disk space when the tape is restored.
- If file-system error 58 (disk free space table is marked bad) appears when you try to access a file, the FST is probably corrupt. If the FST is corrupt, use the WHOLEDISC option to back up the entire disk with the FST intact. Restore the tape to a nonproduction system and use the SCF CONTROL DISK, REBUILDDFS command to repair the corrupt FST.

Completion Information

The BACKUP utility returns a completion code to the process that started it. The completion code indicates if the BACKUP procedure ran successfully. Its primary purpose is to support batch processing by allowing conditional execution of subsequent processes based on the success of the BACKUP procedure.

If TACL started the BACKUP procedure, the completion code is stored in a TACL variable (:_COMPLETION), where you can examine it using TACL functions. TACL also displays the completion codes at the home terminal unless the BACKUP procedure completes normally without any warnings or errors. For more information about completion codes and batch processing, see the *TACL Programming Guide*.

In addition to a completion code, BACKUP also returns a subsystem identifier (TANDEM.74.*version*), where *version* is the release version (for example, D46). The subsystem identifier is also stored in the TACL variable (:_COMPLETION).

The completion codes returned by BACKUP and the operating system are described in Table 3-3 on page 3-44.

Table 3-3. BACKUP Completion Codes

Code	Description
0	A normal voluntary termination. The process completed normally with no errors or warnings.
1	A normal, voluntary termination with warnings. All the files were copied, but warnings occurred on one or more files. (Warnings that are not file specific do not result in this code.)
2	An abnormal, voluntary termination with errors. BACKUP could not copy one or more files because of errors on the tape or disk. The tape is left in a consistent state minus the files in error. For example, if BACKUP encounters a corrupt file, it skips the file and continues with the next file.
3	A premature, voluntary termination with fatal errors. BACKUP terminated because of a fatal error or an invalid user command. The tape generated might be incomplete.
4	The BACKUP process never got started. This completion code is generated by the TACL process executing BACKUP. TACL returns the process creation error in the *terminationinfo* field.
5	The BACKUP process calls ABEND on itself. This code is the default completion code for the ABEND procedure.

BACKUP Guidelines and Examples

Basic Backup Procedures

When you back up files, you use a qualified file set (QFS) to specify the set of files to back up. All files in the QFS must originate from the same node; you cannot specify two nodes in the BACKUP command. BACKUP records the names of the files in the QFS in local internal format, without a node identifier (*volume.subvol.file-id*).

Although the files in the QFS all originate from one node, they might have implicit relationships with files on other nodes. For example, a file might be partitioned over several nodes, or a base table might reside on one node while its indexes reside on another. Unless you specify otherwise, BACKUP backs up these related files along with the files in the QFS. BACKUP stores the names of files from other nodes in remote internal format (*node-number.volume.subvol.file-id*) to prevent accidental overwriting of files that have the same names.

Note. The terms local and remote have a different meaning for QFS. Local implies the node specified in the QFS. Remote implies other nodes on the same network as the QFS.

The change in definition of “local” eliminates ambiguous file names that were caused by implicitly named files; for example, an SQL table that is remote in relation to where BACKUP is run and that has an index that is local in relation to the BACKUP utility. For example, if you perform this BACKUP command from one system (\A):

```
BACKUP $Tape, \B.$Vol.Subvol.Table1
```

where the file (Table1) is an SQL table on a remote node (\B) that has an index (TInd1) local to system \A. The index (TInd1) is being backed up implicitly with the Table1 file.

Prior to the D30.00 RVU, the NonStop files were backed up using:

```
$Vol.Subvol.Table1
\A.$Vol.Subvol.TInd1
```

After the D30.00 RVU, files are backed up using:

```
\B.$Vol.Subvol.Table1
$Vol.Subvol.TInd1
```

This removes the need for a mapping clause to restore files to their original locations.

Examples: Basic Backup Commands

- To copy all files on the volume $MYVOL, and subvolume MYSUBVOL to the tape mounted on the device $TAPE, and to list (on the current output device) the files backed up and files that caused errors:

```
1> BACKUP $TAPE, $MYVOL.MYSUBVOL.*, LISTALL
```

- To back up the file $DATA.RECRDS.MYFILE and all files in subvolume DEBIT on the current default volume ($DATA):

```
1> VOLUME $DATA
2> BACKUP $TAPE, (RECRDS.MYFILE, DEBIT.*)
```

- To back up all files on the volume $DATA, subvolume ACCTS, and list the associated file names on the output device $LP (a line printer):

```
1> VOLUME $DATA
2> BACKUP /OUT $LP/ $TAPE, ACCTS.*, LISTALL
```

- To back up all files on the volume $LIB1, including any primary or secondary partitions on $LIB1 but not partitions on other volumes:

```
1> BACKUP $TAPE, $LIB1.*.*, PARTONLY ON, LISTALL
```

- To use BACKUP's wild-card capabilities:

```
1> VOLUME $SYSTEM
2> BACKUP $TAPE, (SYS*.*UP, SYS*.?MON, SYS*.COBOL?),&
2> &LISTALL
```

  - SYS* matches all subvolumes on $SYSTEM whose names begin with SYS (such as the system subvolumes $SYSTEM.SYSTEM and $SYSTEM.SYS*nn*).
  - *UP matches all of the files that have names ending with UP (such as BACKUP and FUP).
  - ?MON matches all of the files that have four-letter names ending with MON (such as IMON and DMON, but not NETMON).
  - COBOL? matches all files that have six-letter names that begin with COBOL (such as COBOL1, COBOL2, but not COBOL or COBOLLIB).

Backing Up Format 1 and Format 2 Files

The two formats for disk files are:

- A format 1 file is any file created on a system running a NonStop operating system RVU preceding G06.00 or D46.00, or a file created on any newer NonStop operating system RVU that is smaller than 2 GB minus 1 MB.
- A format 2 file is either a large format file, or a file that can contain larger partitions than a file created on RVUs preceding G06.00 or D46.00. A format 2 file can exceed the 2 GB minus 1 MB size limit of a format 1 file.

For more information on handling format 1 and format 2 files, see the *File Utility Program (FUP) Reference Manual*.

▲ **WARNING.** You cannot restore format 2 files to a system that does not support large format files.

If your VOLUME MODE backup tape contains both format 1 and format 2 files, do not perform a RESTORE to a system that does not support format 2 files.

If your FILE MODE backup tape contains both format 1 and format 2 files, you will need the fallback SPR to restore files to a system that does not support format 2 files. The fallback SPR will restore the format 1 files, skip the format 2 files, and issue an error message.

Do not use an older version of RESTORE to process a tape containing format 2 files. When a format 2 file is encountered, the results are unpredictable. BACKUP might abort or create an unusable file. If you encounter this problem, purge the unusable file.

Using Qualified File-Set Lists

These examples display a variety of qualifiers and file attributes:

- To back up all of the files on $SYSTEM (except those on $SYSTEM.SYSTEM):

```
1> BACKUP $TAPE, $SYSTEM.*.* EXCLUDE $SYSTEM.SYSTEM.*
```

- To back up all audited files on $SQL that are owned by SOFTWARE.JNP and are SQL files or EDIT files (file code 101), with a resulting tape file blocksize of 28:

```
1> BACKUP $TAPE, ($SQL.*.* WHERE OWNER=SOFTWARE.JNP  &
1> &AND (SQL OR FILECODE = 101) ), LISTALL, AUDITED, &
1> &BLOCKSIZE 28
```

- To back up all files except object files (file code 100) from all subvolumes whose names match the form T9???D46 on volume $SPECIAL:

```
1> BACKUP $TAPE, $SPECIAL.T9???D46.*  &
1> &WHERE FILECODE <> 100, LISTALL
```

- To back up all files in the system that are licensed and not owned by the super ID (255,255) except those in the $SYSTEM.SYSTEM subvolume or any subvolume of the form $SYSTEM.SYS??:

```
1> BACKUP $TAPE, *.*.*                                &
1> &EXCLUDE ($SYSTEM.SYSTEM.*, $SYSTEM.SYS??.*)  &
1> &WHERE LICENSED AND NOT (OWNER = 255,255), LISTALL
```

- To back up only those files on volumes $BASE and $LADY that are owned by a user in the group SOFTWARE and were modified since May 4, 2000 at 1:30 p.m.:

```
1> BACKUP $TAPE, ($BASE.*.*, $LADY.*.*) &
1> &WHERE OWNER = SOFTWARE.*     &
1> &AND MODTIME AFTER 04 MAY 2000 13:30
```

- To back up all files except those that are audited:

```
1> BACKUP $TAPE, *.*.* WHERE NOT AUDITED, LISTALL
```

- To back up all files for which the FLTrustFlags is set to ME:

```
1> BACKUP $TAPE, *.*.* WHERE TRUSTME, LISTALL
```

Note. The FLTrustFlags attribute is supported only on systems running H-series RVUs or J-series RVUs.

Backing Up SQL Files

The types of SQL files that can be explicitly named in *backup-files* or backed up using the FROM CATALOG qualifier include: base tables, catalog tables, indexes on base tables and catalog tables, individual partitions of base tables and indexes, SQL shorthand views, and SQL object program files.

For SQL backup strategies and options, including guidelines for performing daily and periodic backups, and using the FROM CATALOG, PARTONLY, and INDEXES options, see the *SQL/MP Installation and Management Guide*. For information on backing up SQL/MX objects, see the *SQL/MX Installation and Management Guide*.

△ **Caution.** BACKUP is normally used as a secondary recovery method for SQL objects; TMF recovery operations are used as the primary method. An SQL object recovered with RESTORE might be inconsistent with the current catalog description of the object.

- When a base table is backed up with PARTONLY OFF and INDEXES IMPLICIT:
  - Any SQL protection views defined on the table are backed up automatically. You should not explicitly name protection views in the backup files.
  - Constraints and comments associated with the table are backed up automatically.
  - An SQL shorthand view is not backed up unless it is explicitly named in *backup-files.*
- You must use BACKUP to archive nonaudited SQL tables.

Note. SQL tables can be backed up only if the program file from BACKUP is licensed. Licensing occurs automatically if you use the DSM/SCM to install BACKUP. If you do not use DSM/SCM, you can use the FUP LICENSE command to license the program file.

- By default, BACKUP and RESTORE do not back up and restore SQL catalogs. Use the SQLCATALOGS ON option to back up and restore catalogs. (An SQL catalog is a set of tables containing the descriptions of SQL objects.)

  Always use BACKUP to back up SQL objects, but only use the SQLCATALOGS ON option to dump all SQL catalogs when reporting a problem to HP support.
- BACKUP and RESTORE do not work with remote SQL tables if SQL is not installed on the local node.

- To provide a create-like function for multiple partition SQL objects, use the BACKUP NOSQLDATA option followed by RESTORE MAP NAMES. The resulting SQL object is an empty copy of the original.
- In case of disk failure, use the VOLUMEMODE option of BACKUP and RESTORE to replace any volumes containing SQL catalogs, if possible. Use this option to restore SQL files only if the SQL objects have not changed since they were backed up.

△ **Caution.** Under normal conditions, to ensure that BACKUP and RESTORE handle all related objects together and avoid inconsistencies, use BACKUP with the default options PARTONLY OFF and INDEXES IMPLICIT.

When necessary, use the PARTONLY and INDEXES EXPLICIT options to back up or restore individual SQL components of a set of related objects (files), such as the individual partitions of a partitioned table or indexes of SQL tables. Use these options carefully:

- The consistency checking that BACKUP and RESTORE perform to validate the data during these procedures does not ensure the data consistency of SQL objects.
- If you use options from BACKUP and RESTORE incorrectly, you can cause the primary data to be inconsistent with the alternate indexes. For example, a base table can be inconsistent with its indexes and left invalid after a RESTORE process when PARTONLY ON or INDEXES EXPLICIT is specified. Use these options with extreme care.

Examples

These examples display common BACKUP procedures for SQL files. Each example is followed by the command syntax for the procedure:

- To back up only SQL shorthand views on volume $A, subvolume A:

```
1> BACKUP $TAPE, $A.A.* WHERE VIEW, AUDITED, LISTALL
```

- To back up only SQL programs from volume $A and catalog $C.C:

```
2> BACKUP $TAPE, $A.*.* WHERE SQLPROGRAM
2> &FROM CATALOG $C.C, AUDITED, LISTALL
```

- To explicitly back up index table $B.B.INDX:

```
3> BACKUP $TAPE, $B.B.INDX, AUDITED, LISTALL, &
3> &INDEXES EXPLICIT
```

- To explicitly back up the secondary partition $D.D.PART:

```
4> BACKUP $TAPE, $D.D.PART, AUDITED, LISTALL &
4> &PARTONLY ON
```

Backing Up or Converting DP1 and DP2 Files

You can use BACKUP to back up DP1 files, DP2 files, or a mix of DP1 and DP2 files. You can also use BACKUP to convert files from DP1 to DP2 (or the opposite).

The B40 RVU of the operating system was the last to support DP1 files on disk; the DP1 disk process does not exist in any C-series, D-series, or G-series RVUs. C-series versions of BACKUP and RESTORE support DP1 files on tape and can be used to convert files between DP1 and DP2 formats. This is useful only for transferring files between a C-series system and an A-series or a B-series system or for restoring an old BACKUP tape containing DP1 files made from an A-series or B-series RVU.

- When converting files from one disk-process type to another, BACKUP with the PARTONLY ON option skips all relative files and any DP1 entry-sequenced files that do not have DP2 block sizes. For example, blocks of 1536, 2560, 3072, or 3584 bytes are not compatible with DP2. An error message gives the names of the skipped files. You must back up these files without the PARTONLY option.
- If you specify the PARTONLY ON option, BACKUP backs up entire partitioned files that have some secondary partitions on remote nodes if you include the DP2FORMAT option. BACKUP creates the tape in format 2 and converts any DP1 files it encounters to DP2 format.
- When you include the DP1FORMAT or DP2FORMAT option in the BACKUP command (and the *backup-files* contain files of multiple types):

  1. BACKUP creates temporary files on the default volume while performing the conversions. If the file being converted is a partitioned file, BACKUP creates the temporary files on the volume where the source (primary) file resides.
  2. BACKUP converts the files listed in the *backup-files* on a file-to-file basis. That is, as it converts each file, BACKUP copies the converted (temporary) disk file onto the specified tape.

- If file conversion occurs during a BACKUP procedure, the resulting BACKUP tape always contains files of all one disk-process type. A BACKUP tape produced without file-conversion options can contain both DP1 and DP2 files. The BACKUP procedure produces such a tape if you back up files from disk volumes controlled by each of the two disk processes.
- Both BACKUP and RESTORE try to adjust extent sizes of individual partitions when converting partitioned files from one disk-process type to another. If all the records of a relative or entry-sequenced file can remain in the same partition, then the file is converted; otherwise, it is skipped.
- To convert a DP1 disk file in which the records of a relative or entry-sequenced file are split into different partitions:

  1. Back up the entire partitioned file using the DP2FORMAT option and omitting the PARTONLY option.
  2. Restore the file to a DP1FORMAT disk.
  3. Back up the file again by using DP2FORMAT and omitting PARTONLY.

- If BACKUP returns file-system error 45 (file is full) for a key-sequenced file, back up that file in a separate BACKUP procedure and specify larger extent sizes for the

file with the EXT option. Error 45 can appear for a key-sequenced file when the specified slack values significantly increase the size of the converted file.

Backing Up With Labeled Tapes

You can perform labeled-tape backup and restore procedures using either file mode or volume mode. BACKUP, RESTORE, and BACKCOPY send a labeled-tape request through a CLASS TAPE DEFINE to the labeled-tape server process ($ZSVR). The $ZSVR labeled-tape server process passes messages about labeled-tape operations to the operator console ($0) and the MEDIACOM console (if one exists).

For more information, see Backing Up or Converting DP1 and DP2 Files on page 3-49. For a description of MEDIACOM, see the *DSM/Tape Catalog Operator Interface (MEDIACOM) Manual.*

Three types of labeled tapes can be used with BACKUP, BACKCOPY, and RESTORE: ANSI, IBM, and BACKUP.

- BACKUP can write to scratch ANSI, scratch IBM, and BACKUP labeled tapes.
- RESTORE can accept only IBM and BACKUP labeled tapes.

  Because RESTORE can only read data, it is unable to use scratch tapes. The volume label on an IBM labeled tape conforms to IBM standards. The BACKUP data is accessible only on the NonStop operating system. After the data on the tape expires, the tape can be reused on a NonStop system or IBM system.

Do not use labeled backup tapes as a medium of exchange between NonStop systems and other types of systems. NonStop operating system backup tapes contain physical image copies of Enscribe files and unstructured files in a proprietary archive tape format. If you need to exchange data between a NonStop system and another type of operating system, use blocked records on standard labeled tapes (for example, ANSI or IBM labeled tapes) that can be read and written by FUP or other programs.

Only BACKUP and RESTORE can access NonStop operating system labeled backup tapes.

Labeled Backup Tape Volume Labels

For labeled backup tapes, BACKUP writes either an IBM volume label or a customized ANSI volume label (the BACKUP label) at the beginning of the tape.

- The initialized IBM tape, as defined by the IBM-MVS standard, contains:
  - A beginning-of-volume label (VOL1)
  - A beginning-of-file-section label (HDR1)
  - A tape mark
- The initialized ANSI tape, as defined in the ANSI standard, contains:
  - A beginning-of-volume label (VOL1)

- A beginning-of-file-section label group (HDR1, HDR2)
- Two tape marks
- An end-of-file label group (EOF1,EOF2)
- Two tape marks

Note. The BACKUP label type is a customized version of the ANSI label. The labels are identical except character 80 (from the BACKUP label) is a B instead of a 3. When BACKUP writes to a scratch ANSI labeled tape, it changes it into the BACKUP label type.

Following this volume label, BACKUP writes all labeled backup tapes in tape format 3. Therefore, the files on a labeled backup tape are not in the standard labeled tape format (ANSI or IBM).

Using DEFINEs for Labeled Backup Tapes

You must specify a CLASS TAPE DEFINE (instead of a tape device name) in the BACKUP command to use a labeled backup tape. You create the DEFINE in the same way you create DEFINEs for ANSI and IBM labeled tapes. If you are not familiar with labeled-tape processing on NonStop systems (or with DEFINEs), see the *Guardian User's Guide*. It gives instructions on using different DEFINEs including CLASS TAPE DEFINEs for labeled-tape processing.

The two differences between using TAPE DEFINEs for backup labeled tapes and for ANSI and IBM tapes are:

- You can use only a subset of the CLASS TAPE attributes for a backup labeled tape. The supported attributes are summarized in Table 3-4. For a complete description of these DEFINE attributes, see Appendix E, CLASS TAPE DEFINEs.
- You must specify the LABELS attribute as either BACKUP or IBMBACKUP.

Table 3-4. Permissible DEFINE Attributes for Labeled-Tape Backup (page 1 of 2)

Attribute	Use	Description
CLASS	Required	Must be specified as TAPE.
DEVICE	Optional	Specifies the tape drive to use.
DENSITY	Optional	Specifies tape density. If the DENSITY option is specified in the BACKUP command, it must match this value, and only an open reel tape can be used.
EXPIRATION	Optional	Specifies the month, day, and year after which this tape can be overwritten.
FILEID	Optional/ Required	Specifies the name of the tape (file) set. Optional for LABELS BACKUP; required for LABELS IBMBACKUP.
GEN	Optional	Indicates that this file is part of a generation group.
LABELS	Required	Specifies the tape label type (BACKUP or IBMBACKUP).
MOUNTMSG	Optional	Specifies a message to be displayed to the operator.

Table 3-4. Permissible DEFINE Attributes for Labeled-Tape Backup (page 2 of 2)

Attribute	Use	Description
OWNER	Optional	As many as 14 characters identifying the tape owner.
RETENTION	Optional	Specifies the number of days to retain the tape file. If you do not specify RETENTION, BACKUP sets the retention to one day. That is, the tape expires the day after it is created.
SYSTEM	Optional	Specifies the node where all tapes must be mounted.
TAPEMODE	Optional	Specifies the operating mode of a cartridge tape drive as STARTSTOP or STREAM. If specified, only a cartridge tape can be used.
VERSION	Optional	Specifies a version within one generation.
VOLUME	Optional	Specifies the volume ID of the tape or specifies the tape as SCRATCH.

Using Scratch Tapes

You can use expired BACKUP, IBM, or ANSI labeled tapes for labeled tape BACKUP procedures. If no expired tapes exist, you can create an ANSI or IBM labeled scratch tape using MEDIACOM.

Note. Scratch ANSI or BACKUP tapes can be used with LABELS BACKUP. Only scratch IBM tapes can be used for LABELS IBMBACKUP.

If you want to use specific scratch tapes, specify the volume IDs in the VOLUME attribute of the DEFINE. If any scratch tape will do, omit the VOLUME attribute from the DEFINE (or specify SCRATCH) as its value.

Note. Do not confuse the DEFINE attribute VOLUME SCRATCH with the BACKUP option SCRATCHVOL $*volume*. The VOLUME SCRATCH attribute refers to a tape volume; SCRATCHVOL refers to a disk volume. Use the DEFINE attribute for labeled tape BACKUP procedures. Use the BACKUP SCRATCHVOL option only during file conversion operations with nonpartitioned files or individual partitions of partitioned files.

Examples

These examples demonstrate how to create CLASS TAPE DEFINEs and use them in BACKUP commands. For more information on labeled-tape processing, see the *Guardian User's Guide*. For more information on creating DEFINEs, see Appendix E, CLASS TAPE DEFINEs. For more information on CLASS TAPE CATALOG, see the *DSM/Tape Catalog User's Guide*.

- To use the DEFINE =MYBACKUP to specify using two scratch tape volumes, NY082 and NY004, as backup tapes and to give the expiration date of December 31, 2001:

```
1> ADD DEFINE =MYBACKUP,CLASS TAPE,LABELS BACKUP,&
1> &VOLUME (NY082, NY004),EXPIRATION DEC312001
2> BACKUP =MYBACKUP,*,LISTALL
```

  $ZSVR (the labeled tape server process) sends a mount request to the EMS log and the MEDIACOM console for operator attention. After the operator mounts volume NY082, the BACKUP process begins. A mount request for the NY004 tape volume is issued if the BACKUP process fills NY082 but has not completed.

- To use the DEFINE =MYBACKUP2 to specify that any scratch tape is acceptable as the BACKUP tape, and to give the expiration date of December 31, 2001:

```
1> ADD DEFINE =MYBACKUP2,CLASS TAPE,LABELS BACKUP,&
1> &VOLUME SCRATCH,EXPIRATION DEC312001
2> BACKUP =MYBACKUP2,*,LISTALL
```

  $ZSVR (the labeled tape server process) sends a mount request to the MEDIACOM console for operator attention. After the operator mounts a scratch tape, the BACKUP process begins. This request can use any scratch ANSI or BACKUP tape mounted on any tape drive on the system.

Backing Up in Volume Mode

In volume mode, BACKUP transfers an entire disk volume to tape. Only the super ID (255,255) can use the volume-mode backup procedure. When performing a volume-mode backup:

- The volume being backed up must be in the DOWN disk state. If only one mirror of a mirrored pair is being backed up, only that mirror needs to be in the DOWN state; the other mirror can be UP.
- If you are backing up only one mirror of a mirrored pair, enter the appropriate pair of commands in the order shown:

SCF command	Action
INFO DISK, BAD	Ensures that neither mirror contains any defective sectors
STOP DISK	Puts that mirror in the DOWN state to ensure that data on the disk is consistent for all open files on that volume

- Do not back up the system disk ($SYSTEM) in volume mode without a compelling reason. If you do this, the disk must be restored under a different name. Also, do not use volume mode to restore $SYSTEM because the procedure requires both mirrors to be DOWN. This state is not reasonable for $SYSTEM.
- Volume-mode BACKUP/RESTORE requests are incompatible with DSM/SM virtual disks.

- A volume-mode backup of a disk larger than 2 GB cannot be restored to an operating system version prior to D30.
- DP1 and DP2 file conversion is not possible during volume-mode backup procedures.

Example

For more information on the commands in this example, see the *SCF Reference Manual for the Storage Subsystem*.

In this example, $DATA is the DP2 mirrored volume to be backed up.

1. Close or make inactive any partitioned files.
2. Check the current state of the disk to be backed up.

   ```
   1> SCF
   STATUS DISK $DATA
   ```

   If the output from the STATUS DISK command shows that the disk to be backed up is not in a normal UP state, perform whatever steps are necessary to return this disk to a normal UP state.
3. Determine if there are any unspared defective sectors on either half of the mirrored volume to be backed up:

   ```
   INFO DISK $DATA, BAD
   ```

   If there are no defects, the SCF display will indicate that state.

   If the SCF INFO DISK, BAD command indicates any unspared defective sectors:

   a. Use the SCF CONTROL DISK, SPARE command to correct the situation.
   b. Recheck the status of both halves of the mirrored volume using the SCF INFO DISK, BAD command until there are no new defects.
4. Ensure that no disk I/O activity is occurring against the volume being backed up.
5. Issue an SCF REFRESH command for the mirrored half to be backed up:

   ```
   CONTROL DISK $DATA, REFRESH
   ```
6. Issue SCF STOP DISK commands for both paths to the mirrored half being copied:

   ```
   STOP DISK $DATA-M

   STOP DISK $DATA-MB
   ```
7. Verify that all paths to the removed half of the mirrored volume are DOWN, and then exit SCF:

   ```
   INFO DISK $DATA, DETAIL
   .
   .
   ```

```
.
EXIT
```

In this example, $DATA-M was just removed, so SCF should show the paths $DATA-M and $DATA-MB to be DOWN.

8. Enter the volume-mode BACKUP command. In this example, the mirrored half of $DATA is being backed up to a tape mounted on $TAPE4:

```
1> BACKUP $TAPE4, VOLUMEMODE, $DATA-M, BLOCKSIZE 28,&
1> &DENSITY 6250
```

A display similar to this appears on the console:

```
Volume-Mode BACKUP Program - T9074D46 - (15DEC1997)
Tape: $TAPE4  Operating System: D46   Tape Version: 101
Backup options: NO AUDITED, BLOCKSIZE 28, NO IGNORE,
                NO OPEN, NO PARTONLY

Volume name:  $DATA
Subtype: 3  ( 4104 -- 240MB )
Disk process type:  DP2
Volume label time:  23Dec2001 10:57

$DATA has been copied to tape.
$DATA has been left in the DOWN state

This volume mode backup tape can be restored to a disk
with a formatted capacity of 75110400 bytes or
greater
```

9. After the volume-mode BACKUP procedure finishes, check the state of the disk. Both paths to the mirror should be in the DOWN state.

```
1> SCF
INFO DISK $DATA, DETAIL
```

If a backup error occurred, some paths could have been left in the special (S*) state. If this happens, use the SCF STOP DISK command to bring down the paths before proceeding.

10. After the backup finishes, bring up the mirrored volume so that both halves of the mirrored volume match:

```
START DISK $DATA
EXIT
```

4 Disk Compression (DCOM)

The Disk Compression (DCOM) utility moves disk file extents to yield more usable space on a disk. You can use DCOM to:

- Analyze the current space allocation on a disk
- Relocate file extents on a disk
- Reduce the number of free-space extents, which reduces the effort needed to allocate or deallocate space
- Combine free space into larger extents so that files can be allocated with larger extents, which decreases the incidence of file-system error 43, (unable to obtain disk space for file extent). Although this decreases the incidence of file-system error 43 it does not guarantee elimination of this error.

Note. DCOM does not compress audit-trail files because the TMF must maintain exclusive access to these files for update purposes. However, DCOM does compress audited files.

DCOM supports 512-byte disks and 514-byte disks.

Security

DCOM is privileged, and each customer must determine which users are allowed to run it. To control access to DCOM:

- Do not license DCOM's program file for general use. Without licensing, only the super ID (255,255) can run DCOM.
- License DCOM's program file and limit access to a subset of users by either setting file security attributes or using a Safeguard access-control list.

When determining security, be aware that DCOM improves system performance after it finishes. However, while DCOM is working, it can have a slight negative impact on system performance and should be restricted to system operators.

To set the appropriate level of access to DCOM and other system utilities, refer to the security policy established by your organization. For more information about securing information on the NonStop operating system, see the *Security Management Guide*.

DCOM Syntax

DCOM is noninteractive, and it can be run online while the disk is operating. It displays reports and messages at the home terminal (by default), or sends the output to the file specified in the OUT run-option.

Note. To ensure compatibility with low-level system interfaces, the version of DCOM must match the version of the NonStop operating system. The D42 version of DCOM does not run on any earlier versions of the operating system. All older D-series versions of DCOM run only on versions of the operating system prior to D42.

The DCOM command syntax is:

```
[ run ] [ \node.]DCOM / run-options /
          [ $volume | HELP ] [, options ]

      [, IGNOREBADSECTORS   ]
      [, MAXMOVES n         ]
      [, VERIFY             ]
      [, WORKFILE volume   ]
```

node

is a name of the local or remote node (system) where DCOM runs. If you want to run DCOM on a remote node in a network, the utility must be secured on the remote node for network access. A remote password for *node* must exist for any user that is logged on to the local node where the DCOM RUN command is issued.

If *node* is not specified, DCOM runs on the local node where the command interpreter is running. (This is usually the node that is connected to your terminal.)

run-option

is any option for the TACL RUN command. The two most common run-options are:

```
IN filename
OUT listfile
```

The IN option specifies the input files, and the OUT option specifies the output files. These files usually override the home terminal as the input or output device. An IN file is a text file that contains the remainder of the command line, including parameters and options. For a complete list of *run-options,* see the description of the TACL RUN command in the *TACL Reference Manual*.

volume

specifies the disk volume where the compression is to be performed. The volume specified must reside on the *node* where DCOM will run. The *volume* can contain Enscribe, SQL, Kernel-Managed Swap Facility (KMSF) and SMF files. Optional keywords and parameters can follow the *volume* specification in any order.

If you omit *volume*, a brief description of DCOM and its options is displayed (identical to the display you receive after specifying HELP).

HELP

displays a brief description of DCOM and its options.

IGNOREBADSECTORS

specifies that DCOM is to perform the disk compression even if it detects unspared defective sectors on the disk during the initial analysis. Without this option, DCOM terminates when it detects unspared defective sectors and displays:

```
Please eliminate bad sectors using the PUP SPARE command.
```

MAXMOVES *n*

limits the total number of extents that DCOM can move during any one compression. By using this option, you can compress a heavily fragmented disk in several steps and limit processing time. This is helpful if you need to run DCOM when others are using the system. Allowable values are in the range 1 through 32,767. The default limit is 32,767.

Although this parameter provides an upper limit for the number of extent moves, it does not indicate the number of moves DCOM actually performs. If DCOM exhausts its possibilities for extent moves, it concludes before reaching the specified limit.

VERIFY

specifies that every write operation DCOM performs to the disk must be verified. This option causes DCOM to run much longer but with more integrity checking.

`WORKFILE` *volume*

overrides the default work file volume. This option is useful when the default work file volume is almost full and would cause DCOM to terminate in error. The default is the volume where the program swap file resides unless you modified it using the BINDER program as described in Changing the Default Work File Volume on page 4-10. The DCOM HELP display shows the default work file volume.

Unlike DSAP, DCOM does not recognize a permanent WORKFILE option because the compression process changes the work file to an unusable form.

Handling SMF Files

DCOM is used to reduce the number of free space fragments on a disk. Storage Management Foundation (SMF) provides system-managed storage capabilities for NonStop systems. DCOM interacts with SMF as follows:

- Because SMF virtual disks do not have free space and can also span several physical disks, DCOM will not run on a virtual disk. If you attempt DCOM $L, an error message is displayed, and DCOM terminates.

  Note. This behavior only affects virtual disks. Running DCOM $P is permitted.

- DCOM performs on a single physical volume basis. It will not move extents from one volume to another, even if the two physical volumes are part of the same SMF storage pool.
- Some SMF files must be accessible to SMF at all times. Because DCOM opens files exclusively during a move, it will not attempt to move these SMF file types. DCOM identifies these SMF files with 460, 461, and 462 file codes.

Exiting DCOM

Exit DCOM by pressing the Break key. DCOM continues to run in the background until it is finished. If your terminal is the output device, you can reenter DCOM by typing PAUSE at the TACL prompt.

Stopping DCOM

Stop DCOM by pressing the Break key and typing STOP at the TACL prompt. (Without parameters, STOP applies to the process that the current TACL most recently started.) This leaves the disk partially compressed and fully functional. Stopping DCOM has no adverse effects.

Completion Information

DCOM returns completion information to the process that started it. The completion code indicates how successful DCOM ran. Its primary purpose is to support batch processing by allowing conditional execution of subsequent processes based on the success of the DCOM process. If TACL started the DCOM utility, the completion information is stored in a TACL variable (:_COMPLETION), where you can examine it using TACL functions.

TACL also displays the completion information at the home terminal unless DCOM completes normally without any warnings or errors. For more information on completion information and batch processing, see the *TACL Programming Guide*.

The completion information returned by DCOM includes these fields:

Field	Description
completioncode	An integer that describes how DCOM terminated (see Table 4-1, DCOM Completion Codes, on page 4-6)
terminationinfo	An integer that describes the nature of errors or warnings (see Table 4-1, DCOM Completion Codes, on page 4-6)
subsystem	Contains the DCOM subsystem ID, TANDEM.32.*release*, where *release* is the release version of DCOM (D42, for example)
text	A character string describing the code in *terminationinfo* (see Table 4-1, DCOM Completion Codes, on page 4-6)
textlength	The number of characters in *text*

Table 4-1. DCOM Completion Codes

Code	Description
0	A normal, voluntary termination of DCOM with no errors or warnings.
1	A normal, voluntary termination of DCOM with warnings. Disk compression completed, but DCOM reported disk structure diagnostics, space allocation problems, or a nonfatal internal program error. DCOM includes this termination information with the completion code: **Terminationinfo Text** 1 Disk structure anomalies or I/O errors 2 Internal program error
2	An abnormal, voluntary termination with errors. DCOM performed partial compression, but a disk error or an internal error prevented full compression. DCOM includes this termination information with the completion code: **Terminationinfo Text** 1 Disk structure anomalies or I/O errors 2 Internal program error
3	An abnormal, voluntary termination with fatal errors. DCOM did not compress the disk because of severe disk errors, severe structural errors, a fatal program error, or an invalid user command. DCOM includes this termination information with the completion code: **Terminationinfo Text** 1 Disk structure anomalies or I/O errors 2 Internal program error 3 Command syntax or parameter error
4	DCOM never started, so the completion code is generated by the TACL process executing DCOM. The *terminationinfo* field holds an error code from the process creation procedure.
5	DCOM ended abnormally and returned 0 in the *terminationinfo* field.

Guidelines

Consider these suggestions and guidelines while running DCOM:

- DCOM compresses a disk even while that disk is in service. When DCOM is executing, there are three visible effects:
  - DCOM must have exclusive access to the file whose extents it is moving. Do not run DCOM if an important operation will fail when it takes exclusive possession of a file. DCOM can move open files for all releases above D45, otherwise, it does not move open files.
  - DCOM burdens disk I/O channels and can degrade the performance of important disk I/O applications. Running DCOM at a low priority reduces (but does not eliminate) performance degradation.
  - DCOM rearranges file extents. Any file that was specially allocated in contiguous extents to reduce disk seek time might end up in a noncontiguous state after it is compressed.
- DCOM is designed for online disks so it does not perform perfect compression. You can sometimes achieve additional compression by running DCOM several times (serially), but you need to consider the performance implications. The starting overhead of each run is significant because DCOM must read and sort the disk directory. DCOM cannot determine when additional runs are pointless.
- DCOM prevents more than one DCOM process from running at a time by exclusively opening the SYSDCOM.RECOVERY file to signify that it is running. This error message occurs if a second DCOM process is attempted:

  ```
  Unable to open SYSDCOM.RECOVERY file, error: 12
  ```

- DCOM allocates space for data segment and swap file for storing data about file labels and extents. When DCOM needs to allocate 1024 MB of workspace, it displays "`WARNING: 1024MB SPACE WILL BE REQUIRED`". If the required space is not available, DCOM abends and displays "`DIRECTORY WORKFILE CREATION/OPEN/MAPPING ERROR: 43. TRY AGAIN WITH 1024MB WORKSPACE`".
- If the system crashes while DCOM is running, temporary files could be left on the disk being compressed. The files are then owned by the super ID (255,255) and secured "---N". These files have the CLEARONPURGE option set if they have been used to move the extents of the files set for CLEARONPURGE. For more information on the CLEARONPURGE option, see the FUP SECURE command in the *File Utility Program (FUP) Reference Manual*.
- DCOM has no internal security restrictions except that it requires licensing. The system manager controls access to DCOM by using the standard disk file security scheme.

- The version of DCOM must match the version of the NonStop operating system to ensure compatibility with low-level system interfaces:

DCOM Version	Compatible Operating System Versions
D42	D42 or later
D45	D45 or later
D48	D48.01 or later
G06	G06.06 or later
G07	G06.23 or later
Pre-D42	Pre-D42

- DCOM needs exclusive access to the file so it does not move the extents of some system files such as Disk Directory, SMS (storage management system) files, corrupt files, swap files, and audit trail files.
- Using the MAXMOVES option, you can perform a fixed amount of compression each day, depending on the amount of time a system is inactive (when no performance-critical applications are running).
- DCOM supports SQL format2 files for SQL/MP.
- DCOM supports up to 4,500,000 files on a single disk volume.
- DCOM supports a 512 byte sector disk and a 514 byte internal disk.
- DCOM does not permit multiple volumes in the command line.

Examples of Compression Output

In its preliminary stage, DCOM produces the same summary report and space-distribution reports produced by DSAP. For examples of these reports, see Report Formats on page 5-17.

You can have Enscribe and SQL files on the volume you specify for the DCOM procedures. Space allocation for SQL file output is not distinguished from space allocation for Enscribe file output in DCOM. The reports from DCOM are unchanged for SQL file support with one exception: If DCOM reports a file with double allocated extents, and the message “SQL Shadow” appears after the file name, the file is an SQL table that was dropped, but the drop is not yet committed. This type of file is invisible, but its extents are still allocated until the drop is committed.

Before DCOM Moves File Extents

When the compression phase begins (but before DCOM moves any extents), it sends messages to the event message collector ($0).

Figure 4-1. Format of DCOM Operator Messages (Before the Move)

```
DCOM:  Starting disk compression for $DATA.
DCOM:   23199 pages of free space in 356 extents (20.3%).
DCOM:    1332 pages in largest free extent.
```

While DCOM Moves File Extents

DCOM sends a report to the current output device, denoting each file extent it moves and which files were directly affected. For example, in “Move Num” 5 displayed in Figure 4-2, DCOM moves extent #11 of the file XRAYENF.QCACHE from page 52181 on the disk to its new location at page 5110 on the disk at 12:30 and 22 seconds.

Figure 4-2. Format of DCOM Report (During the Move)

Move Num	To Page-Num	From Page-Num	Extent Size	Extent Num	File Name	Time
1	3424	52710	2	0	XRAYENF.ZQTXRATE	12:30:16
2	3565	52705	2	1	XRAYENF.ZQCPU	12:30:18
3	4794	52704	2	0	XRAYENF.ZQCPU	12:30:19
4	5089	52438	2	0	XRAYENF.TRANRATE	12:30:20
5	5110	52181	2	11	XRAYENF.QCACHE	12:30:22
6	5184	52180	2	0	XRAYENF.QCACHE	12:30:23
7	5189	52179	2	1	XRAYENF.QBALANC2	12:30:24
8	5208	52178	2	0	XRAYENF.QBALANC2	12:30:26
9	5210	51973	2	1	XRAYENF.OQP	12:30:27
10	5221	51972	2	2	XRAYENF.OQP	12:30:29

After DCOM Moves File Extents

When it finishes moving extents, DCOM produces a report.

Figure 4-3. Format of DCOM Report (After the Move)

```
227 extents moved, 1019 pages.
23199 pages of free space in 147 extents (20.3%).
1402 pages in largest free extent.
```

- The first line reports the total number of extents and pages moved.
- The second line gives the new number of free extents, which are fewer and larger than before the move, depending on the success of the compression.
- The third line lists the number of pages in the largest free extent on the disk. To see the remaining free extents, refer to the space distribution reports that DCOM also produces at this time.

When compression is complete, DCOM sends operator messages to the event collector.

Figure 4-4. Format of DCOM Operator Messages (Concluded)

```
DCOM:  Terminating disk compression for $DATA.
DCOM:      227 extents moved, 1019 pages.
DCOM:    23199 pages of free space in 147 extents (20.3%).
DCOM:     1402 pages in largest free extent.
```

The time required for a single compression run varies according to the disk size and its degree of fragmentation. Subsequent compressions take less time than the first.

Changing the Default Work File Volume

You cannot change the default workfile because DCOM/DSAP has become a native object since G06.16. In the object files of DSAP and DCOM, the default work file volume name is located at the beginning of the procedure SELECT^WORK^VOLUME.

In this pre-G06.16 example, the volume name for the DCOM work file is set to $TEMP. The characters for the volume name must be specified by twos within quotation marks. $TEMP is specified “$T”, “EM”, “P), “ ” (for a total of eight characters).

```
1> VOLUME $SYSTEM.SYSTEM
2> BIND
@FILE DCOM
@ADD *
@MODIFY CODE SELECT^WORK^VOLUME ASCII 0, "$T","EM","P ","  "
@BUILD DCOM!
@EXIT
3> FUP LICENSE DCOM
```

The FUP LICENSE command is optional, depending on what kind of file security you want. For information about the FUP LICENSE command, see the *File Utility Program (FUP) Reference Manual*. It is necessary to relicense the utility because the BINDER BUILD command removes the old license. For more information about BINDER, see the *Binder Manual*.

5

Disk Space Analysis Program (DSAP)

The Disk Space Analysis Program (DSAP) analyzes how disk space is used on a specified volume.

DSAP copies the disk directory and free-space table to the current work file. By specifying options, you can manipulate this data to produce several different reports about the use of the disk space for that volume.

The free-space table is limited only by your primary (main) and secondary (contiguous disk space) memory requirements.

Note. The version of DSAP must match the version of the NonStop operating system to ensure compatibility with low-level system interfaces. Therefore, the G08 version of DSAP runs on G06.24 and later RVUs, the G07 version of DSAP runs only on the G06.23 RVU, and the G06 version runs on G06.00 through G06.22 RVUs.

Note. DSAP supports 512-byte disks and 514-byte disks. For a 512 byte sector disk, DSAP reports the actual physical extent sizes including the check block information. The unused pages column of the DSAP report considers the check blocks. If an unstructured file with checksum blocks has 14 pages (28 K) allocated for an extent and only 1 page (2 K) is utilized, 2 pages (4 K) contain the checksums. Therefore, the unused pages are shown as 14 pages. The Dealloc pages column of the DSAP report considers the check blocks.

Security

DSAP is privileged. Each customer determines which users are allowed to run it. To control access to DSAP:

- Do not license the program file for general use. Without licensing, only the super ID (255,255) can run DSAP.
- License the program file and limit access to a subset of users by setting file security attributes or by using a Safeguard access-control list.

DSAP measures disk space and has no effect on the system, so restricting its use is often unnecessary. However, it might not be preferable in some environments to permit general access to some of the reports generated by DSAP.

To set the appropriate level of access to DSAP and all other system utilities, see the security policy standards established by your organization. For a complete discussion about securing information on a NonStop system, see the *Security Management Guide*.

DSAP Syntax

DSAP is a noninteractive, privileged program that runs online while the disk is operating. It can generate reports for more than one disk volume during a single run.

The default output device is the home terminal, but you can specify other types of output devices, such as a disk file or printer. When the output device is a terminal, the output width is 79 characters (for most options), and output is not formatted into pages. When the output is to the spooler, output width is 132 characters, and page length is 60. A header message prints at the top of each page in the following format:

```
PAGE n DSAP--$volume ON \system--report-name--time
```

The syntax for the DSAP utility appears is:

```
[ \node.]DSAP [ / run-option [, run-option ] ... / ]
  [ $volume-specification | HELP ] [ , options ] ...
```

node

is the name of a local or remote node (system) where the DSAP utility runs. If you want to run DSAP on a remote node in a network, you must secure it for remote access on the remote node. A remote password for *node* must exist for any user that is logged on to the local node where DSAP RUN command is issued.

If *node* is not specified, DSAP runs on the local node where TACL is running (usually the node where your terminal is connected).

run-option

is any option for the command interpreter RUN command. The two most common run-options are:

```
IN filename
OUT listfile
```

The IN option specifies input files, and the OUT option specifies output files. These files usually override the home terminal as the input or output device. An IN file is a text file that contains the remainder of the command line, including parameters and options. For a complete list of run options, see the description of the command interpreter RUN command in the *TACL Reference Manual*. DSAP and DCOM ignores the IN option.

volume-specification

identifies the disk volumes to be analyzed. Any volume specified must reside on the node where DSAP runs. DSAP creates a separate report for each volume specified. The reports are generated in alphabetical order by volume name.

The *volume-specification* has the format:

$*volume*

for reports on a single volume.

($*volume*, $*volume*,...)

for reports on a list of volumes.

$* | *

for reports on all volumes.

If *volume-specification* matches more than one volume name, DSAP displays a list of all the volumes it finds before producing the first report.

If you omit *volume-specification*, a brief description of DSAP and its options is displayed (identical to the display shown if you specify HELP.

Each volume name can contain wild-card characters (*) or alias characters (?). The asterisk (*) matches 0 to 7 characters, and the question mark (?) matches any single character. For example:

DSAP *	Reports on ALL volumes
DSAP ($DATA1,$SYSTEM)	Reports on the specified volumes
DSAP $D*	Reports on all volumes beginning with the letter D
DSAP $DATA?	Reports on all volumes beginning with $DATA and followed by a single character

HELP

displays a brief description of DSAP and its options.

options

can include *report-options*, *detail-selection-options*, EXTENTCHECK, and WORKFILE. For descriptions of all these options, see DSAP Options on page 5-6.

The *report-options* and *detail-selection-options* parameters can follow *volume-specification* in any order. Using commas to separate options is optional. If no options are specified with *volume-specification*, the DSAP utility only produces the summary report. The summary report is printed with each report—except the SHORT report.

report-options

```
ANALYSIS
BYSUBVOL
BYUSER
DETAIL
FILESIZE
FILESPACE
FREESPACE
NEWFORMAT (G-series only)
SHORT
SPACE
SUMMARY
TERAFORM (G-series only)
```

Each *report-option* specifies the type of report or set of reports created.

detail-selection-options

```
AGE { [ OVER ] number | UNDER number }
AUDITED
BROKEN
CRASHOPENED
DEALLOC number
ENSCRIBE
EXPIRED
LICENSED
OPENED { [ OVER ] number | UNDER number }
PARTITIONED
PROGID
ROLLFORWARD[NEEDED]
SEPARATE
SHOWNAMEMAP
SIZE { [ OVER ] number | UNDER number }
SQL
TEMPORARY
UNEXPIRED
UNUSED number
USER    [ group-name.user-name  ]
        [ group-number , user-number  ]


      [ -1 ]
      [ group-name.* ]
      [ group-number,* ]
```

Each *detail-selection-option* limits the scope of the DETAIL report. If you specify the DETAIL report option and one or more *detail-selection-options*, only the files that meet the requirements of at least one *detail-selection-option* are analyzed and included in the DETAIL report.

Reporting SMF Files

DSAP creates reports about the contents of disk volumes. SMF provides system-managed storage capabilities for NonStop systems (including location-independent naming). DSAP interacts with SMF as follows:

- The file names that DSAP displays are dependent on the type of volume specified (virtual or physical). If a virtual volume is specified (DSAP $L), the virtual file names are displayed, with some restrictions (report options that manage free space are ignored on virtual volumes).
- The behavior of DSAP on virtual volumes is the same behavior that results when you specify the USER option. All options are supported, with these exceptions:
  - The FREESPACE option is ignored on a virtual volume. Any other report that includes FREESPACE (SPACE and ANALYSIS) will produce all reports except FREESPACE.

- The SHORT option displays the name of the virtual volume and an error message indicating that this is a virtual volume.
- The SUMMARY report does not produce the Space Allocation Consistency Analysis or the Media Failure Analysis on a virtual volume. A message is printed indicating that this is a virtual volume instead of a message indicating the size of the disk.
- All percentages that compare space use with the size of the disk are displayed as 0.0%, which affects the SUMMARY, FREESPACE, and FILESPACE reports.
- Wild-card characters in DSAP expand to include both virtual and physical volumes. DSAP * displays all volumes on the system.

- If a physical volume is specified (DSAP $P), the physical file names are displayed. DSAP $P shows all files on the disk including the “hidden” SMF files in the ZYT and ZYS subvolumes.
- DSAP supports the BYSUBVOL, BYUSER, DETAIL, USER, SEPARATE, WORKFILE, and SUMMARY options when it is invoked on a virtual disk volume.

Exiting DSAP

Exit DSAP by pressing the Break key. DSAP continues to run in the background until it is finished. If your terminal is the output device, you can reenter DSAP by typing PAUSE at the TACL prompt.

Stopping DSAP

To stop DSAP, press the Break key and type STOP at the TACL prompt. (Without parameters, STOP applies to the process that the current TACL most recently started.)

DSAP Options

This subsection describes each option in DSAP.

`AGE { [ OVER ] `*`number`*` | UNDER `*`number`*` }`

OVER *number* causes the DETAIL report to include all of the files in the file set whose most recent modification occurred *number* days (or more) ago.

UNDER *number* causes the DETAIL report to include all of the files in the file set modified in the last *number* days.

number is an integer in the range 0 through 32,767.

`ANALYSIS`

generates all of the analysis reports (except the DETAIL report). That is, DSAP generates the SUMMARY, FREESPACE, FILESPACE, FILESIZE, BYSUBVOL,

and BYUSER reports. The FREESPACE report is not generated for virtual volumes. Each of these reports is described in Table 5-2, Summary of DSAP Report Types, on page 5-17.

AUDITED

causes the DETAIL report to include all of the files in the file set that are audited by the TMF.

BROKEN

causes the DETAIL report to include the files in the file set that are marked as broken or corrupt (due to a detected inconsistency in the file or a read-write I/O error), or the files whose contents are in question.

BYSUBVOL

generates the subvolume summary report, an analysis of space allocation for each subvolume. Includes support for SMF virtual volumes. See Figure 5-6, Sample Subvolume Summary Report, on page 5-23.

BYUSER

generates the user summary report, an analysis of space allocation for each user ID. Includes support for SMF virtual volumes. See Figure 5-7, Sample User Summary Report, on page 5-24.

CRASHOPENED

causes the DETAIL report to include all of the files in the file set that are in the crash-opened state. Such a file was open at the time of a total system crash or when the disk on which it is located became unavailable. The crash-opened state is cleared when the file is opened successfully.

DEALLOC *number*

causes the DETAIL report to include all the files in the file set with *number* or more pages of deallocatable extents. The *number* is an integer in the range 0 through 2,147,483,647 (2). See Figure 5-8, Sample User Detail Report, on page 5-25.

DETAIL

generates the user DETAIL report in addition to the summary report. The DETAIL report lists the names of all the files in the file set that meet at least one criterion specified by *detail-selection-options*. The DETAIL report is sorted by user ID and file name and includes support for SMF virtual volumes.

If you do not specify any *detail-selection-options*, DSAP lists all of the files in the file set. You do not need to specify DETAIL if you specify one or more *detail-selection-options*.

ENSCRIBE

causes the DETAIL report to include the files in the file set that are Enscribe files and SQL object program files.

EXPIRED

causes the DETAIL report to include only those files with an expiration date (NOPURGEUNTIL) of today or earlier.

EXTENTCHECK

requests DSAP to make a repeated effort to perform an extent consistency check. Whenever all files are being analyzed (that is, when you do not specify the USER detail selection option), DSAP normally checks the consistency of the allocated and free-space disk extents and displays the results in the space allocation consistency-analysis portion of the summary report. To perform this check, DSAP must be able to copy the disk directory when it is not changing.

If the EXTENTCHECK option is not specified, then directory changes occur during the copy attempt, DSAP continues the summary report without checking the extent consistency and issues this message:

```
Free space changed during directory copy.
Space allocations are too frequent.
Extent check cannot be performed.
```

If you specify EXTENTCHECK, DSAP tries to copy the directory three times and issues these messages:

```
Free space changed during directory copy.
Retrying.
```

If DSAP cannot copy the directory in three attempts without the directory changing, it terminates, issues these messages, and does not complete the summary report:

```
Free space changed during directory copy.
Space allocations are too frequent.
Extent check cannot be performed.
DSAP is being terminated.
```

If the disk you are checking is exceptionally busy (for example, $SYSTEM), try to perform EXTENTCHECK when the disk is less active.

FILESIZE

generates the file-size distribution report, an analysis of the distribution of file sizes. See Figure 5-5, Sample File-Size Distribution Report (FILESIZE), on page 5-22.

FILESPACE

generates the file extent-size distribution report, an analysis of the distribution of allocated extent sizes. The presence of a virtual volume causes all percentages that compare space use with the size of the disk to be displayed as 0.0 percent.

See Figure 5-4, Sample File Extent-Size Distribution Report (FILESPACE), on page 5-21.

FREESPACE

generates the free-space distribution report, an analysis of the distribution of free-space extent sizes. This option is ignored on a virtual volume. It also causes all percentages that compare space use with the size of the disk to be displayed as 0.0 percent. See Figure 5-3, Sample Free-Space Distribution Report, on page 5-20.

LICENSED

causes the DETAIL report to include all of the licensed files in the file set.

NEWFORMAT (G-series only)

causes the DETAIL report to display in the new format that supports 36 GB disks. You must use this option to run DSAP on any disk larger than 18 GB.

OPENED { [OVER] *number* | UNDER *number* }

OVER *number* causes the DETAIL report to include all of the files in the file set whose most recent open date occurred *number* days or more ago.

UNDER *number* causes the DETAIL report to include all of the files in the file set that were opened in the last *number* days.

number is an integer in the range 0 through 32,767.

PARTITIONED

causes the DETAIL report to include all of the partitioned files in the file set. DSAP does not support the PARTITIONED option for SQL/MX format objects.

PROGID

causes the DETAIL report to include all of the files in the file set that have the PROGID option set. The PROGID option is set by the FUP SECURE command.

ROLLFORWARD[NEEDED]

causes the DETAIL report to include all of the TMF-audited files in the file set that are marked needing a ROLLFORWARD because a crash-recovery process failed.

SEPARATE

causes DSAP to format the DETAIL report so that the portion of the report belonging to each user begins on a new page. This option also supports SMF virtual volumes.

SHORT

produces a brief report about disk-space capacity and available free space. DSAP produces this report faster than standard DSAP report.

Any virtual volume names are displayed, and an error message is used to indicate their presence.

It does not read the disk directory or perform any consistency checks for space allocation to create this report. See Figure 5-9, Sample of the Short Report, on page 5-27.

Note. The SHORT option has no parameters. You cannot use SHORT with any other DSAP options.

SHOWNAMEMAP

displays the physical/virtual name-mapping when used with any of the DETAIL report options.

This report is generated in 132-column format.

If the disk specified is a physical disk, then the column header displays “Virtual Name” and the virtual name of the file if one exists; otherwise, the column header displays “Physical Name” and the physical name of the file.

This option is not part of the selection criteria, and it applies only when a DETAIL option is also specified.

SIZE { [OVER] *number* | UNDER *number* }

OVER *number* causes the DETAIL report to include all of the files in the file set with *number* pages or more.

UNDER *number* causes the DETAIL report to include all of the files in the file set with fewer than *number* pages.

number is an integer in the range 0 through 2,147,483,647 (2GB).

SPACE

generates both the free-space distribution and file extent-size distribution reports; that is, FREESPACE plus FILESPACE. The FREESPACE report is not generated for virtual volumes.

SQL

causes the DETAIL report to include in the file set all of the SQL/MX format objects that are SQL files and SQL object program files.

SUMMARY

generates the summary report. The DSAP utility automatically generates the summary report unless you specify the SHORT option. This option also supports SMF virtual volumes.

If you only want to see the summary report, specify SUMMARY (or omit all report options). For example, these commands produce the same results:

```
1> DSAP $volume SUMMARY
1> DSAP $volume
```

The Space Allocation Consistency Analysis and the Media Failure Analysis reports are not generated for a virtual volume, and a message is printed to indicate that it is a virtual volume. This option also causes all percentages that compare space use with the size of the disk to be displayed as 0.0 percent.

TEMPORARY

causes the DETAIL report to include all of the temporary files in the file set.

TERAFORM (G-series only)

causes the DSAP reports to display in the format that supports 1 terabyte disks. You must use TERAFORM or TABULAR options to run DSAP on any disk larger than 72 GB.

TABULAR

formats the output for different DSAP options by arranging the data in columns. The report can be directly ported to any popular spreadsheet program.

UNEXPIRED

causes the DETAIL report to include only those files with an expiration date (NOPURGEUNTIL) that is later than the current date.

UNUSED *number*

causes the DETAIL report to include all of the files in the file set with *number* or more pages of unused space. The *number* is an integer in the range 0 through 2,147,483,647 (2GB).

USER

instructs DSAP to analyze only the files owned by a single user (or by a single group of users). If you omit the USER option, DSAP analyzes all of the files on the disk. This option also supports SMF virtual volumes.

USER without any qualification specifies the current user. It must be the last option specified in the DSAP command or be followed by a comma.

group-name.user-name | *group-number,user-number*

specifies a particular user in a particular group.

-1

specifies the super ID (255,255).

group-name.* | *group-number* , *

specifies all users belonging to the specified group name or number.

WORKFILE { $*volume* | *filename* }

lets you specify either a temporary or permanent file. If you do not specify a work file, DSAP uses its swap-file volume as a default location for the work file.

If you do not specify a permanent work file, DSAP creates a temporary work file.

$*volume*

specifies a volume for the allocation of a temporary file. This can also be an SMF virtual disk volume.

filename

specifies a file name for the allocation of a permanent work file (used by the Enform product to generate additional reports). The *filename* is of the form $*volume*.*subvolume*.*file-id*, or simply *file-id*, if you want to put the work file into the current subvolume.

DSAP $(volume-set), including $*, with a permanent workfile option will display reports for the selected volumes as long as the collective workfile size is less than or equal to (500MB - SQL Buffer size). SQL Buffer size = 32 pages. (1Page = 2048Bytes.) It should display this message for the specific disk on which workfile size exceeds 500MB - SQL Buffer:

```
"Unable to allocate enough space for workfile. Try again with
an empty workfile."
```

The SQL buffer size might be increased later to support the SQL/MX tables with a very large number of partitions.

For SQL/MX objects

For DSAP reports generated from the permanent work file when you use the WORKFILE option, DSAP:

- Displays “*SQL” in the SECURITY column
- Displays the name of the resource fork for the user files in the CATALOG column.

- Displays in the RCODE:

  0 Executables

  1 Tables, views, LOB tables, and so on

  2 Indexes

- Displays SQL for the FILE-KIND column
- Differentiates between SQL/MP and SQL/MX objects
- Differentiates between SQL/MX tables, views, LOB tables, and so on in SQL-OBJECT-TYPE
- Differentiates between ANSI and materialized views in SQL-VIEW-TYPE

Completion Information

DSAP returns completion information to the process that started it. The completion code indicates how successful DSAP ran. Its primary purpose is to support batch processing by allowing conditional execution of subsequent processes based on the success of the DSAP process. If TACL started the DSAP process, the completion information is stored in a TACL variable (:_COMPLETION), where you can examine it by using TACL functions.

The TACL also displays the completion information at the home terminal, unless the DSAP process completes normally without any warnings or errors. For more information about completion information and batch processing, see the *TACL Programming Guide*.

The completion information returned by DSAP includes:

Field	Meaning
completioncode	An integer that describes how DSAP terminated, as described in Table 5-1, DSAP Completion Codes, on page 5-14.
terminationinfo	An integer that describes the nature of errors or warnings, as described in Table 5-1, DSAP Completion Codes, on page 5-14.
subsystem	The DSAP subsystem ID, TANDEM.32.*release*, where *release* is the product version of the DSAP utility (C30, for example).
text	A character string describing the code in *terminationinfo*. See Table 5-1, DSAP Completion Codes, on page 5-14.
textlength	The number of characters in *text*.

Table 5-1. DSAP Completion Codes

Code	Explanation
0	A normal, voluntary termination of DSAP with no errors or warnings.
1	A normal, voluntary termination of DSAP with warnings. The disk report was created normally, but DSAP reported disk structure diagnostics, space allocation problems, or a nonfatal internal program error. DSAP includes this termination information with the completion code. **terminationinfo** **text** 1 Internal error 2 Disk error or warning
2	An abnormal, voluntary termination with errors. DSAP generated an incomplete report because of a disk error or an internal error. DSAP includes this termination information with the completion code. **terminationinfo** **text** 1 Internal error 2 Disk error or warning
3	An abnormal, voluntary termination with fatal errors. DSAP did not produce a report because of severe disk errors, severe structural errors, a fatal program error, or an invalid user command. DSAP includes this termination information with the completion code. **terminationinfo** **text** 1 Internal error 2 Disk error or warning 3 Command syntax error
4	DSAP never started, so the completion code is generated by the TACL process that is executing DSAP. The *terminationinfo* field holds an error code from the process creation procedure.
5	DSAP ended abnormally and returned 0 in the *terminationinfo* field.

Guidelines: Running DSAP

- DSAP has no internal security restrictions except that it requires licensing. The system manager controls access to DSAP by using the standard disk-file security scheme. For more information, see Security on page 5-2.
- The version of DSAP must match the version of the NonStop operating system to ensure compatibility with low-level system interfaces:

DSAP Version	Compatible Operating System Versions
D42	D42 or later
D45	D45 or later
D48	D48.01 or later
G06	G06.06 or later
G07	G06.23 or later
Pre-D42	Pre-D42

- DSAP allocates a large extended data segment and swap file to hold file labels and extent information. If the current work file volume is nearly full, DSAP might try to map the extended data segment to a temporary file on that disk but terminate with file-system errors 31 (unable to obtain file system buffer space) or 43 (unable to obtain disk space). To avoid this, use the WORKFILE option to specify a disk that is not as full as the current work file volume.
- When DSAP needs to allocate 500 MB of workspace, it displays “WARNING: 500MB SPACE WILL BE REQUIRED” before proceeding. If the required space is not available, DSAP abends and displays “DIRECTORY WORKFILE CREATION/OPEN/MAPPING ERROR: 43. TRY AGAIN WITH 500MB WORKSPACE”.
- Extent overlaps (double allocated pages) indicate a hardware or software error. A doubly allocated page is an error condition and occurs when the same page is allocated more than once to a single file, two different files, or a file and free space. Report such errors to your service provider.
- Specify a permanent work file only if you want to retain file-label information in machine-readable form. Using the Enform product or a custom program, you can then generate additional reports with more complex selection conditions than those offered by DSAP options. For more information, see the *ENFORM User’s Guide*.
- Free-space extents are usually tracked as disk extents by the free-space table. Allocated-space extents are tracked by the disk directory. If an extent is not recorded in either place, it is lost. Lost free space results from a total system failure or a double processor failure.

  You can recover lost free space in two ways:

  - By backing up and restoring system files

- By using the SCF CONTROL DISK, REBUILDDFS command (See the *SCF Reference Manual for the Storage Subsystem*.)

The simplest method for recovering lost free space is to use REBUILDDFS because it does not need tapes.

- Because DSAP creates reports about the contents of disk volumes, you can also use it on virtual volumes (DSAP $L), but some of the report options are ignored (such as any reports that deal with free disk space). This is similar to DSAP interaction with the USER option.
- DSAP/DCOM have a DSAPCSTM file to customize the DSAP environment. DSAPCSTM file features are:
  - You can customize the DSAP environment using the DSAPCSTM file.
  - You can specify any of the DSAP OPTIONS in the DSAPCSTM file.
  - The DSAP OPTIONS specified in the DSAPCSTM file are taken as default. That is, the options in the DSAPCSTM file are appended to the DSAP command which you specify at the TACL prompt.
  - The DSAPCSTM file exists in the DEFAULT VOL-SUBVOL of the user running DSAP.
  - DSAP creates the DSAPCSTM file in your DEFAULT VOL-SUBVOL the first time you run DSAP or when the file has been purged.
  - If the DSAPCSTM is already present in your DEFAULT VOL-SUBVOL, DSAP reads information from this file.
  - DSAP closes the custom file after reading it.
  - If you do not own the DSAPCSTM file and its security is OOOO, DSAP options in the file are not executed and this error message is displayed:

    ```
    Dsapcstm error 48 encountered as the dsapcstm file does
    not belong to user
    ```
- DSAP supports SQL format 2 files for SQL/MP.
- DSAP supports up to 4,500,000 files on a single disk volume.

Report Formats

Table 5-2. Summary of DSAP Report Types

Report Options	Report Name	Report Contents
SUMMARY	Summary	Physical disk description, space use summary, exceptional condition report
USER	User Summary	Summary of space use for a specific user
FREESPACE	Free Space Distribution	Space distribution analysis of free space extents
FILESPACE	File Extent Size Distribution	Space distribution analysis of allocated file extents
SPACE	Combines Free Space and File Extent Size reports	Space distribution analysis of free space extents and allocated file extents
FILESIZE	File Size Distribution	Space distribution analysis of file sizes
BYSUBVOL	Subvolume Summary	Space allocation for each subvolume
BYUSER	User Summary	Space allocation for each user ID
ANALYSIS	Summary, Free Space, File Extent, File Size, BYUSER, and Subvolume reports	Provides all of the reports listed for it under report name
DETAIL	User Detail	File names and types based on detail selection options used
SHORT	Short	Brief report of disk capacity and free space; disk directory is not read
NEWFORMAT	User Detail	File names and types based on detail selection options, used but with increased column width. Must use this option for disks larger than 36 GB.
TABULAR	User Detail	File names and types based on detail selection options used in tabular format.
TERAFORM	Use with any report except SHORT	Reports based on selection options used but with increased column width. You must use this option for disks larger than 72 GB.
Other options:		
EXTENTCHECK	Summary	Adds consistency check to Summary Report
WORKFILE	Use with any report option except SHORT	Specifies work file location

Note. Total pages for unstructured files on a 512-byte disk will include the checksum space occupied in such files.

Summary Report

The summary report has three sections:

- Physical disk description
- Summary of space use
- Exceptional condition report

Physical Disk Description

The physical disk description includes device type and subtype, the approximate formatted disk capacity, and the actual usable disk capacity (the disk space available for the disk directory, free-space table, volume label, disk files, and so on). The difference between the formatted and usable disk capacity is due to the areas reserved for defect handling and maintenance.

Summary of Space Use

For the specified disk, the summary of space use lists the total number of:

- Free pages (available for file allocation)
- Allocated pages (assigned to a file)
- Unused pages (assigned to a file but beyond the current end of file)
- Deallocatable pages (unused pages in assigned but empty extents)

To reclaim deallocatable space, use the FUP DEALLOCATE command to return unused extents.

Exceptional Condition Report

The exceptional condition report consists of:

- Space-allocation consistency analysis indicates the consistency of free space and allocated space and notes any overlapped extents, lost free space, unspared defective sectors, or free extents out of order in the free-space table. For information on lost free space and extent overlaps, see Guidelines: Running DSAP on page 5-15.
- The summary of space allocation anomalies, which divides the space-allocation consistency analysis into disk free-space table inconsistencies, doubly allocated pages, and lost free-space pages.
- The media-failure analysis, which reports if the disk (primary or mirror) is DOWN or has unspared defective sectors. If the disk has any unspared defective sectors, DSAP displays:

```
{ Primary } disk has num unspared defective sectors.
{ Mirror  }
```

```
   Please eliminate bad sectors using the
   PUP SPARE command.
```

When DSAP performs a summary analysis for a single user, the summary report is truncated (see Figure 5-1). These are examples of commands that produce this report:

```
1> DSAP $DATA, USER KIN.MARTIN
2> DSAP $DATA, USER 3,85
```

Figure 5-1. Sample Summary Report for a Single User

```
Disk Space Analysis Program -- T9543D42 - (15AUG96) -- 8/15/96    16:17:21
Tandem Computers Incorporated 1981, 1983, 1985-1996
Summary of space use for KIN.MARTIN on $DATA
  2,187 allocated pages in 238 files in 867 extents (1.0%).
    218 unused pages in 132 files (0.1%).
      0 deallocatable extent pages in 0 files (0.0%).
No SQL views.
```

To generate a summary report for a full-volume analysis:

```
1> DSAP $SYSTEM
```

Figure 5-2. Sample Summary Report for an Entire Disk

```
Disk Space Analysis Program -- T9543D30 - (31OCT94) -- 11/8/94    16:21:17
Tandem Computers Incorporated 1981, 1983, 1985-1994
Volume $SYSTEM is logical device 6
Device type is 3, subtype 10 ( 4130 -- 415MB )

    203,014 pages (2048 bytes) on volume
415,772,672 bytes on volume

Summary of space use on $SYSTEM

     39,947 free pages in 580 extents (19.6%).
    161,564 allocated pages in 2,785 files in 7,488 extents (79.5%).
     24,799 unused pages in 1,633 files (12.2%).
      8,436 deallocatable extent pages in 66 files (4.1%).
No SQL views.

Space Allocation Consistency Analysis:

     64 lost free space pages (@page 78289)
     64 lost free space pages (@page 79502)
     64 lost free space pages (@page 157763)
     64 lost free space pages (@page 184545)

Summary of space allocation anomalies:

      0 disk free space table inconsistencies.
      0 doubly-allocated pages.
    256 lost free space pages.

Media Failure Analysis:

Primary disk has no unspared defective sector(s).
Mirror disk has no unspared defective sector(s).
```

In the parentheses of the summary of space use are percentages of the total disk pages on the volume. The allocated pages percentage plus the free pages percentage

might not equal 100 percent. The system tables (containing the disk directory, free-space table, spare tracks table, volume label, and others) use the remaining space.

If DSAP reports that a file has doubly allocated extents (and the message “SQL Shadow” appears after the file name), the file is an SQL table that was dropped, but the drop is not yet committed. The file is invisible, but its extents are still allocated until the drop is committed. If there are no outstanding drop transactions, use the SQLCI CLEANUP command to remove this entry. For more information, see the *SQL/MP Installation and Management Guide*.

Space Distribution Reports

DSAP performs space-distribution analysis for these types of space:

Type of Space	Report Option
Free-space extents	FREESPACE
Allocated file extents	FILESPACE
Both free and allocated extents	SPACE
Entire files	FILESIZE

All report types first display the summary report. (See Figure 5-2 on page 5-19.)

To produce a free-space distribution report:

```
1> DSAP $volume FREESPACE
```

For each extent size, the free-space distribution report displays:

- Number of free extents
- Total pages (of 2048 bytes each) occupied by the free extents
- Total pages (of 2048 bytes each) occupied by the free extents of that size or smaller
- Free space used as a percentage of the total free space on the disk
- Free space used as a percentage of total disk space

Figure 5-3. Sample Free-Space Distribution Report

```
Free Space Distribution

---------Free Extents---------|----Free Space----|----Disk Space----|
Extent                Cumulative|---Distribution---|---Distribution---|
 Size   Count     Pages    Pages | Space  Cumulative| Space  Cumulative|
    2      30        60       60 |   0.0%     0.0% |   0.0%     0.0%  |
    4      16        64      124 |   0.0%     0.1% |   0.0%     0.0%  |
    6       2        12      136 |   0.0%     0.1% |   0.0%     0.0%  |
    8       2        16      152 |   0.0%     0.1% |   0.0%     0.0%  |
   20       1        20      172 |   0.0%     0.1% |   0.0%     0.0%  |
   64       3       192      364 |   0.1%     0.3% |   0.0%     0.1%  |
12057       1     12057    12421 |  99.6%   100.0% |  59.1%    59.3%  |
```

To produce a file extent-size distribution report:

```
1> DSAP $volume FILESPACE
```

This report displays information in the same manner as the free-space distribution report.

Figure 5-4. Sample File Extent-Size Distribution Report (FILESPACE)

```
File Extent Size Distribution

----Allocated File Extents---|----File Space----|----Disk Space----|
Extent             Cumulative|---Distribution---|---Distribution---|
Size  Count     Pages   Pages | Space  Cumulative| Space  Cumulative|
   1     29        29      29 |   0.0%    0.0%   |   0.0%    0.0%   |
   2   5866     11732   11761 |  14.3%   14.3%   |   5.7%    5.7%   |
   4   1288      5152   16913 |   6.3%   20.6%   |   2.5%    8.3%   |
   6    249      1494   18407 |   1.8%   22.5%   |   0.7%    9.0%   |
   8    597      4776   23183 |   5.8%   28.3%   |   2.3%   11.4%   |
  10    111      1110   24293 |   1.3%   29.7%   |   0.5%   11.9%   |
  12     56       672   24965 |   0.8%   30.5%   |   0.3%   12.2%   |
   .      .         .       .      .       .          .       .
   .      .         .       .      .       .          .       .
   .      .         .       .      .       .          .       .
  34     10       340   66296 |   0.4%   81.0%   |   0.1%   32.6%   |
  36      6       216   66512 |   0.2%   81.3%   |   0.1%   32.7%   |
  38      6       228   66740 |   0.2%   81.6%   |   0.1%   32.8%   |
  39      1        39   66779 |   0.0%   81.6%   |   0.0%   32.8%   |
  40      8       320   67099 |   0.3%   82.0%   |   0.1%   33.0%   |
  42      1        42   67141 |   0.0%   82.1%   |   0.0%   33.0%   |
   .      .         .       .      .       .          .       .
   .      .         .       .      .       .          .       .
   .      .         .       .      .       .          .       .
  98      1        98   68903 |   0.1%   84.2%   |   0.0%   33.9%   |
 100      1       100   69003 |   0.1%   84.4%   |   0.0%   33.9%   |
 102      1       102   69105 |   0.1%   84.5%   |   0.0%   34.0%   |
 120      1       120   69225 |   0.1%   84.6%   |   0.0%   34.0%   |
 182      1       182   69407 |   0.2%   84.8%   |   0.0%   34.1%   |
 200     16      3200   72607 |   3.9%   88.8%   |   1.5%   35.7%   |
2964      1      2964   75571 |   3.6%   92.4%   |   1.4%   37.2%   |
2984      1      2984   78555 |   3.6%   96.0%   |   1.4%   38.6%   |
3198      1      3198   81753 |   3.9%  100.0%   |   1.5%   40.2%   |
```

To produce a file-size distribution report:

```
1> DSAP $volume FILESIZE
```

This report displays information in the same manner as the free-space distribution report.

Figure 5-5. Sample File-Size Distribution Report (FILESIZE)

```
File Size Distribution

-------Total File Size--------|----File Space----|----Disk Space----|
File                  Cumulative|---Distribution---|---Distribution---|
Size  Count      Pages    Pages  | Space  Cumulative| Space  Cumulative|
   0     46          0        0 |   0.0%    0.0%   |   0.0%    0.0%   |
   1      7          7        7 |   0.0%    0.0%   |   0.0%    0.0%   |
   2   1058       2116     2123 |   2.5%    2.5%   |   1.0%    1.0%   |
   3      4         12     2135 |   0.0%    2.6%   |   0.0%    1.0%   |
   4    606       2424     4559 |   2.9%    5.5%   |   1.1%    2.2%   |
   6    221       1326     5885 |   1.6%    7.1%   |   0.6%    2.8%   |
   .      .          .        . |     .       .    |     .       .    |
   .      .          .        . |     .       .    |     .       .    |
   .      .          .        . |     .       .    |     .       .    |
  26     11        286    17447 |   0.3%   21.3%   |   0.1%    8.5%   |
  28     17        476    17923 |   0.5%   21.9%   |   0.2%    8.8%   |
  30     41       1230    19153 |   1.5%   23.4%   |   0.6%    9.4%   |
  32     34       1088    20241 |   1.3%   24.7%   |   0.5%    9.9%   |
  33      1         33    20274 |   0.0%   24.7%   |   0.0%    9.9%   |
  34      6        204    20478 |   0.2%   25.0%   |   0.1%   10.0%   |
  36     73       2628    23106 |   3.2%   28.2%   |   1.2%   11.3%   |
   .      .          .        . |     .       .    |     .       .    |
   .      .          .        . |     .       .    |     .       .    |
   .      .          .        . |     .       .    |     .       .    |
 434      1        434    46539 |   0.5%   56.9%   |   0.2%   22.9%   |
 450      1        450    46989 |   0.5%   57.4%   |   0.2%   23.1%   |
 480      2        960    47949 |   1.1%   58.6%   |   0.4%   23.6%   |
 510      1        510    48459 |   0.6%   59.2%   |   0.2%   23.8%   |
 512      1        512    48971 |   0.6%   59.8%   |   0.2%   24.1%   |
 660      1        660    49631 |   0.8%   60.7%   |   0.3%   24.4%   |
1500      1       1500    51131 |   1.8%   62.5%   |   0.7%   25.1%   |
2070      1       2070    53201 |   2.5%   65.0%   |   1.0%   26.2%   |
2964      1       2964    56165 |   3.6%   68.6%   |   1.4%   27.6%   |
2984      1       2984    59149 |   3.6%   72.3%   |   1.4%   29.1%   |
3198      1       3198    62347 |   3.9%   76.2%   |   1.5%   30.7%   |
4050      1       4050    66397 |   4.9%   81.2%   |   1.9%   32.7%   |
7680      2      15360    81757 |  18.7%  100.0%   |   7.5%   40.2%   |
```

Report by Subvolume

To produce a subvolume summary report:

```
1> DSAP $volume BYSUBVOL
```

Figure 5-6. Sample Subvolume Summary Report

```
Subvol Summary Report
Subvolume Name                 Total Unused Dealloc Large  Min Age  Num
                         Files  Pages  Pages  Pages  File  Mod,Opn  Exp
FREE SPACE                      120409
DISK DIRECTORY                     844
TEMPORARY FILES             23     760   167     64   128    0,  0   23
ARFDEMO                     20     478   113      0   230   14,  0   20
AS87                        32     167    26      0    20  776,---   32
AS88                         3       6     1      0     2  287,---    3
BOBBY                       82   25558   258      0  7680   10,  0   82
BONDS                        2      16     0      0    14  221,---    2
CLARK                      159    2694   192      0   288    1,  0  159
DSMSTD                       9     262    41      0   116  547,---    9
EPS                         14      58     6      0    16  744,---   14
ERD                          2       0     0      0     0  437,---    2
ERIC                       230    2093   207      0   510    9,  0  230
EXMAN                       61    1384   197      0   148  438,---   61
FIERCE                       2     114     1      0    96  472,---    2
GOETHE                       1       2     0      0     2  255,---    1
HANKG                      155    1484   155      0   120    5,  0  155
HENRYSM                      3       8     1      0     4   17, 17    3
LYLE                        76     798   111      8   144    0,  0   76
MARICHAL                   121    2064   458      0   180   21,  0  121
MAYS                        48     348    40      0    42  575,---   48
MCCOVEY                      8      60     2      0    14  282,---    8
PERRY                       13      42    18      0     6    0,  0   13
STACKS                       3      92    21      0    36    0,  0    3
T6946DOC                     4      30     9      0    18  578,---    4
TFOUT                        5     506   488      0   200   65,---    5
THENOLT                      1       0     0      0     0  116,---    1
WORKS                       10      82     3      0    14   73,---   10
```

The first two entries of the subvolume summary report portion always list the number of free-space pages available, followed by the number of pages devoted to the disk directory. The next entry is for any temporary files that exist. The rest of the report is an alphabetically lists subvolumes with this information for each:

- Number of files
- Total pages allocated
- Total pages of unused space
- Pages of unused space in deallocatable extents
- Pages in the largest file
- Age (in days) of the most recently modified file, including the number of days since a file was modified (MOD) and opened (OPN). 1K+ indicates that the age is greater than 999. If a file has never been opened, three dashes (---) appear under OPN.
- Number of expired files

Report by User

To produce a user summary report:

```
1> DSAP $volume BYUSER
```

Figure 5-7. Sample User Summary Report

```
User Summary Report

          User                    Total Unused Dealloc Large  Min Age  Num
     Name            ID    Files  Pages  Pages  Pages   File  Mod,Opn  Exp
FREE SPACE                      120405
DISK DIRECTORY                     844
MANUALS.DAVE         8,2        4    108      2      0     90   10,  8    4
MANUALS.PAT          8,18       2     24     10      0     20    4,  4    2
MANUALS.LYLE         8,37    1000  11638   1669    118    182    0,  0  1K+
MANUALS.????????     8,49      50    584     51      0     80   10,  0   50
MANUALS.HOWARD       8,63     478  31108    980      0   7680   10,  0  478
MANUALS.KIRK         8,89     210   1660    189      0    120    1,  0  210
MANUALS.NOLTE        8,95      43    573     70      0     80  847,  0   43
MANUALS.ERIC         8,112    238   2187    218      0    510    9,  0  238
MANUALS.PETER        8,121     53   1152    226      0    148    7,  4   53
MANUALS.LYLEB        8,122    356   7818   1471      0    180   21,  0  356
MANUALS.KEVIN        8,128    274  15660    803      0   3198    0,  0  274
MANUALS.????????     8,131      1      4      3      0      4   28, 28    1
MANUALS.DILL         8,142    421   8667   3888      0    288    1,  0  421
MANUALS.SHEILA       8,175     16     58     18      0     12    0,  0   16
MANUALS.HENRY        8,182      1      4      3      0      4   28, 28    1
MANUALS.RICHT        8,183      1      4      3      0      4   28, 28    1
SUPER.SUPER        255,255      1    512    504    480    512   24, 28    1
```

The user IDs 8,49 and 8,131 displayed in Figure 5-7 have files allocated on the volume. The question marks mean that the user ID is unknown to the system, which occurs if users are deleted from the system, but the files they owned are not deleted. The sample user summary report displays this information for each user ID:

- Number of files
- Total pages allocated
- Pages of unused space
- Pages in deallocatable extents
- Pages in the largest file
- Age information (two columns), including the number of days since a file was modified (MOD) and opened (OPN). 1K+ indicates that the age is greater than 999. If a file has never been opened, three dashes (---) are displayed in the age open (AGE OPN) column.
- Number of expired files

Detail Report

To produce a user detail report for the volume $DATA:

```
1> DSAP $DATA, DETAIL, DEALLOC 1, UNUSED 50
```

Figure 5-8. Sample User Detail Report

```
User Detail Report

Selection Criteria:  total unused pages >= 50
                     pages in unused extents >= 1
User                                    Total   Unused Dealloc       Age    SQL
Name/ID   Filename    Type   Code    Pages   Pages  Pages  Ext  Mod,Opn  Type

MANUALS.KEVIN
(8,255)  ZLIB.ES                100L  584     151     0     2    441, 9
         ZLIB.ERRLOG       E    541   2048    190    128    16    12,13
MANUALS.DILL
(8,142)  SMGOUT.APPARTF         180   200     196     1     1     10,10
         SMGOUT.APPBRTF         180   200     198     1     1     10,10
         SMGOUT.NEWCHRTF        180   200     198     1     1      2, 2
         SMGOUT.PREFRTF         180   200     196     1     1      2, 2
         SMGOUT.SEC1RTF         180   200     195     1     1     10,10

MANUALS.LYLE
(8,37)   LSTAG1.CONTENTS        101   150      50    16     2    492,---
         LSTAG1.INMEASUX        101   150      60    16     3    492,---
         SQL.MAP           K    900A   16      16    16     1    394,  1    TA
         SQL.MAPI          K    900A   16      16    16     1    394,  1    IN
```

The user detail report displays a detailed list of all the files (or a selected subset of the files) on a disk. To control the contents of this report, use the DSAP command options. For each selected file, the DETAIL report displays:

- Type: A maximum of five characters can be displayed in this field:

  1: R — `Redoneeded` (or `undoneeded`) is necessary for the file or label (base and index tables) or for the label (views).

  ? — The file is crashed. For SQL base and index tables, the file is crash-open or the label is crashed; for protection and shorthand views, the label is crashed. For Enscribe files, the file is crash-open.

  2: B — The file is broken (base and index tables).

  3: C — The data or definition is invalid (SQL protection or SQL shorthand views); the file is corrupt (Enscribe files).

  4: K — Key-sequenced file structure (base and index tables).

  E — Entry-sequenced file structure (base and index tables).

  R — Relative file structure (base and index tables).

  blank — Unstructured file

  5: P — The file is a primary partition (base and index tables).

  X — The file is a secondary partition (Enscribe files only).

Note. Only Enscribe files have secondary partitions. Each partition in an SQL file is marked “P” as a primary partition.

- Code: A maximum of three numeric characters followed by one alphabetic character can be displayed in this field. The first three characters identify the file code. For more information on file codes, see the FUP INFO command in the *File Utility Program (FUP) Reference Manual*.

  Adjacent to the file code are indications of whether a file is:

  A Audited

  L Licensed

  I PROGID (program ID) option set

- SQL Type: A maximum of three characters can be displayed in this field.
  - The first position of the SQL Type field displays one of these values:

    P SQL protection view

    S SQL shorthand view

    blank Any other type of SQL object or Enscribe file

  - The second and third positions of the SQL Type field display one of these values:

    TA SQL table

    IN SQL index

    VI SQL protection view or SQL shorthand view depending on if a P or an S precedes these two characters

    SL SQL shadow label (for uncommitted drops)

    PG SQL object program files

    blanks Enscribe files other than SQL object program files

- These fields of the DETAIL report display values for Enscribe files, SQL tables, and SQL indexes, but are left blank for SQL protection views and SQL shorthand views: CODE, TOTAL PAGES, UNUSED PAGES, DEALLOC PAGES, EXTS, AGE MOD, and AGE OPN.
- If DSAP reports that a file has doubly allocated extents and the message “SQL Shadow” appears after the file name, the file is an SQL table that has been dropped, but the drop is not yet committed. Such a file is invisible but its extents are still allocated until the drop is committed.
- If there are no outstanding drop transactions, use the SQLCI CLEANUP command to remove this entry. For more information, see the *SQL/MP Installation and Management Guide*.

Short Report

To produce a short report:

```
1> DSAP *, SHORT
```

Figure 5-9. Sample of the Short Report

```
            -- Capacity (MB) --        %  -- Free Extents --

Volume    (M)     Total       Free       Free    Count     Biggest

$SYSTEM    Y        246        17.70       7      469        1.84
$DATA              122       101.88      82        8       94.35
$DATA1     Y       233        21.40       9      455        9.28
     .               .            .       .        .           .
     .               .            .       .        .           .
```

For each volume selected, the short report displays:

Volume	Disk volume name
M	Mirrored volume (Y for yes, appears only if both the primary and mirror disks are up; blank for no)
Total	Total capacity of the volume in megabytes (where MB = 1,000,000)
Free	Total free space in megabytes (where MB = 1,000,000)
% Free	Percent of the total capacity that is free
Free Extents	Number of free-space fragments
Biggest	Largest free-space fragment in megabytes

Newformat Report

You must use the NEWFORMAT option for disks 36 GB or larger. While running with discs 36 GB and larger, parameters like FILES and PAGES (TOTAL, UNUSED and DEALLOC) have larger values to display. These parameters do not fit in existing DSAP report file formats, so column widths are increased to avoid the overlapping problem.

The NEWFORMAT option can also be used if the user wants the report in new format irrespective of the disk size for disks of size less than 36 GB.

Tabular Report

Add the TABULAR option if you want the report in the tabular format regardless of the disk size. You can port DSAP reports to a Microsoft PC for use as Microsoft Excel reports. The TABULAR option can work with different DSAP options. TABULAR formats the output report for different DSAP options by introducing new columns. Column width of TABULAR reports varies based on the disk size. The column width is the same as in the NEWFORMAT report for disks of 72 GB or smaller. Otherwise the column width is the same as in the TERAFORM report. If you enter any of the

NEWFORMAT or TERAFORM options with the TABULAR option, the TABULAR report is displayed.

Teraform Report

You must use the TERAFORM option in DSAP command for disks of 72 GB or larger. When you run DSAP on disks of 72 GB or larger, fields that display size in pages (Total Pages, Unused Pages, Dealloc Pages, and Large File) have larger values to display. These parameters do not fit in existing DSAP report file formats, so column widths are increased. You can also use the TERAFORM option to report in Tera format regardless of the disk size for disks smaller than 72 GB.

Note. Some fields cannot be shown on the screen in Tera format. For a full report, redirect the output to a spooler.

DSAP Examples

These examples display the typical uses of DSAP at a variety of user levels, including system managers, group managers, and individual users.

System Manager or System Operator Examples

- To create a hard copy of all analysis reports from DSAP for the volume $DATA, including the list of all user files that the DETAIL report provides:

```
1> DSAP /OUT $S.#PRTR/ $DATA, ANALYSIS, DETAIL
```

- To produce a free-space distribution report about $DATA, which can be used to track space use and detect errors on the disk before they cause problems:

```
1> DSAP $DATA, EXTENTCHECK, FREESPACE
```

- To create a report for distribution to each user of the volume $DATA that describes the files allocated to each of those users:

```
1> DSAP /OUT $S.#LINEPTR/ $DATA, DETAIL, SEPARATE
```

- To perform a general security check:

```
1> DSAP $DATA, DETAIL, LICENSED, PROGID
```

Group Manager Example

- To create a report for distribution to each user in the group ADMIN that describes the files allocated to each user on the volume $DATA:

```
1> DSAP /OUT $S.#LSRPTR/ $DATA, USER ADMIN.*, SEPARATE,&
1>&DETAIL
```

User Examples

- To create a listing of all the files on the volume $DATA that are owned by an individual user (Janet) in the specified group (Sales):

```
1> DSAP $DATA, USER SALES.JANET, DETAIL
```

- T o create only the summary report on the volume $DATA for the current user:

```
1> DSAP $DATA, USER
```

- To create a detail report for the files on volume $DATA that are Enscribe files or SQL object program files:

```
1> DSAP $DATA, DETAIL, ENSCRIBE
```

- To create a detail report for the files on volume $DATA that are SQL files:

```
1> DSAP $DATA, DETAIL, SQL
```

- To create a DETAIL report limited to SQL files that are owned by the current user on volume $DATA:

```
1> DSAP $DATA, SQL, USER
```

- To create a DETAIL report limited to SQL files owned by user 9,99 on volume $DATA:

```
1> DSAP $DATA, SQL, USER 9,99
```

- To create a DETAIL report limited to SQL files owned by users in group 9 on volume $DATA:

```
1> DSAP $DATA, DETAIL, SQL, USER 9,*
```

Format of the Permanent Work File

The work file is a machine-readable file that you can use to create Enform reports based on the output of DSAP. If you do not specify a permanent work file with the WORKFILE option, DSAP uses its swap-file volume as a default location for a temporary work file. The Data Definition Language (DDL) description of the permanent work file is contained in the file $SYSTEM.SYSTEM.DSAPDDL. The DSAPDDL file is released with each version of DSAP.

The order of the records in a permanent work file depends on the order DSAP reports your request. Initially, the records are ordered in the standard collating sequence by their names. The by-user and DETAIL reports sort the records by owner ID and then by file name. The format for the records in the permanent file is provided in a DDL description released with the program.

If the WORKFILE option is specified and the work file does not exist, DSAP creates the file and allocates as many extents as needed based on the amount of free space on the volume. It is best to specify a nonexistent work file, because the file created is

automatically the appropriate size. The extent sizes depend on the capacity of the work file volume.

Although the preferred method is to let DSAP create the work file, you can specify an existing work file, but the DSAP process terminates if the file is not large enough. You can also create a new work file before you run DSAP. The result is the required work file size in bytes:

```
(Number of volumes + 1) *
Average number of files in each volume *
Byte length of the directory record =
Required workfile size in bytes
```

The byte length of the directory record is defined in DSAPDDL.

When you use the WORKFILE option to create a multiple-volume report, the work file contains file records for all of the selected volumes grouped by volume name. The volume name appears as the first eight characters in each file-label record.

If you are the system manager and prefer to use a particular disk for large temporary files, you can specify that volume as the default by modifying DSAP with Binder, as described in Changing the Default Work File Volume on page 5-30.

If you do not specify a WORKFILE option, the work file from DSAP is allocated by default on the volume where the program swap file resides. To alter this default permanently, use Binder.

Any errors that occur when DSAP creates or allocates the work file space are reported in the output file and cause DSAP to terminate.

Always use the current DSAPDDL to analyze the work file. For more information on DDL, see the *Data Definition Language (DDL) Reference Manual*.

Note. If the SMF files are on virtual volumes, the external name field contains the virtual name of the file, and the physical field contains the physical name of the file.

Changing the Default Work File Volume

DSAP lets you alter the default work file volume specification using the BINDER program. The following example illustrates the brief BINDER session that is required to accomplish the change. In the object files of DSAP and DCOM, the default work file volume name is located at the beginning of the procedure SELECT^WORK^VOLUME.

In this example, the volume name for the work file from DSAP is set to $TEMP. The characters for the volume name must be specified as a character pair in quotation marks. $TEMP is specified “$T”, “EM”, “P), “ ” (for a total of eight characters).

```
1> VOLUME $SYSTEM.SYSTEM
2> BIND
@FILE DSAP
@ADD *
@MODIFY CODE SELECT^WORK^VOLUME ASCII 0,"$T","EM","P ","   "
@BUILD DSAP!
```

```
@EXIT
3> FUP LICENSE DSAP
```

The FUP LICENSE command is optional. This command depends on the type of file security you need. You must relicense the program because the BINDER BUILD command removes the old license. For information on the FUP LICENSE command, see the *File Utility Program (FUP) Reference Manual.* For information on the BINDER program, see the *Binder Manual*.

The format of the DSAPDDL template used by Enform to produce written reports is displayed in Figure 5-10, DDL Format of the Permanent Work File, on page 5-32.

Note. An SQL sensitive object file (program) is not considered an SQL file. The file-kind field in the Workfile Report displays SQL sensitive object files (programs) as “ENS” for Enscribe.

Figure 5-10. DDL Format of the Permanent Work File (page 1 of 5)

```
*  DSAPDDL

record directory-record.
file is assigned unstructured.

*                        internal format object name

   02 fnam.
      03 volume          type binary occurs 4 times.
      03 subvol          type binary occurs 4 times.
      03 filename        type binary occurs 4 times.

   02 file               redefines fnam .
      03 volume          pic "x" occurs 8 times.
      03 subvol          pic "x" occurs 8 times.
      03 filename        pic "x" occurs 8 times.

*                        Owner's group/user id

   02 owner              type binary.

   02 owning             redefines owner.
      03 group           type binary 8 unsigned.
      03 user            type binary 8 unsigned.

*                        network external format of object name

   02 external-name      pic x(35).

*                        network external format of physical
*                        object name for SMF files

   02 physical-name      pic x(35).

*                        PROTECTION and decoded protection

   02 protection         type binary unsigned.
   02 security.
      03 read            pic "x".
      03 write           pic "x".
      03 execute         pic "x".
      03 purge           pic "x".

*                        The next two fields 'crtime' and
*                        'crtime-3word' are for internal use
*                        only. They are a fix to reserve 4 words
*                        for the timestamp, and still continue to
*                        use 3-word timestamps.

   02 crtime             type binary.
   02 crtime-3word       type binary occurs 3 times.

*                        timestamp of creation.

   02 creation           type character 6 redefines crtime-3word.
```

Figure 5-10. DDL Format of the Permanent Work File (page 2 of 5)

```
*                       Kind of object:
*                       "ENS" for Enscribe files.
*                       'SQL' for SQL files and SQL sensitive
*                       object files (programs)

   02 file-kind         pic x(3)

*                       Crashopened file: "?", else " ".

   02 crashopened       pic "x" .

*                       Audited file needing roll-forward:
*                       'R', else " ".

   02 rollfwdneeded      pic "x" .

*                       Corrupt file: "C", else " ".

   02 corrupt           pic "x" .

*                       VSN of label creation (internal use)

   02 crvsn             type binary occurs 3 times.

*                       Disk label type (for internal use only)

   02 rcode             type binary.

************************************************************

* The following fields are pertinent to files and tables *

************************************************************

*                       File code.

   02 filecode          type binary unsigned.

*                       FCBFLAGS
*                       FCBFLAGS.<13:15> = filetype

   02 fcbflags          type binary unsigned.

*                       More flags (internal use only)

   02 flags             type binary unsigned.

*                       More flags (internal use only)

   02 flags1            type binary unsigned.

*                       More flags (internal use only)

   02 flags2            type binary unsigned.

*                       File type: U, E, R, K.

   02 filetype          pic "x" .

*                       Audited: "A", else " ".
```

Figure 5-10. DDL Format of the Permanent Work File (page 3 of 5)

```
   02 audited           pic "x" .

*                       Licensed: "L", else " ".
*                       " " for SQL tables and views.

   02 licensed          pic "x".

*                       Primary partition: "P";
*                       Secondary partition: "X", else " ".

   02 partitioned       pic "x" .

*                       Broken: "B", else " ".

   02 broken            pic "x" .

*                       PROGID'ed: "I", else " ".

   02 progid            pic "x" .

*                       Writes are buffered (DP2):
*                       "B", else " ".

   02 buffered          pic "x" .

*                       Compressed audit-checkpoint messages
*                       generated (DP2): "C", else " ".

   02 auditcompress     pic "x" .

*                       Writes to file are verified (DP2):
*                       "V", else " ".

   02 verifiedwrites    pic "x".

*                       Serial mirror writes (DP2):
*                       "S", else " ".

   02 serialwrites      pic "x" .

*                       The next two fields 'ModTime' and
*                       'ModTime-3Word' are for internal use
*                       only. They are a fix to reserve 4 words
*                       for the time stamp, and still continue
*                       to use 3-word time-stamps for now.

   02 modtime           type binary.
   02 modtime-3word     type binary occurs 3 times.

*                       timestamp of latest modification.

   02 lastmod          type character 6 redefines modtime-3word.

*                       The next two fields 'OpenTime' and
*                       'OpenTime-3Word' are internal use only.
*                       They are a fix to reserve 4 words for
                        the time stamp, and still continue to
                        use 3-word time-stamps.
```

Figure 5-10. DDL Format of the Permanent Work File (page 4 of 5)

```
   02 opentime           type binary.
   02 opentime-3word     type binary occurs 3 times.

*                        timestamp of latest open.

   02 lastopen           type character 6 redefines
                         opentime-3word.

*                        Number of extents. Defined as a double-
*                        word because the free-space entry can
*                        exceed 32k.

   02 numexts            type binary 32.

*                        For DP2 files, the maximum number of
*                        Primary extent size, 0 if not pertinent.

   02 primary            type binary 32 unsigned .

*                        Secondary extent size,
*                        0 if not pertinent.

   02 secondary          type binary 32 unsigned.

*                        Third extent size, 0 if not pertinent.

   02 third              type binary unsigned.

*                        Fourth extent size, 0 if not pertinent.

   02 fourth             type binary unsigned.

*  02 fifth              Fifth extent size, 0 if not pertinent

                         02 fifth type binary 32 unsigned

*                        File size in pages, 0 if not pertinent.

   02 filesize           type binary 32.

*                        Byte address of eof, 0 if not pertinent.

   02 eof                type binary 64 unsigned.

*                        Pages in extents past the eof,
*                        0 if not pertinent.

   02 dealloc            type binary 32.

*                        Default internal transfer size for DP2
*                        unstructured files, 0 otherwise.

   02 unstructured-
      buffersize         type binary.

************************************************************
*  The following fields are pertinent only to SQL objects *
************************************************************
```

Figure 5-10. DDL Format of the Permanent Work File (page 5 of 5)

```
*                       "  " for Enscribe files (except SQL
*                       sensitive object files).
*                       "TA" for SQL Base tables.
*                       "PG" for SQL sensitive object files.
*                       "IN" for SQL Index tables.
*                       "VI" for SQL Views.
*                       "SL" for SQL shadow labels (uncommitted
*                       purges/drops).

   02 sql-object-type pic x(2).

*                       "P" for Protection Views.
*                       "S" for Shorthand Views.
*                       " " if not a view.

   02 sql-view-type pic "x".

*                       Catalog associated with this SQL file.
*                       Contains spaces for Enscribe files.

   02 catalog-name pic x(16).

   *                    3 word timestamp of the date before
   *                    which all purges will be rejected.
   *                    Binary 0's for Enscribe files.

   02 expiretime        type binary.
   02 no-purge-3word    type binary occurs 3 times.
   02 no-purge-until    type character 6 redefines
                        no-purge-3word.end
```

Work File Guidelines

If the entire disk is analyzed, the first two file records are named ****FREE.SPACE and ****DISC.DIRECTRY. These records indicate, respectively, the amount of free space and overhead space (directory, free-space table, and so forth) on the disk. Because the first two file records are not completely filled in, they should not be treated as user file records. If a group or user is analyzed, only the files of the group or user appear in the work file.

The order of the file records depends on the selected report options. The by-user and DETAIL reports sort the records by owner ID. Otherwise, records are sorted by subvolume and file name.

Work File Examples

This example runs an Enform query against the work file. The example assumes that the work file is owned by user ID 1,100 and creates a list of all the EDIT files including the corresponding end-of-file markers:

1. Create an EDIT file for the Enform product to use as a source. The file should contain:

```
?DICTIONARY
?ASSIGN Directory-Record TO <workfile-name>
OPEN Directory-Record:
LIST File, EOF WHERE (FileCode = 101)   AND
           (Owning.Group = 1) AND (Owning.User = 100)
CLOSE Directory-Record;
```

2. Run ENFORM:

```
1> ENFORM / IN edit-file, OUT outfile /
```

Alternatively, you can run the Enform product interactively and enter the commands listed in the example.

You can display the creation and last open dates in an Enform report in the same manner as the last modification date. For example, to list the creation time and date:

```
> OPEN directory-record;
> LIST timestamp-time ( creation ) as time *,
> LIST timestamp-date ( creation ) as date *;
```

DSAPCSTM File

The DSAPCSTM file customizes the DSAP environment. Features of the DSAPCSTM file are:

- You can customize the DSAP environment using the DSAPCSTM file.
- You can specify any DSAP options in the DSAPCSTM file.
- The DSAP optioins specified in the DSAPCSTM file are taken as default. That is, the options specified in the DSAPCSTM file are appended to the DSAP command you specify at the TACL prompt.
- The DSAPCSTM file exists in the DEFAULT VOL-SUBVOL of the user running DSAP.
- DSAP creates the DSAPCSTM file in your DEFAULT VOL-SUBVOL the first time you run DSAP or when the file has been purged.
- If the DSAPCSTM is already in your DEFAULT VOL-SUBVOL, DSAP reads information from this file.
- DSAP closes the custom file after reading it.

For usage guidelines, see Guidelines: Running DSAP on page 5-15.

6 PAK/UNPAK

This section describes the PAK and UNPAK utilities, which are used to compress and decompress files on a NonStop system.

PAK compresses Guardian files into a single unstructured archive file on any Expand-connected NonStop system. You can create a self-extracting archive file or use UNPAK to decompress the archive file when needed.

PAK efficiently and compactly collects data, which is useful in situations such as transferring large numbers of files to the GCSC for analysis.

PAK and UNPAK work directly with BACKUP and RESTORE:

- PAK uses BACKUP to read files so its syntax is identical to BACKUP syntax. For example, you can use multiple file sets, wild cards, WHERE conditions and most BACKUP options.
- UNPAK uses RESTORE to decompress files from the archive file.
- PAK starts BACKUP, and UNPAK starts RESTORE.
- To run PAK or UNPAK, the BACKUP and RESTORE programs must be available (that is, they must exist and be secured so you can execute them).

Because PAK and UNPAK use BACKUP and RESTORE, you need to be familiar with these utilities.

PAK Command Options

The syntax to run PAK is:

```
PAK [run-option] archive, volumemode, fileset-list, backup-option
```

For example, to compress all files in subvolume MYSUBVOL into archive file MYPAK:

```
PAK mypak, mysubvol.*
```

PAK uses BACKUP to read the files so the syntax is identical to BACKUP syntax. For example, you can use multiple file sets, wild cards, WHERE conditions, and most BACKUP options.

```
PAK archive, *. * where filecode=101 and modtime > 1 jan 2001, blocksize 28
```

To compress volume-mode files, use the `volumemode` option.

To compress a set of files whose names are stored in an IN file:

```
PAK [run-option] archive, volumemode, @infile, backup-option
```

There are two alternatives:

- The IN file contains one file name per line and nothing else. For example:

  ```
  $vol1.subvol1.filename1
  ```

- The IN file contains the complete syntax that would otherwise come after the archive name (that is, *fileset-list*,*backup-options*). This can be multiple lines long, so the length is effectively unlimited. The syntax is similar to BACKUP IN files except that the tape drive name is not included. For example:

  ```
  $vol1.subvol1.*,blocksize 28
  ```

  In this example, blocksize 28 is the *backup-option*. This syntax is enabled by the -i option.

  For such an IN file containing multiple file sets, the syntax is:

  ```
  (fileset1,fileset2,...),backup-options
  ```

  For example:

  ```
  ($vol1.subvol1.a*,$vol2.subvol2.b*),listall, audited
  ```

In both cases, you can specify additional options in the command line. These options are added to the options in the IN file. PAK does not check for duplicate or contradicting options though BACKUP returns an appropriate error message.

For the @*infile* syntax, PAK creates a temporary file (with the name given by C function tmpname()) and writes the file names and backup options to that file, which is then used as the BACKUP infile.

The temporary file is purged when PAK stops unless it is stopped prematurely. In this case, purge the temporary file (CTMP*nnnn*) explicitly. You can also use the @*infile* syntax can with UNPAK and self-extracting files.

PAK Run Options

Table 6-1. PAK Run Options

Option	Description
-i	@*infile* contains complete BACKUP IN file syntax (*fileset*, *backup-option*)
-s	Creates a self-extracting archive
-c '*params*'	Specifies parameters to use by default for self-extracting files
-after '*command*'	Specifies a command to be automatically executed after self-extraction

Table 6-1. PAK Run Options

Option	Description
-ext *nn*	Specifies the extent size for the archive
-max *nn*	Specifies maxextents
-purge	Purges the file if it already exists
-backup *backup-program*	Specifies the BACKUP object to use
-password *password*	Encrypts the archive
-split *nnn*	Creates a multifile archive
-nocompress	Disables compression
-comment '*text*'	Adds comment text to the archive
-pri *nnn*	Specifies the priority.

PAK run options are not case-sensitive.

`-i`

@*infile* contains complete BACKUP infile syntax (see above)

`-s`

creates a self-extracting archive (object file with file code 700). The self-extracting archive can be extracted by running the archive with any options. For example:

```
run archive, *.*.*, vol $vol.subvol, listall
```

The self-extracting file should not be manipulated with NLD or NOFT or in any other way. Otherwise the self-extracting structure is corrupted, and the archive can no longer be used.

`-c 'params'`

creates a self-extracting archive that automatically executes the specified UNPAK command when run.

For example, to create a self-extracting program SETUP and store an arguments string *.*.* ,... in the file:

```
PAK -s -c '*.*.*, map names *.*.* to *, keep, listall'
SETUP,*
```

If you run SETUP without any arguments:

```
RUN SETUP
```

It decompresses as if it were executed with:

```
RUN SETUP *.*.*, map names *.*.* to *, keep, listall
```

- If any arguments are given, they are used. The stored arguments are ignored.
- If you specify -c, the -s option is implied. You need not specify -s.

`-after '`*`command`*`'`

specifies a TACL command that is automatically executed after the files are restored. Use this option only when creating self-extracting files, in which case you must also specify -s.

If you specify -after '*command*', the self-extracting code executes the specified TACL command after restoring the files. This is intended for software installation to start an automatic software installation macro that can also ask the user for information.

The command is executed as:

1. PAK creates a temporary file (named CTMP*nnnn*) and writes the command to that file.
2. A TACL process is started using this file as the IN file and PAK's OUT file as the OUT file (usually the terminal).
3. The temporary file is purged after the TACL process completes.

If the command needs any input from the user, it must be programmed as a TACL macro that first changes its IN and OUT files to the user's terminal. After that, normal I/O to the terminal can be performed from the macro. However, if the macro starts any programs, they again use $RECEIVE, which usually means that the programs fail. For this reason, the programs should be started with the /INLINE/ option or with the INV or OUTV options.

`-ext` *`extent-size`*

specifies the extent size of the archive (in pages). *`extent-size`* can be any value in the range 1 through 65535. The default is 72 pages. The `-ext` and `-max` options together determine the maximum size of the archive file or each file for multifile archives. See Considerations on page 6-7.

`-max` *`maxextents`*

specifies the *`maxextents`* value of the archive file. *`maxextents`* can be any value in the range 16 through 978. The default is 900. The `-max` and `-ext` options together determine the maximum size of the archive file or each file for multifile archives. See Considerations on page 6-7.

`-backup` *`backup-program`*

specifies the BACKUP object to use. The default is $SYSTEM.SYS*nn*.BACKUP. This can even be on another system (for example, *node*.$SYSTEM.SYSTEM.BACKUP).

`-password` *`password`*

Encrypts the archive contents with 56-bit DES encryption. The password is case-sensitive, and cannot contain any blank spaces.

`-split { nnn | nnnK | nnnM }`

Creates a multifile archive. Each file is at most *nnn* bytes, where *nnn* is any integer up to 2,147,483,647 bytes. You can also use *nnn*K to specify the value in kilobytes, or *nnn*M to specify it in megabytes.

When the first file fills up, a second one is created. The names of the additional files are formed by appending a sequence number to the name of the first file. For example:

```
file
file0001
file0002
```

Note. The maximum number of files that can be generated is determined by the length of the base archive name. For example, if the base archive name length is seven characters, only ten files can be generated. Use smaller lengths to generate more files.

`-nocompress`

disables compression.

`-comment 'text'`

adds comment text to the archive. UNPAK automatically displays the comment. To enter a line break in a comment, enter `\n`. For example:

```
-comment 'First-line\nSecond-line'
```

`-pri {nnn}`

specifies the priority with which the PAK command must run. The `-pri` option can have a value in the range of 2 through 199. The default value is 2. In the following example, the priority is set to 100:

```
-pri 100
```

UNPAK Command Options

UNPAK decompresses an archive file or self-extracting into the given subvolume.

The syntax to run UNPAK is:

```
UNPAK archive,fileset [,restore-options...]
```

For example, to decompress the archive MYPAK:

```
UNPAK mypak, *.*.*, vol $vol.subvol
```

To decompress a self-extracting file:

```
RUN archive [ [,] fileset [,restore-options...] ]
```

As in RESTORE, *fileset* refers to the names of the original files as stored in the archive. To restore the files to another name, use the RESTORE options VOL or MAP NAMES.

To list the files in the archive, use one of:

```
UNPAK archive
UNPAK archive,fileset,LISTONLY
RUN archive,,LISTALL
RUN archive,fileset,LISTONLY
```

Similar to PAK, you can use the syntax '@*infile*' instead of *fileset*:

```
UNPAK [-i] archive,@fileset [,restore-options...]
RUN archive [-i] @fileset [,restore-options...]
```

UNPAK and self-extracting archives also accept these run options:

`-restore` *restore-program*

specifies the RESTORE object file to be used. The default is $SYSTEM.SYS*nn*.RESTORE.

`-password` *password*

specifies the password for an encrypted archive.

Considerations

- PAK compresses files to a single unstructured archive file (code 1729).
- PAK compresses files one record at a time so better compression is achieved with longer block sizes (for example, use BLOCKSIZE 28).
- The self-extracting archive created by PAK is a code 700 file.
- The owner and security of extracted files is the same as for the original files (unless you use MYID). Therefore, it is important to use the MYID option to ensure you can access the extracted files.
- Complete syntax, including BACKUP and RESTORE options and qualified file-set syntax, is available in the PAK/UNPAK online help.
- PAK compresses files only from one node at a time. The node can be different than the one PAK is running on.
- You cannot use PAK on OSS files.
- The -ext and -max options determine the maximum size of the archive file or each file for multifile archives. Using the defaults for each (*extent-size* is 72 pages, *maxextents* is 900) results in a maximum archive file size of about 132 MB (132,710,400 bytes). When specifying these two values, you must ensure that the maximum archive file size is less than 4 GB (*extent-size* * *maxextents* must be less than 2,097,000). Otherwise, when PAK tries to open the file, it returns error 580.

  If you also use the -split option, the maximum size of a single archive file is 2 GB (2,147,483,647 bytes).
- PAK cannot compress OPEN files if the files are opened exclusively. To PAK files opened for write, you can use the SHARE option but the resulting PAK file might be corrupted. For example, you cannot use PAK on edit files opened in read/write mode by EDIT or TEDIT.
- In addition to their own messages, PAK and UNPAK can display any BACKUP or RESTORE messages during their processing.
- Syntax errors in BACKUP/RESTORE commands are not noticed until the BACKUP or RESTORE process starts.
- If a PAK or UNPAK operation fails for any reason, the program displays an appropriate message and stops with completion code 2. For an internal error, the program abends and produces a saveabend file ZZSA*nnnn* if Inspect is running.
- If a PAK operation fails and displays the message "Error 45 when writing to file filename", verify that the split size specified in the PAK options is approximately 2 GB. Also, ensure that the split size is less than the archive file size by around 300 KB.

- The PAK or UNPAK operation used with the /OUT option logs entries for files that are packed in the output file specified in the operation. The output file contains information about the PAK or UNPAK utility and a description for each packed or unpacked file.

  A PAK or UNPAK operation can log entries up to 99999 lines in the output file. If total number of files to be packed or unpacked needs more than 99,999 lines for logging, PAK or UNPAK terminates displaying an error message "Error: Output file exceeded its limit, cannot write in Output file further". In this case PAK or UNPAK does not create any archive file or unpax all files. To PAK or UNPAK files that required more than 99,999 lines for logging records, run the PAK or UNPAK operation without the /OUT option.

7 RESTORE

The RESTORE utility copies files from magnetic tape to disk. The BACKUP utility (discussed in Section 3, BACKUP) performs the complementary function of copying files from disk to magnetic tape. RESTORE lets you:

- Copy files from magnetic tape to disk (The tape must be created with BACKUP.)
- List the contents of a tape without restoring data
- Convert files from one disk-process type to another type (for example, from DP1 to DP2)

Note. Newer operating system software uses the DP2 standard and does not always support the DP1 standard. A conversion might be a one-time maintenance operation (or an infrequent necessity) when an old backup tape is restored.

△ **Caution.** The syntax for Backup and Restore 2.0 is different from the syntax for the Backup and Restore utilities (T9074). If you enter the syntax incorrectly for each file type at the BRCOM prompt, BRCOM might not be able to determine if the command needs to be forwarded to the Backup and Restore utilities (T9074).

You must use the syntax documented in this manual for Enscribe and SQL/MP files. For OSS and SQL/MX files, you must use the syntax documented in the *Backup and Restore 2.0 Manual*.

Note. BRCOM does not support the same tape format as the Backup and Restore utilities (T9074). You cannot mix Enscribe and SQL/MP files on the same tape with OSS and SQL/MX files.

- Use labeled tapes
- Move files from one system to another

Note. Do not run RESTORE under a user ID with Safeguard default protection because all of the files that are restored to disk will have Safeguard protection.

Security

RESTORE is a privileged program. Each customer determines which users are allowed to run it. You can control access to RESTORE in several ways, including:

- Do not license the program file for general use. Without licensing, only the super ID (255,255) can run the RESTORE utility. Also, if RESTORE is not licensed, SQL tables cannot be restored by anyone, including the super ID.
- License the program file and limit access to some subset of users by either setting file security attributes or using a Safeguard access-control list.

  Note. DSM/SCM automatically licenses the RESTORE program file. If you do not use DSM/SCM to install RESTORE, the RESTORE program is not licensed. The super ID can use the FUP LICENSE command to license the RESTORE program file at any time.

- Give the super group Execute access to a PROGID copy of RESTORE with PROGID set to the super group user (255,*n*). You must update this copy of RESTORE when you install a new version of it.

To set the appropriate level of access to RESTORE and all other utilities, see the security policy established by your organization. For more information about securing information on a NonStop system, see the *Security Management Guide*.

BACKUP and RESTORE Interaction

The BACKUP command options you use to create a tape have a direct effect on how the tape can be restored. After creating a backup, you cannot view all the options you used when you ran BACKUP, so note the options you use each time you run BACKUP. For more information, see RESTORE Guidelines and Examples on page 7-44.

RESTORE and BACKUP have two modes of operation:

- File mode, in which RESTORE transfers individual files from tape to system disk.
- Volume mode, in which RESTORE transfers the contents of a whole tape to a whole disk volume. Only the super ID (255,255) can use volume mode.

  Note. Disks with a capacity greater than 2 GB cannot be restored to a version of the operating system prior to D30 when a volume-mode backup is performed.

A tape created by BACKUP using one mode cannot be restored using the other mode.

You should be familiar with these subsections that apply to both utilities:

- Entering BACKUP and RESTORE Commands on page 1-7
- Mounting a Tape on page 1-8
- Tape Formats on page 1-10
- File Sets, File-Set Lists, and Qualified File-Set Lists on page 3-5

RESTORE Syntax

RESTORE has different syntax options in file mode and volume mode. For a concise statement of RESTORE syntax, see Appendix D, Syntax Summaries.

The file-mode RESTORE syntax is:

```
[\node.][$volume.][subvolume.] RESTORE [ /run-option
  [, run-option ] ... / ]

  { tape-device-name                                }
  { ( tape-device-name1, tape-device-name2, ... ) }

  , restore-files

  [ , file-mode-restore-option ] ...
```

The volume-mode RESTORE syntax is:

```
[\node.][$volume.][subvolume.] RESTORE [ /run-option
  [, run-option ] ... / ]

  { tape-device-name                                }
  { ( tape-device-name1, tape-device-name2, ... ) }

  , VOLUMEMODE, { $volume | * }

  [ , volume-mode-restore-option ] ...
```

run-option

is any option for the TACL RUN command. The two most common run options are:

```
IN filename
OUT listfile
```

The IN option specifies the input files, and the OUT option specifies the output files. These files usually override the home terminal as the input or output device. An IN file is a text file that contains the remainder of the command line, including parameters and options. For a complete list of *run-options,* see the description of the TACL RUN command in the *TACL Reference Manual*.

tape-device-name

is the name of the tape drive that contains the unlabeled tape for the RESTORE process or a DEFINE name for a labeled-tape RESTORE process. For unlabeled-tape operations, specify one tape drive for all tape reels, or specify up to four tape

drives for use with the MULTIDRIVE on page 7-21. The *tape-device-name* is in one of these formats:

```
[ \node.]$device
[ \node.]$ldev
define-name
```

Note. The current node in a RESTORE process is where RESTORE is running. When you restore backed up files, run RESTORE on the same node where the TACL process is running and where the files to be restored were backed up.

---

device

is the logical name of the magnetic tape unit, such as $TAPE1.

$ldev

is the device number of the magnetic tape unit, such as $17.

define-name

specifies a DEFINE name of CLASS TAPE or TAPECATALOG for a restore from labeled tape. This DEFINE sends a request to $ZSVR (the labeled-tape server process) for labeled-tape processing.

restore-files

designates the files that are to be restored from tape using one of these formats:

```
fileset
fileset-list
qualified-fileset-list
```

as described in File Sets, File-Set Lists, and Qualified File-Set Lists on page 3-5.

file-mode-restore-option

specifies one or more conditions for the file-mode RESTORE process (see RESTORE Options on page 7-6).

VOLUMEMODE

instructs RESTORE to run in volume mode (only possible on the local node).

volume

specifies the disk volume to be restored from tape. The *volume* must match the disk volume name (on tape) when it is specified.

*

specifies that any disk volume on the tape is to be restored. The asterisk matches any disk volume name on the tape.

volume-mode-restore-option

specifies one or more conditions for the RESTORE process (see RESTORE Options on page 7-6).

File-Mode RESTORE Options

The RESTORE options fall into five categories in file mode. For descriptions, see RESTORE Options on page 7-6.

- Standard RESTORE options (usable with other RESTORE option categories):

AUDITED	MYID	PARTONLY
DETAIL	NOPROMPT	REMOTEIOSIZE
IGNORE	NOSAFEGUARD	START
KEEP	NOT	TAPEDATE
LISTALL	NOREWIND	TURNOFFAUDIT
LISTONLY	NOUNLOAD	VERIFY
MAP NAMES	OPEN	VERIFYTAPE
MULTIDRIVE	PAGELENGTH	VOL

- File-conversion RESTORE options (to convert files from one tape format to another):

DSLACK	ISLACK
EXT	SCRATCHVOL

- Enscribe RESTORE options (to restore Enscribe files):

ALTFILE	NOPURGEUNTIL	PARTOF
PART	REBUILD	

- SQL RESTORE options (to restore SQL files):

AUTOCREATECATALOG	INDEXES	REGISTERONLY	SQLTAPEPARTARRAY
CATALOGS	MAP NAMES	SQLCATALOGS	
COLLATIONS	NOSQLDATA	SQLCOMPILE	

Note. SQL files cannot be restored to optical disk.

- SMF RESTORE option (to place SMF files on specific physical disks within SMF storage pools):

```
PHYSVOL
```

Note. Restoring SMF catalog files (file codes 460 through 462) causes RESTORE to change the file code to 0. Use SMFIXUP to change these files back to their original codes.

Volume-Mode RESTORE Options

These RESTORE options are available in volume mode:

LISTONLY	NOREWIND	TARGET
MULTIDRIVE	NOUNLOAD	VERIFYTAPE
NOPROMPT	RENAME	VOLUMEMODE

RESTORE Options

This subsection describes each of the file-mode and volume-mode RESTORE options.

ALTFILE

The ALTFILE option changes the name of an alternate-key file in the file label of the primary-key file. (Enscribe files only.)

```
ALTFILE (key-file-number,[$volume.][subvolume.]file-id )
```

key-file-number

is an integer in the range 0 through 255 that designates the alternate-key file you are naming.

volume.subvolume.file-id

is the new name of the alternate-key file.

Guidelines

- Because the ALTFILE option affects all *restore-files* that have alternate-key files, use the ALTFILE option only when you are restoring a single primary-key file.
- If you specify the ALTFILE option, you cannot specify the MAP NAMES option. The MAP NAMES option is recommended instead of the ALTFILE option because it provides a complete mechanism for changing file names in the RESTORE utility. For more information, see MAP NAMES on page 7-18.

Example

- This command changes the name of an alternate-key file in the file label of the primary-key file AFILE:

```
1> RESTORE $TAPE, $MYVOL.BASE.AFILE, ALTFILE (2, MYFILE6)
```

- ALTFILE 2 must already be a defined attribute of $MYVOL.BASE.AFILE.

AUDITED

The AUDITED option is the only way to restore files audited by TMF when they were backed up. For more information, see TURNOFFAUDIT on page 7-41.

```
AUDITED [, TURNOFFAUDIT]
```

Guidelines

- For a summary of the RESTORE actions during the restoration of TMF-audited files, see Table 7-1, Interaction Between RESTORE and TMF, on page 7-8.
- TMF must be up during any RESTORE process that includes audited or nonaudited SQL files in *restore-files*.
- If a file was audited by TMF when it was backed up, you must include the AUDITED option in the RESTORE command when restoring the file. If you do not include the AUDITED option, RESTORE skips the audited file and sends the message “* WARNING* Audited File Skipped” to the output file.
- After restoring audited files, use TMF to perform an online dump of the restored audited files before you update them. Purging an audited file erases its online dump information in the TMF catalog. Without an online dump, TMF cannot perform media recovery for the file. For more information on managing files audited by TMF, see the *TMF Planning and Configuration Guide*.
- TMF has its own recovery mechanisms for audited files. However, you might want to use BACKUP and RESTORE to:
  - Transport audited files to another system
  - Archive files and retrieve files that are used infrequently
  - Keep old versions of files
- TMF must be running when you run RESTORE if you want to audit previously audited files when you restore them. If an audited file was backed up and is being restored as a nonaudited file because TMF is not running during the RESTORE process, this warning message is sent to the output file:

  ```
  *WARNING*: COULDN'T MAKE FILE AUDITED
  ```

- You can use BACKUP and RESTORE with the TMFCOM DUMP command for disks containing a combination of audited and nonaudited files, or containing only audited files:

  1. Ensure the safe, complete recovery of files when media recovery is needed. At an SCF prompt, type:

     ```
     1> CONTROL DISK $DATA, REFRESH
     2> BACKUP $TAPE, $DATA.*.*,OPEN,AUDITED, &
     2> &PARTONLY ON ,LISTALL
     3> TMFCOM DUMP FILES $DATA.*.*
     ```

The SCF CONTROL DISK, REFRESH command updates the file labels and writes all changed buffers to the disk.

The backup includes nonaudited and audited files.

The online dump performed by TMFCOM DUMP includes only audited files and ensures that there is an accessible copy of these files if TMF fails.

If the disk ($DATA) becomes damaged or unusable, you can use SCF to recover the files on a new disk that is properly labeled and initialized.

2. Restore the nonaudited files, and recover the audited files:

```
1> RESTORE $TAPE, $DATA.*.*,OPEN,PARTONLY ON,LISTALL
2> TMFCOM RECOVER FILES $DATA.*.*
```

Omitting AUDITED in the RESTORE command causes only nonaudited files to be restored. Then the TMFCOM command recovers the audited files.

If you include the TURNOFFAUDIT option (with the AUDITED option) in the RESTORE command:

- Restored files are not audited even if TMF is running when you run RESTORE.
- The message "*WARNING* Audit Turned Off" is sent to the output file for each file that was audited (but not open) when it was backed up. The message "*WARNING* OPEN, AUDIT Turned Off" is issued for each file that was both open and audited when it was backed up.

- When you restore a file that exists on disk, RESTORE first purges the old file. If it is audited, the purge deletes all corresponding dump entries from the TMF catalog.
- If a file was audited when it was backed up, you cannot restore it if an audited file with the same fully-qualified file name exists and TMF is configured (but not running). Trying to restore such a file causes RESTORE to skip the file and issue the message "*WARNING* File Skipped." You can restore such a file by changing its subvolume name with either the VOL or MAP NAMES option.

Table 7-1. Interaction Between RESTORE and TMF (page 1 of 2)

RESTORE Command Used	Conditions	What RESTORE Does
No AUDITED option		Audited file is skipped
AUDITED	TMF running	File is restored as an audited file.
AUDITED	TMF not running	If file with same name already exists, RESTORE issues Purge Error 82. Otherwise, the file is restored nonaudited, and RESTORE issues a warning message.

Table 7-1. Interaction Between RESTORE and TMF (page 2 of 2)

RESTORE Command Used	Conditions	What RESTORE Does
AUDITED and TURNOFFAUDIT	File does not already exist	File is restored nonaudited.
AUDITED and TURNOFFAUDIT	File exists but is not audited	File is restored nonaudited.
AUDITED and TURNOFFAUDIT	File exists and is audited	If TMF is running, the file is restored nonaudited. Otherwise, RESTORE issues Purge Error 82 and does not restore the file.

△ **Caution.** Restoring any audited file backed up in a WRITE OPEN state can make TMF bring down the whole volume. If you restore these audited files, use them in nonaudited mode.

AUTOCREATECATALOG

The AUTOCREATECATALOG ON option specifies that SQL catalogs are created at the time of the RESTORE process. Otherwise, SQL catalogs are not created.

```
AUTOCREATECATALOG [ ON | OFF ]
```

ON

directs RESTORE to automatically create each SQL catalog that you need (that does not already exist on disk). ON is the default when you specify this option.

OFF

specifies that RESTORE does not create the SQL catalogs. OFF is the same as omitting the AUTOCREATECATALOG option from the RESTORE command.

Guidelines

- When you use AUTOCREATECATALOG ON, an SQL catalog is created and RESTORE inserts appropriate entries with the first instance of an SQL object.
- If you provide a user ID (MYID), specify the AUTOCREATECATALOG ON option, and run RESTORE, you become the catalog owner (and its security). This ensures that the entire SQL object is restored successfully.
- The name of each automatically created catalog is reported in the listing created by RESTORE. Any errors that occur while trying to create a catalog are reported in the same way that other SQL errors are reported.

CATALOGS

The CATALOGS option describes where to register SQL objects that are restored.

```
CATALOG[S] (catalog-name FOR fileset-list
   [ , catalog-name FOR fileset-list ] ... )
```

catalog-name

specifies the name and location of the catalog as [*node*.][$*volume*.]*subvolume*.

fileset-list

specifies the files whose SQL objects are to be registered in *catalog-name*. These files are on the backup tape that is being restored. If you used the MAP NAMES option to change the name of a file, *fileset-list* should be the new name assigned by this option (MAP NAMES).

Guidelines

- To register SQL objects in a different catalog than they were originally registered in, use CATALOGS. If the objects are being restored on a different system, use a fileset that describes the destination fileset, not the original.
- If you use the MAP NAMES option to move SQL objects, make sure to define the MAP NAMES and CATALOGS parameters correctly for the dependent tables, indexes, views, and programs. An incorrect mapping scheme can leave objects invalid or cause the RESTORE process to fail. For more information, see Restoring Using Name Mapping on page 7-53 and Restoring SQL Files on page 7-47.

△ **Caution.** It is critical that the CATALOGS and MAP NAMES mapping lists are complete even when recovering an individual partition:

- If the CATALOGS option is specified, every associated catalog must be mapped.
- If the MAP NAMES option is specified, every partition must be mapped.

If the maps are incomplete or incorrect, severe catalog inconsistencies for the object will result. These inconsistencies are quite difficult to repair, so make sure to avoid them.

- If you specify CATALOGS, you cannot specify the PARTONLY ON or PARTONLY PARTIAL option unless you also specify the KEEP option. You might use these options together with MAP NAMES to move SQL partitions between production and development systems when data volume exceeds practical tape set limitations, or for migration operations. This does not necessitate any purges on the target system.
- If you omit the CATALOGS option, or if there are some SQL objects to restore for which you do not specify a catalog, RESTORE first tries to use the catalog in which the object was described during the BACKUP process. If this catalog no longer

exists or is unavailable and the AUTOCREATECATALOG ON option is not specified, the SQL object is not restored, and an error is reported.

- For complete instructions on moving or renaming a SQL/MP catalog, see the *SQL/MP Installation and Management Guide*.
- For complete instructions on altering or dropping a SQL/MX catalog, see the *SQL/MX Installation and Management Guide*.

Example

To assign $NEW.CAT as the catalog for SQL objects that are restored during this RESTORE process:

```
1> RESTORE $TAPE, *.*.*, CATALOGS ($NEW.CAT FOR *.*.*),&
1> &AUDITED
```

COLLATIONS

The COLLATIONS option specifies collation maps to use when restoring files.

```
COLLATION[S] { ( mapping-rule [, mapping rule ..... )       }
```

mapping-rule

is an ordered set of rules (collation map) that determines how to map a file name:

(*collation-name*, *collation-name*) FOR *simple-fileset-list*

collation-name

is a mapped collation name.

simple-fileset-list

is a general file-set list that includes the names of any objects to restore using the collation.

Guidelines

- You can use the COLLATIONS option with MAP NAMES, CATALOG, or both.
- You can specify up to 10 file-set names or 80 collation names in a collation specification.
- The order of mapping rules is important. The first collation specified whose simple name matches the simple name of a collation referenced in the object being restored is mapped to the restored object. If the restored object is referenced in more than one specified file set, the first file set specified is used. If no collation names match, or if no specified file set contains the new object, no mapping occurs.

- If you omit the COLLATION option, a restored object that uses a collation references the same collation that the original object references.

Examples

```
> RESTORE $TAPE, *.*.*, COLLATION[S] ((coll1 FOR a), (col1, col2) FOR (a,b)),
LISTALL, AUDITED

> RESTORE $TAPE, *.*.*, COLLATION (( col1, col2, col3 ) FOR ($A.*.*, $B.*.*),
(coll1 FOR $A.A.A1)), LISTALL, AUDITED

> RESTORE $TAPE, *.*.*, MAP NAMES ($old1.osubvol.* to $new1.nsubvol.*,
                                   $old2.osubvol.* to $new2.nsubvol.*),
                        COLLATION ($new1.nsubvol.collation_1]
                                   for $new1.nsubvol.*,
                                   $new2.nsubvol.collation_1]
                                       for $new2.nsubvol.*),LISTALL, AUDITED

> RESTORE $TAPE, *.*.*, CATALOG    ($new1.nsubvol for
                                    $new1.nsubvol.*),
                        MAP NAMES   ($old1.osubvol.* to
                                    $new1.nsubvol.*),
                        COLLATION   ($new1.nsubvol.collation_1]
                                    for $new1.nsubvol.*,
                                    $new1.nsubvol.collation_2]
                                       for $new1.nsubvol.*), LISTALL, AUDITED
```

DETAIL

The DETAIL option provides information about the file characteristics information for each file in *restore-files*.

```
DETAIL
```

Guidelines

- The display format provided by the DETAIL option is similar to the one supplied by the FUP INFO, DETAIL command.
- Before restoring files, you can use DETAIL with the LISTONLY option to get information about files that cannot be restored (and to see what BACKUP options were used to create the tape). For more information, see LISTONLY on page 7-18.

Example

To run RESTORE and produce a detailed listing (without restoring the files):

```
1> RESTORE/OUT $MYTERM/ $TAPE, $STAR.GAIL.*, &
1> &DETAIL, LISTONLY
```

The listing for H-series RVUs looks like this:

```
File Mode RESTORE Program - T9074H01 (01MAY2005) (H01)
(C)2000 Compaq (C)2003 Hewlett-Packard Development Company, L.P.
Drives: ($SG)
System: \TCSDRP1  Operating System: H06  Tape Version: 3
Backup options: NO AUDITED, BLOCKSIZE 8, NO IGNORE, NO OPEN, PARTONLY OFF,
INDEXES IMPLICIT
Restore time:  1Jul2005 14:55  Backup time:  1Jul2005 14:55      Page: 1

Tape: 1
$DATA4.SGHOSH.TEST
      ENSCRIBE
      TYPE U
      FORMAT 1
      CODE 100
      EXT ( 6 PAGES, 6 PAGES )
      MAXEXTENTS 900
      OWNER -1
      SECURITY (RWEP): NUNU, PROGID, TRUST ME
      MODIF:  14Jul2004  8:10
      CREATION DATE:   1 Jul 2005, 14:55
      LAST OPEN:  30 Jun 2005, 17:21
      EOF 2076 (0.0% USED)
      EXTENTS ALLOCATED: 1
$DATA4.SGHOSH.VT
      ENSCRIBE
      TYPE U
      FORMAT 1
      CODE 101
      EXT ( 6 PAGES, 6 PAGES )
      MAXEXTENTS 900
      OWNER -1
      SECURITY (RWEP): NUNU
      MODIF:   7Dec2004 11:24
      CREATION DATE:   1 Jul 2005, 14:55
      LAST OPEN:   1 Jul 2005, 14:53
      EOF 340 (0.0% USED)
      EXTENTS ALLOCATED: 1


Summary Information
```

Note. This RESTORE example is supported only on systems running H-series RVUs or J-series RVUs.

DSLACK

The DSLACK option specifies the minimum percentage of slack space for data blocks. It is used only to convert key-sequenced Enscribe files from the DP1 to DP2 format.

```
DSLACK percentage
```

percentage

is an integer in the range 0 through 99. The default value is 0.

EXT

The EXT option specifies new extent sizes for restored files. It is used only to convert Enscribe files from the DP1 to DP2 format.

```
EXT { extent-size   }
    { ( pri-extent-size, sec-extent-size   }
```

Note. The use of extent sizes over 65535 requires format 2.

You can specify these values for *extent-size*, *pri-extent-size* (primary extent size), and *sec-extent-size* (secondary extent size):

`0:512,000,000 [ PAGE[S] ]`

specifies the extent size in pages (2048-byte units). The minimum extent size is one page, so specifying 0 pages allocates one page (2048 bytes). PAGE is the default unit of measurement for the EXT option.

`0:2,147,483,647 BYTE[S]`

specifies the extent size in bytes. RESTORE rounds up to the next full page. For example, if you specify 2047 bytes, RESTORE allocates one page; for 2049, it allocates two pages, and so on.

`0:2,147,483,647 REC[S]`

specifies the extent size based on the current settings for record length, data-block length, index-block length, key-field lengths, and compression settings. RESTORE rounds up to the next full page.

Guidelines

- You cannot include the EXT option if your RESTORE command also contains either the PARTOF or PARTONLY ON option.
- Key-sequenced files sometimes become larger during conversion. To avoid file-system error 45 (file is full), use the EXT option to increase the destination file size.

Examples

- To set primary and secondary extent sizes to 2 pages and 3 pages, respectively, for the files in $DP1.CONV:

```
1> RESTORE $TAPE, $DP1.CONV.*, EXT (2,3)
```

- To set the data slack and index slack values and set file extents to 3:

```
2> RESTORE $TAPE1, *.*.*, DSLACK 20, ISLACK 10, EXT 3
```

IGNORE

The IGNORE option directs RESTORE to ignore certain data errors on tape. The RESTORE utility copies the invalid data to disk (if possible). Otherwise, RESTORE writes zeros in place of invalid data.

```
IGNORE
```

Guidelines

- RESTORE ignores data errors in DP1 files on a sector-by-sector basis (a sector is 512 bytes). For DP2 files, RESTORE ignores errors on a block-by-block basis. For structured files, the length of the block is the data-block length; for unstructured files, the block is the buffer length. For more information about block and buffer sizes, see the description of the SET command in the *File Utility Program (FUP) Reference Manual*.
- If you omit the IGNORE option and a data error occurs, RESTORE purges the corrupt or incomplete file from disk and begins to restore the next file.

INDEXES

The INDEXES option specifies whether the indexes defined for SQL tables are to be automatically restored when the tables are restored.

```
INDEXES [ IMPLICIT | EXPLICIT ]
```

`IMPLICIT`

specifies that the indexes defined for a table are restored automatically when the table is restored. This is the default when you specify INDEXES, and when you omit INDEXES from the RESTORE command when combined with PARTONLY OFF.

`EXPLICIT`

specifies that the indexes defined for a table are not restored automatically when the table is restored. Only indexes explicitly named in *`restore-files`* are restored. EXPLICIT is the default and only allowed value when PARTONLY ON is specified in RESTORE.

Guidelines

- You cannot specify the INDEXES IMPLICIT option and the PARTONLY ON option in the same command.
- You can use INDEXES EXPLICIT option to replace a damaged index file without replacing the corresponding table.

- The INDEXES EXPLICIT option restores the index as it was backed up. No checks are made to assure that the index matches the current state of the table it indexes. To replace a damaged index on an existing table, if you know that the table has not changed since the index was backed up, use RESTORE with the INDEXES EXPLICIT option.
- If a base table and its indexes are backed up with the INDEXES IMPLICIT option, you can restore the base table only by specifying the INDEXES EXPLICIT option and the base table name in the RESTORE command.

Examples

- To restore all SQL indexes whose tables were on the volume $FIN:

```
1> RESTORE $TAPE, $FIN.*.* , INDEXES IMPLICIT, AUDITED
```

  You can produce the same result if you do not specify the IMPLICIT parameter.

- To restore only SQL indexes that have the primary partition on the volume $FIN:

```
1> RESTORE $TAPE1, $FIN.*.*, INDEXES EXPLICIT, AUDITED
```

ISLACK

The ISLACK option sets the minimum slack space for index blocks. Used it only when converting key-sequenced Enscribe files from the DP1 to DP2 format.

```
ISLACK percentage
```

percentage

is an integer in the range 0 through 99. The default value is 0.

KEEP

The KEEP option specifies that a file on tape is restored only if it does not have the same name as an existing disk file.

```
KEEP
```

Guidelines

- If you omit the KEEP option, the file on disk is purged when a file of the same name is restored from tape. However, if the file on tape is an Enscribe file and the file on disk is an SQL file (or vice versa), the file is not purged.
- To specify the MAP NAMES or CATALOG[S] and PARTONLY ON options in the same RESTORE command, you must also specify the KEEP option.

LISTALL

The LISTALL option lists the names of the disk files that are copied successfully from tape (by RESTORE) and the names of disk files that were not restored (because of errors).

```
LISTALL
```

Guideline

If you omit the LISTALL option, RESTORE lists only the file names associated with error messages. RESTORE sends the listing to the OUT *listfile* specified in your command or to the device where you entered the command.

Example

This example is a listing created from the RESTORE command:

```
1> RESTORE $GCC, *.*.*, LISTALL, DETAIL
```

The listing shows some of the BACKUP options used to create the tape. This information helps to verify the contents of the tape.

```
File Mode RESTORE Program - T9074D46 (07SEP98)       System: \BOSTON
Copyright Tandem Computers Incorporated 1981-1998
Drive: $GCC  Operating System: G06  Tape Version: 3
Backup options: NO AUDITED, BLOCKSIZE 8, NO IGNORE, NO OPEN, NO PARTONLY
Restore time:  19Dec1998 10:51  Backup time: 30Feb1998 10:42 Page: 1 Tape: 1
$AEP.LIT.MDVL
    ENSCRIBE
    TYPE  U
    CODE  101
    EXT ( 4 PAGES, 4  PAGES )
    MAXEXTENTS 16
    OWNER 1,1
    SECURITY (RWEP):  CUCU
    MODIF:  2Oct1997  9:57
    CREATION DATE:  30 Aug 1997, 10:51
    EOF 2056 ( 1.6% USED )
    EXTENTS ALLOCATED: 1
$AEP.LIT.VCTRN
    ENSCRIBE
    TYPE  E
    CODE  0
    EXT ( 1 PAGES, 1  PAGES )
    MAXEXTENTS 16
    REC 80
    BLOCK 1024
    OWNER 1,1
    SECURITY (RWEP):  CUCU
    MODIF:  2Oct1997  9:57
    CREATION DATE:  30 Aug 1997, 10:51
    EOF 0 ( 0.0% USED )
    EXTENTS ALLOCATED: 0

Summary Information

Files restored = 2  Files not restored = 0
```

LISTONLY

The LISTONLY option causes RESTORE to run without restoring any files. This option generates a listing of the files that are specified in *restore-files*.

```
LISTONLY
```

Guidelines

- The only file-mode options that are meaningful with the LISTONLY option are DETAIL, NOUNLOAD, and VERIFYTAPE.
- Before restoring files, use the LISTONLY option to see some of the BACKUP options specified when creating the tape. The BACKUP options have a direct effect on how that tape is restored.
- You can use the LISTONLY option with the VOLUMEMODE option, but it provides only an abbreviated listing, showing the disk that was backed up. Unlike the LISTONLY option for FILEMODE, the LISTONLY option for VOLUMEMODE BACKUP does not list the files that were on the disk when the backup was performed.
- For examples, see Viewing the Contents of a Tape on page 7-44.

MAP NAMES

The MAP NAMES option can be used for:

- Renaming files as they are restored
- Restoring files to a new volume or node within the network where they were backed up
- Restoring files to a different network from the one where they were backed up

```
MAP NAME[S] ( old-fileset-list TO new-fileset-list
   [ , old-fileset-list TO new-fileset-list ] ... )
```

old-fileset-list

identifies the files on the backup tape.

new-fileset-list

specifies new names or destinations to be used in the RESTORE process.

Both *old-fileset-list* and *new-fileset-list* have the following format:

```
fileset
( fileset [ , fileset ] ... )
```

Guidelines

For more information on the MAP NAMES option, see Restoring Using Name Mapping on page 7-53. For more information on moving SQL files, see Restoring SQL Files on page 7-47.

- If you specify the MAP NAMES option, you cannot specify the ALTFILE, PART, or VOL options.
- If you specify MAP NAMES or CATALOG[S], you cannot specify the PARTONLY option unless you also specify the KEEP option. You might use these options together to move SQL partitions between production and development systems when data volume exceeds practical tape set limitations, or for migration operations. This does not necessitate any purges on the target system.

△ **Caution.** It is critical that the CATALOGS and MAP NAMES mapping lists are complete even when recovering an individual partition:

- If the CATALOGS option is specified, every associated catalog must be mapped.
- If the MAP NAMES option is specified, every partition must be mapped.

If the maps are incomplete or incorrect, severe catalog inconsistencies for the object will result. These inconsistencies are quite difficult to repair, so make sure to avoid them.

- Verify that no other parts of any backed up SQL objects already exist. Because individual SQL objects are implicitly included in a BACKUP qualified file set, RESTORE could accidentally purge a piece of a SQL object while restoring another piece of that object. The KEEP option does not prevent this kind of purge.
- To restore an SQL object to a different location, use MAP NAMES. If you will restore files to their original locations, do not use MAP NAMES in the command. If the different location is on a different system, make sure to specify both the old and new system names.
- To change file names, use the MAP NAMES option, not the VOL option. The VOL option has limitations handling secondary partitions, alternate-key files, and SQL objects.
- Wild-card characters can be used in the *volume*, *subvolume*, and *file-id* fields of the *old-fileset-list*:

  * (asterisk) matches from 0 through 8 characters in the position where it appears.

  ? (question mark) matches one character in the position where it appears.

  For example, the volume name $SB?? matches all four-character volume names that begin with SB. The file name *CH? matches all file names that end with CH followed by any single character.
- The *new-fileset-list* parameter cannot include the question mark (?) wild-card character, but can include the asterisk (*).

- When more than one renaming pattern is specified in the MAP NAMES option, the first applicable pattern is used if there is a conflict.
- Use the RESTORE AUTOCREATECATALOG and CATALOGS options (with the MAP NAMES option) to move SQL files to another node. In addition to these options, use the SQLCOMPILE ON option to move an SQL program file. The SQL program file must have been compiled before it was backed up.

  Be careful that you define the MAP NAMES and CATALOGS parameters correctly for the dependent tables, indexes, views, and programs. An incorrect mapping scheme can leave the objects invalid or cause the RESTORE process to fail. In addition to the examples here, for more information, see Restoring Using Name Mapping on page 7-53 and Restoring SQL Files on page 7-47.

Examples

- To restore all files on the volume $DATA to the volume $PAYROLL:

```
1> RESTORE $TAPE1, *.*.*, &
1> &MAP NAMES $DATA.*.* TO $PAYROLL.*.*
```

  The files were backed up from and restored to the same node.

- To move the shorthand view $A.SH.VIEW, which refers to the base table $B.VIEW.TAB, to volume $C:

```
1> RESTORE $TAPE, $A.SH.VIEW, &
1> &MAP NAMES ($A.SH.VIEW TO $C.SH.VIEW,
                $B.VIEW.TAB TO $C.VIEW.TAB), AUDITED
```

  Moving the base table with the shorthand view ensures that the view can find and access the table. It accesses table $C.VIEW.TAB if it exists.

- To move the partitions of an SQL file to another volume during a RESTORE process, use a BACKUP command on node \A in the form:

```
BACKUP $TAPE, ($B.B.B, $D.B.B) WHERE SQL, AUDITED
```

  where:

  - \A.$B.B.B is the primary partition of an SQL table that has secondary partitions on \C.$C.B.B and \D.$D.B.B, and...
  - \A.$D.B.B is another SQL object unrelated to \A.$B.B.B

  These names are stored on tape in the form:

```
$B.B.B , \C.$C.B.B , \D.$D.B.B , $D.B.B
```

- To restore these SQL objects to different volumes on node \A:

```
RESTORE $TAPE, *.*.* WHERE SQL, &
&MAP NAMES ($B.*.* TO $F.*.*,
  \C.$C.*.* TO $F1.*.*,
  \D.$D.*.* TO $F2.*.*,
  $D.*.* TO $F3.*.*), AUDITED
```

The file names for secondary partitions \C.$C.B.B and \D.$D.B.B are stored in remote format, so that is how they must be identified in the MAP NAMES option. With the MAP NAMES option, instead of restoring these partitions to their source nodes, they are restored to different volumes on \A. The MAP NAMES option also directs the files \D.$D.B.B and \A.$D.B.B to different locations on node \A to avoid a file-name conflict.

MULTIDRIVE

The MULTIDRIVE option lets you use up to four tape drives for an unlabeled-tape RESTORE process. The sequence of *tape-device-names* specified in the RESTORE command determines the order RESTORE processes the tapes.

```
MULTIDRIVE
```

Guidelines

- The RESTORE utility verifies the sequence of each tape reel before it begins restoring the tape. If a tape is mounted out of sequence, RESTORE issues an error message and waits for the correct tape.
- If a tape drive is unavailable or off-line, or if a tape is not mounted when the RESTORE utility attempts to process on that drive, RESTORE polls the drive and displays:

  ```
  $tape: device not ready -- beginning to poll --
  ```

  The RESTORE process continues in this state until it detects that the drive is ready, with the correct tape mounted.

Example

To restore a four-reel tape set by using two tape drives:

```
1> RESTORE ($TAPE1, $TAPE2), *.*.*, MULTIDRIVE
```

RESTORE reads from alternate tape drives until the tape set is finished. In this example, RESTORE reads tape #1 from $TAPE1, tape #2 from $TAPE2, tape #3 from $TAPE1, and tape #4 from $TAPE2.

MYID

The MYID option sets the owner ID of all of the files that are being restored to that of the user who is running RESTORE. As each file is restored, it is given the default security of the current user.

```
MYID
```

Guideline

If the NOMYID option was used during the BACKUP process, the MYID option of RESTORE is disallowed. For more information, see NOMYID on page 3-26.

△ **Caution.** The MYID option does not actually decrease tape security. The security of BACKUP tapes is always based on physical possession of the tape. Each customer site should have procedures to control access to BACKUP tapes that contain confidential information.

NOPROMPT

The NOPROMPT option instructs RESTORE not to prompt the user before beginning to read each tape but to begin when it detects the tape drive is ready.

```
NOPROMPT
```

Guideline

To mount a tape for a RESTORE process using the NOPROMPT option:

1. Before starting a RESTORE process, remove the write-enable ring from each tape reel.
2. Enter a complete RESTORE command, including the NOPROMPT option.
3. Because you used the NOPROMPT option, this noninteractive message appears:

   ```
   $tape: device not ready -- beginning to poll --
   ```

   Mount the first (or next) tape. For each reel, the process continues when it detects that the drive is ready.
4. When there are no more tape prompts, the process is complete.

NOPURGEUNTIL

The NOPURGEUNTIL option instructs RESTORE to preserve the NOPURGEUNTIL date of the Enscribe files if you set the option before the Enscribe files are backed up. This option only applies to the Enscribe files.

```
NOPURGEUNTIL
```

Guideline

For the SQL files, the NOPURGEUNTIL date is restored by default. Therefore, if the NOPURGEUNTIL option is specified for the SQL files, it is ignored.

NOREWIND

The NOREWIND option directs the BACKUP utility to leave the tape positioned at its current location and leave it online when the BACKUP process is completed. This option lets the tape be labeled for the next BACKUP without having to search for the end of tape. This option is only available starting with product version G06.

```
NOREWIND
```

NOREWIND is mutually exclusive with NOUNLOAD. If neither is specified the default is to unload the tape.

If NOREWIND is specified for an unlabeled tape and an attempt is made to read or write to the tape without first manually repositioning the tape, the utility abends with Error 8103.

NOSAFEGUARD

The NOSAFEGUARD option excludes Safeguard information from a RESTORE process. This option applies only to Enscribe files.

```
NOSAFEGUARD
```

Guidelines

- If you use the NOSAFEGUARD option, files with Safeguard security information are restored but do not retain Safeguard protection.
- When you restore files that have Safeguard protection and do not use the NOSAFEGUARD option, the files retain that protection.

NOSQLDATA

The NOSQLDATA option makes RESTORE recover only the SQL file label for all SQL files in the qualified file set. It does not skip the SQL files entirely. It skips only the data transfer portion of the RESTORE process. NOSQLDATA sets the EOF in the file labels for all SQL files to 0. The result is an empty version of the SQL object, without any SQL data.

```
NOSQLDATA
```

Guidelines

- This option does not affect ENSCRIBE files. Any ENSCRIBE files on the backup tape and included in the qualified file set are recovered with all included data. Only SQL data is omitted.

- An SQL object can effectively be copied using a standard BACKUP tape and the RESTORE MAP NAMES and CATALOG[S] options with NOSQLDATA. This provides a CREATE LIKE function for multiple partition objects.
- When you use this option, this warning message is displayed at RESTORE time:

  ```
  *WARNING-7159* The NOSQLDATA option is active. No SQL data is
  being recovered from this tape.
  ```

- You cannot use NOSQLDATA with the SQLCATALOGS ON option.
- For NonStop SQL/MP tables, you can use NOSQLDATA with the PARTONLY ON or PARTONLY PARTIAL option.

NOT

The NOT option excludes the files named in *not-fileset-list* from the RESTORE process. That is, all the files specified in *restore-files* except the files specified in *not-fileset-list* are restored.

```
NOT not-fileset-list
```

not-fileset-list

is one of:

```
fileset
( fileset [ , fileset ] ... )
```

Guidelines

- The *not-fileset-list* parameter must be completely contained in *restore files*.
- The NOT option is equivalent to the EXCLUDE qualifier in qualified file-set syntax.

Example

To restore all files from volume $SYSTEM except those from $SYSTEM.SYSTEM.*:

```
1> RESTORE $TAPE1, $SYSTEM.*.*, NOT $SYSTEM.SYSTEM.*
```

NOUNLOAD

The NOUNLOAD option causes the final tape to remain rewound and left online after the RESTORE process is completed. If neither NOUNLOAD or NOREWIND is specified the last tape is rewound and unloaded when the process is completed.

```
NOUNLOAD
```

Note. If you specify the NOUNLOAD option, other users can write to the tape after the RESTORE process is completed (if the tape is write enabled). Use of the NOREWIND option prevents another process from using the drive until the drive is manually reset. If you are backing up critical data, do not use the NOUNLOAD option if it is possible that the tape can be inadvertently overwritten. The use of labeled tapes with a TAPECATALOG DEFINE can also prevent other users from overwriting the tape.

NOUNLOAD is mutually exclusive with NOREWIND. If neither is specified the default is to unload the tape.

OPEN

The OPEN option restores files that were open during the BACKUP process. If you omit the OPEN option, files of this type are not restored.

```
OPEN
```

- For more information on backing up open and audited files, see AUDITED on page 7-7.
- For more information on restoring open files, see OPEN on page 3-29.

△ **Caution.** If audited files are backed up in the WRITE OPEN state, they could be corrupt. Restoring such files and using them in audited mode can cause TMF to bring down the whole volume. If audited files are restored, they should probably be used in nonaudited mode.

PAGELENGTH

The PAGELENGTH option specifies the number of lines that are generated per page of output from RESTORE.

```
PAGELENGTH number
```

number

is an integer in the range 20 through 100 that specifies the number of lines per page of output from RESTORE.

Guidelines

- When output from RESTORE is sent to a printer, a form feed or page eject is generated after the number of lines specified in the PAGELENGTH option.
- If the PAGELENGTH option is not specified, RESTORE defaults to 60 lines per page.

Example

- To direct output from RESTORE to a printer (and generates 50 lines per page):

```
1> RESTORE /OUT $S.#BOOK/ $TAPE, $MYVOL.*.*, PAGELENGTH 50
```

PART

The PART option specifies a new node or volume name for a secondary partition of a partitioned file. The PART option causes the secondary-partition name to be inserted in the file label of the primary partition (Enscribe files only).

```
PART ( sec-partition-num , [ [ \node.]$volume ]
       [ , pri-extent-size, [ sec-extent-size ] ] )
```

sec-partition-num

is an integer in the range 1 through 15, inclusive, that designates the secondary partition. This number was assigned to the partition when the partitioned file was created.

node

is the name of the node (system) where the secondary partition is to reside. If you omit the node name, RESTORE uses the original node name of the secondary partition.

volume

is the name of the volume where the secondary partition is to reside. If you omit the volume name, RESTORE uses the original volume name of the secondary partition.

pri-extent-size
sec-extent-size

defines the primary and secondary extent sizes, respectively. The default value is 1 page (2048 bytes). You can specify these values for *pri-extent-size* and *sec-extent-size*:

`0:512000000 [ PAGE[S] ]`

specifies the extent size in pages (2048-byte units). The minimum extent size is one page, so specifying 0 pages allocates one page (2048 bytes). PAGE is the default unit of measurement.

`0:2147483647 BYTE[S]`

specifies the extent size in bytes. RESTORE rounds up to the next full page. For example, if you specify 2047 bytes, RESTORE allocates one page; for 2049, it allocates two pages, and so on.

`0:2147483647 REC[S]`

specifies the extent size based on the current settings for record length, data-block length, index-block length, key-field lengths, and compression settings. RESTORE rounds up to the next full page.

Guidelines

- If you specify the PART option, you cannot specify the MAP NAMES option. The MAP NAMES option provides a complete mechanism for changing file names; the PART option has limitations.
- You cannot specify extent sizes with the PART option if your RESTORE command also contains the PARTOF or PARTONLY option.
- When you specify the PART option, the original name of the partition indicated by *sec-partition-num* is replaced by the new partition name you specify. The *sec-partition-num* parameter must already exist in the source file.
- If you omit the volume name, RESTORE uses the original volume name of the secondary partition.
- You can use the PART option to specify the destination partition extent sizes when converting files from one disk-process type to another.
- If you use the PART and PARTONLY ON options, RESTORE changes the name of the secondary partition in the file label of the primary partition. If you use the PART option without the PARTONLY ON option, RESTORE changes the secondary partition name in the primary partition file label and also changes the actual file name of the secondary partition.

△ **Caution.** The PART option affects all partitioned files in *restore-files*. More than one partition might have the specified *sec-partition-num*, so use the PART option carefully if you are restoring more than one partitioned file.

Examples

- To restore the secondary partition of the file $PUBS.BOOKS.SECT1 to the volume $SAVE:

```
1> RESTORE $TAPE, $PUBS.BOOKS.SECT1, PART (2, $SAVE)
```

- To restore the secondary partition of the file $DATA.GOOD.ONE to the volume $SAFE and sets the primary and secondary file extents to 2 and 3, respectively:

```
1> RESTORE $TAPE1, $DATA.GOOD.ONE, PART (3, $SAFE, 2, 3)
```

PARTOF

The PARTOF option restores only the partitions in *restore-files* whose primary partitions reside on a volume named in *volume-specification*. Both primary and

secondary partitions are restored. Nonpartitioned files defined by *restore-files* are not restored. (Partitioned Enscribe files only.)

```
PARTOF volume-specification
```

volume-specification

specifies where the primary partition must reside if the partition is to be restored. The *volume-specification* parameter is in one of these formats:

*

restores any partitions defined by the *restore-files*.

$volume
($volume , $volume ...)

restores only the partitions whose primary partition resides on that volume and that are defined by *restore-files*. Volume names used with the PARTOF option can include wild-card characters (? and *):

* (asterisk) matches from 0 through 8 characters in the position where it appears.

? (question mark) matches one character in the position where it appears.

For example, the volume name $SB?? matches all four-character volume names that begin with SB. The file name *CH? matches all file names that end with CH followed by any single character.

Guidelines

- You cannot use the PARTOF option if your RESTORE command also contains the EXT option.
- If you include the PARTOF option, the PARTONLY option of the BACKUP process must not have been specified when the tape was made. If it was, the RESTORE process terminates.
- If you include the PARTOF option, RESTORE must read the tape set (until it finds the primary partition) before it can restore individual secondary partitions.
- RESTORE lets both *restore-files* and the PARTOF option define more than one volume. Enter the RESTORE command carefully.
- The PARTOF option applies to Enscribe files, but you should use the PARTONLY ON option to achieve the same effect.

Examples

- To restore file partitions on the backup tape whose primary partitions reside on the volume $GEN:

```
1> RESTORE $TAPE, *.*.*, PARTOF $GEN
```

- To restore all file partitions on the volume $DATA:

```
1> RESTORE $TAPE2, $DATA.*.*, PARTOF *
```

PARTONLY

The PARTONLY option specifies if all of the partitions of a partitioned file (or table) are to be restored. This option applies to Enscribe and SQL files.

```
PARTONLY [ ON | OFF | PARTIAL ]
```

ON

specifies that only the partitions of a file explicitly named in *restore-files* are restored. ON is the default if you specify the PARTONLY option in the RESTORE command.

If the PARTONLY option was ON when the file was backed up, the PARTONLY ON option is automatically applied to the restore.

OFF

specifies that if the primary partition is specified in *restore-files*, then both the primary and secondary partitions are restored. OFF is the same as omitting the PARTONLY option from the RESTORE command.

With the PARTONLY OFF option, if a secondary partition is named in *restore-files* and the primary is not, the secondary partition is not restored.

PARTIAL

specifies that RESTORE function identically to PARTONLY ON until the verification phase. PARTONLY PARTIAL then skips the partition online operation, index linking, and SQL/MP object verification steps performed by PARTONLY ON.

For Enscribe files, PARTONLY PARTIAL is equivalent to PARTONLY ON.

Use this option when you have additional partitions to recover in subsequent RESTORE operations. All logically related SQL/MP objects are left offline in a non-functional state at the end of this RESTORE session. To successfully place table and index partitions online, link all indexes, and perform verification, you must perform a final RESTORE PARTONLY ON session. If the BACKUP tape contains partitioned or multiple indexes for a single table, HP strongly recommends that you perform the final PARTONLY ON operation specifying the primary partitions of the table and index.

PARTIAL prevents RESTORE from attempting time consuming I/O to all other partitions that cannot succeed until all partitions are present. It reduces the error and warning messages generated by missing partitions (some messages remain due to unsuccessful timestamp and drop operations).

After the RESTORE operations, use the FILEINFO *table-name*, DETAIL command on the primary and secondary partitions to verify that the index values in all file labels point to the primary partition of the index.

PARTONLY PARTIAL removes many resource conflicts that prevented concurrent RESTORE operations involving multiple partitions of the same object. The only remaining resource conflict involves multiple SQLCAT processes trying to access the same catalog files. Internal retry attempts cannot handle a large number of SQLCAT processes (this number is limited only by the number of tape drives available for RESTORE processes). To eliminate these conflicts, you can register partitions in multiple catalogs so that each parallel RESTORE PARTONLY PARTIAL qualified file set operates in a separate catalog. The final PARTONLY ON operation cannot be performed in parallel.

After a PARTONLY PARTIAL operation, NonStop SQL/MP objects might be in an unaudited state, owned by the RESTORE user ID, and secured OOOO. The final PARTONLY ON operation corrects these temporary states by putting all partitions online and linking all indexes.

Guidelines

- If you specify the PARTONLY ON or PARTIAL option, you cannot specify the EXT, INDEXES or IMPLICIT options.
- If you specify the PARTONLY ON or PARTIALoption, you can specify the MAP NAMES or CATALOG[S] option only if you also specify the KEEP option.
- If you want to restore partitions to resemble tables using the PARTONLY option, delete the partitions from the target locations using the GOAWAY command.

  The following example describes how to restore the partitions using the PARTONLY option.

  Consider that the source table (TABLE1) and target table (TABLE2) have three partitions each:

```
SQLCAT.TABLE1:
CATALOG $USER0.SQLCAT
KEY ( COLUMN 0, OFFSET 0, LENGTH 2, ASC )
PART ( 0, $USER0, 16 PAGES, 64 PAGES, MAXEXTENTS 160, FORMAT
1, 0 )
PART ( 1, $USER1, 16 PAGES, 64 PAGES, MAXEXTENTS 160, FORMAT
1, 10 )
PART ( 2, $USER2, 16 PAGES, 64 PAGES, MAXEXTENTS 160, FORMAT
1, 20 )

SQLCAT.TABLE2
CATALOG $USER1.SQLCAT
```

```
KEY ( COLUMN 0, OFFSET 0, LENGTH 2, ASC )
PART ( 0, $USER0, 16 PAGES, 64 PAGES, MAXEXTENTS 160, FORMAT
1, 0 )
PART ( 1, $USER1, 16 PAGES, 64 PAGES, MAXEXTENTS 160, FORMAT
1, 10 )
PART ( 2, $USER2, 16 PAGES, 64 PAGES, MAXEXTENTS 160, FORMAT
1, 20 )
```

The source and target tables have the same DDL (column and key definition, and partitioning keys) except the catalog.

Back up partition 1 from `TABLE1` using the PARTONLY option:

```
TACL>BACKUP
$TAPE,$USER1.SQLCAT.TABLE1,PARTONLY,AUDITED,LISTALL
```

Remove target partition 1 from `TABLE2`:

```
TACL>GOAWAY $USER1.SQLCAT.TABLE2
```

Restore partition 1 to the target partition 1 of `TABLE2`:

```
TACL>RESTORE,
$TAPE,*.*.*,AUDITED,LISTALL,PARTONLY,KEEP,MAP NAMES (
$USER0.SQLCAT.TABLE1 to
$USER0.SQLCAT.TABLE2, $USER1.SQLCAT.TABLE1 to
$USER1.SQLCAT.TABLE2,
$USER2.SQLCAT.TABLE1 to $USER2.SQLCAT.TABLE2 ), CATALOG
($USER1.SQLCAT for $USER0.*.TABLE2, $USER1.SQLCAT for
$USER1.*.TABLE2, $USER1.SQLCAT for $USER2.*.TABLE2)
```

In this example, the partition 1 of `TABLE1` is backed up and restored to the target partition 1 of `TABLE2`.

△ **Caution.** It is critical that the CATALOGS and MAP NAMES mapping lists are complete even when recovering an individual partition:

- If the CATALOGS option is specified, every associated catalog must be mapped.
- If the MAP NAMES option is specified, every partition must be mapped.

If the maps are incomplete or incorrect, severe catalog inconsistencies for the object will result. These inconsistencies are quite difficult to repair, so make sure to avoid them.

- For Enscribe files, PARTONLY PARTIAL is equivalent to PARTONLY ON.
- When converting Enscribe files from one disk-process type to another, the RESTORE process (with the PARTONLY ON or PARTIAL options) skips all relative files and any DP1 entry-sequenced files that do not have DP2 block sizes. For example, blocks of 1536, 2560, 3072, or 3584 bytes are not DP2-compatible. An error message provides the names of the skipped files. You must restore these files without the PARTONLY option.
- If you use the PART and PARTONLY ON or PARTIAL options, RESTORE changes the name of the secondary partition in the file label of the primary partition. If you use the PART option without the PARTONLY ON option, RESTORE changes the

secondary partition name in the primary partition file label (and also changes the actual file name of the secondary partition).

- After finishing the restores, to verify that the index values in both responses point to the primary partition, use the FILEINFO TAB, DETAIL command on both the primary and secondary volumes.

Examples

- To restore all of the files on the volume $BOOKS (from both the primary and secondary partitions on $BOOKS, but no partitions on other volumes):

```
1> RESTORE $TAPE1, $BOOKS.*.*, PARTONLY ON, LISTALL
```

- To restore all files on the subvolume MORE that are on the volume $WORDS:

```
1> RESTORE $TAPE2, $WORDS.MORE.*, PARTONLY OFF, LISTALL
```

  Primary partitions on $WORDS.MORE, and secondary partitions on other volumes whose primary partition is on $WORDS.MORE, are restored. Secondary partitions on $WORDS.MORE whose primary partitions reside elsewhere are not restored.

  Using this command without the PARTONLY OFF option gives the same results.

PHYSVOL

The PHYSVOL option places SMF files onto specific physical disks within SMF storage pools. This option only applies when the target volume is an SMF virtual disk process.

```
PHYSVOL ( sms-file-spec ON physical-volume-name
        [, sms-file-spec ON physical-volume-name ]...)
```

sms-file-spec

specifies a fully-qualified file name with an SMF virtual disk volume name component. Wild-card characters are supported.

physical-volume-name

specifies the name of a physical disk volume that is a member of an SMF storage pool.

Guidelines

- The ALTFILE, PART, and MAP NAME[S] RESTORE options are used to rename files, and the VOL RESTORE option can be used to change the node, volume, and subvolume of Enscribe files. Renaming a file can affect device residence, and each of these RESTORE options lets the volume name be changed.

  To restore all files on the BACKUP tape $TURBO to the SMF virtual disk $BRVC4:

```
RESTORE $TURBO, *.*.*, VOL $BRVC4, LISTALL
```

- Mapping from the PHYSVOL option is not performed until after the ALTFILE, PART, MAP NAME[S], or VOL RESTORE options can be applied. You cannot use the original file name; you must use the last *sms-file-specification* after applying the other mapping options.
- The PHYSVOL option is not required if the target is an SMF virtual volume. If the PHYSVOL option is not supplied, the virtual disk process (VDP) selects the location. This is preferred; use the PHYSVOL option sparingly.

Example

If the tape mounted on $TURBO is a BACKUP of the SMF virtual disk $BRVC3.*.*, to map all the files from virtual disk $BRVC3 to virtual disk $BRVC4, then ensure that DELME.FILE3 is created on physical volume $DATA of the $BRVC4 storage pool:

```
RESTORE $TURBO, *.*.*, MAP NAMES ($BRVC3.*.* TO $BRVC4.*.*),&
PHYSVOL ($BRVC4.DELME.FILE3 ON $DATA), LISTALL
```

Validation is not performed on the *physical-volume-name* value.

REBUILD

The REBUILD option specifies that the files in a group of related tape sets are defined by the file directories on the first tape set. This option only applies to Enscribe files; it has no effect on SQL files. Any SQL files named in the *restore-files* are restored as they would have been without the REBUILD option.

```
REBUILD
```

Guidelines

- Use the REBUILD option to restore tapes that were created using the PARTIAL BACKUP option.
- The tape sets that are used with the REBUILD option are usually created using the PARTIAL option of BACKUP. Start the RESTORE process with the most current tape set. If any files defined in the directory on the first tape set do not exist on that tape set, RESTORE prompts you to mount the preceding (older) tape set.

  This process of restoring files from preceding tape sets continues until all of the files are restored or until you terminate the RESTORE process by typing STOP when another tape is requested.
- Attempts to restore multivolume labeled tapes are simplified if an ADD DEFINE command is used. RESTORE prompts you with a mount message specifying the name of the first tape expected (as defined in the ADD DEFINE command).

  After the first tape is read, a mount message prompts you for the next specified tape. If the ADD DEFINE specifies no additional tapes, RESTORE prompts you to enter the volume ID of the next tape (if applicable).

- Any files on the volumes before the REBUILD process begins are retained (REBUILD implies KEEP).
- You cannot include the REBUILD option in a RESTORE command that converts files from DP2 to DP1 (or vice versa).

REGISTERONLY

The REGISTERONLY option directs RESTORE to restore a program without recompiling it. This option is valid only when the SQLCOMPILE option is ON.

```
REGISTERONLY [ ON | OFF ]
```

ON

directs the SQL compiler to register a program that was previously compiled with SQL in the specific catalog without compiling any SQL statements in the program. The SQL compiler marks the program file label as sensitive to SQL and valid to SQL. The program retains its existing execution plans.

You can use this option to install a program in a catalog after you have compiled the program with SQL and moved the program. Although the REGISTERONLY option requires you to run the compiler, this option is much faster than explicitly recompiling the entire program.

If the program was not previously compiled with SQL, the operation fails with SQL error 2115.

OFF

directs the SQL compiler to explicitly SQL compile the program and perform all SQL compiler functions.

Note. For more information, see the *SQL/MP Installation and Management Guide*.

REMOTEIOSIZE

The REMOTEIOSIZE option specifies the maximum size of each data block transferred between systems during a RESTORE process. This option is needed mainly for SQL files where the small default remote I/O transfer size can cause a performance problem.

```
REMOTEIOSIZE data-block-size
```

data-block-size

is the maximum number of 1024-byte increments (blocks) in each remote I/O transfer, specified as 2, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, or 52.

Guidelines

- REMOTEIOSIZE values above four (4096 bytes) cause RESTORE to use BulkIO logic.
- The default SQL remote I/O transfer size on the NonStop operating system is 4 KB (4096 bytes). Local I/O transfer size for SQL objects is 28 KB.
- Use of the BLOCKSIZE option when creating the BACKUP limits the available REMOTEIOSIZE values.

  For example, specifying a BLOCKSIZE of 28 in the BACKUP command limits the remote I/O read operation transfers to 28 KB blocks even if you specify a REMOTEIOSIZE of 52 in your RESTORE command. When restoring, REMOTEIOSIZE cannot increase the remote I/O write operation transfer size beyond what BACKUP put on the tape. Therefore, a REMOTEIOSIZE greater than the BLOCKSIZE is reduced internally. RESTORE does not display any messages when the BLOCKSIZE overrides the REMOTEIOSIZE in this manner.
- BLOCKSIZE values larger than 28 KB are only supported by the 3215 and 3216 controllers. All other transfers must be in the range of 2 through 28 KB. BLOCKSIZE can only be used if both the backup and restore are done on the same type of drive.
- The 52 KB limit is due to current tape BLOCKSIZE limitations. Expand can handle data transfers larger than 52 KB.
- Older Expand environments might not handle 52 KB transfer sizes. Different qualified file sets might involve systems with different Expand limitations, and the systems at RESTORE time might be different than those that created the BACKUP tape. For these reasons, individual limitations and automatically limited remote I/O size cannot be identified. If the REMOTEIOSIZE exceeds current hardware or software limitations, RESTORE returns this error:

  ```
  TABLENAME    *ERROR*  Guardian error: 21 (BulkWrite)
  ```

  You must repeat the operation using a smaller REMOTEIOSIZE.
- REMOTEIOSIZE does not affect any BACKUP tape contents (it only affects performance). Therefore, RESTORE has no way to know or display the value of REMOTEIOSIZE from when the BACKUP was made.
- The default ENSCRIBE I/O transfer size is equal to BLOCKSIZE, but you can use REMOTEIOSIZE to reduce it if you need to because of Expand limitations.

RENAME

The RENAME option sets the name of a restored disk volume, replacing the volume name that existed during the BACKUP process (volume mode only).

```
RENAME $volume
```

volume

is a valid volume name that is not being used for any other device or process in the system.

Guidelines

- For target volumes that are not removable, the RENAME option is ignored, and the disk retains its permanent name.
- SQL objects can be dependent on files in multiple volumes. Before renaming a volume, consider the dependencies of SQL objects that might be affected. For more information, see the *SQL/MP Installation and Management Guide*.
- You must use the RENAME option if you use the TARGET and VOLUMEMODE options to restore data to a disk that is different from the original source disk (and both the target disk and the source disk are on the same system). Otherwise, error 66 is returned when an SCF START command is issued to the restored disk. To prevent this problem:

```
RESTORE $tape,VOLUMEMODE,*,TARGET $lev-for-$data1,RENAME
   $data1
```

SCRATCHVOL

The SCRATCHVOL option specifies the volume where RESTORE creates the temporary files that are used in a DP1/DP2 conversion process for nonpartitioned files, key-sequenced files, or individual partitions of partitioned files.

```
SCRATCHVOL $volume
```

volume

is a volume where temporary files are created.

Guideline

You can use the SCRATCHVOL option if disk space is inadequate when converting key-sequenced files. If you omit the SCRATCHVOL option and file conversion is performed, RESTORE places the temporary files (with prefix ZZRA) on the current default volume and subvolume. If disk space is inadequate, RESTORE returns error 43 (unable to obtain disk space for file extent) for the file that was to be converted.

SQLCATALOGS

The SQLCATALOGS option specifies whether SQL catalogs are to be restored. The catalogs are not restored without this option.

```
SQLCATALOGS [ ON | OFF ]
```

ON

enables you to restore a table that is part of an SQL catalog. It is restored as an ordinary (noncatalog) table with a file code of 0. ON is the default if you specify the SQLCATALOGS option in the RESTORE command.

OFF

specifies that catalogs are not restored. OFF is the same as omitting the SQLCATALOGS option from the RESTORE command.

Guidelines

- If you try to restore a table to the same file name as an existing catalog table, an SQL error occurs.
- The SQLCATALOGS ON option should be used only to recover catalog data when TMF recovery is not possible. To restore SQL catalogs using RESTORE, consult your service provider. For more information on restoring SQL/MP catalogs, see the *SQL/MP Installation and Management Guide*. For more information on restoring SQL/MX catalogs, see the *SQL/MX Installation and Management Guide*.

SQLCOMPILE

The SQLCOMPILE option specifies whether an SQL object program file is recompiled automatically during a RESTORE process.

```
SQLCOMPILE [ ON | OFF ]
```

ON

specifies that when an SQL program file is restored, the program is automatically recompiled using the logical DEFINEs stored with the program description when it was previously compiled. ON is the default if you specify the SQLCOMPILE option in the RESTORE command.

OFF

specifies that the SQL program file is restored without being recompiled. OFF is the same as omitting the SQLCOMPILE option from the RESTORE command.

Guidelines

- All SQL programs that you are restoring, as well as all programs that use tables, indexes, or views, must be recompiled. This is performed automatically if you use SQLCOMPILE ON in your RESTORE command.
- The SQLCOMPILE ON option works only if the SQL program file on the source tape was compiled before it was backed up (and is therefore sensitive to SQL). For more information on compiling SQL program files, see the *SQL/MP Programming*

Manual for C, the *SQL/MP Programming Manual for COBOL* and the *SQL/MX Programming Manual for C and COBOL*.

- If you use the SQLCOMPILE ON option during a RESTORE process of a program file that is not sensitive to SQL, the file is restored but not as a program that is sensitive to SQL.

SQLTAPEPARTARRAY

The SQLRAPEPARTARRAY option makes RESTORE update all partitions of the target object with the partition array from the tape, and makes RESTORE recover data for the target partition.

```
SQLTAPEPARTARRAY
```

Guidelines

- Use this option when the SQL partition array has been corrupted online and a valid partition array exists on a backup tape. The online partition array is not saved and cannot be recovered unless it has been saved using BACKUP.
- No warning messages are displayed to indicate this option is in use.
- The default behavior of the PARTONLY option was changed in T9074ADA to use a current SQL partition array off of disk for recovery instead of the partition array on the backup tape. This is the correct action for most situations where the object's DDL has changed since the backup was made. This effort was referred to as Label Fixup. In rare cases, users have corrupted the SQL partition array on disk and needed to recover using a backup tape with the correct DDL. This option turns off the Label Fixup modifications and uses the partition array from tape.

START

The START option indicates where in the *restore-files* the RESTORE process is to begin. You must give a specific file name where the RESTORE process should start.

```
START [$volume.][subvolume.]file-id
```

volume.subvolume.file-id

indicates the volume, subvolume, and file where you want the RESTORE process to begin restoring files. RESTORE uses the current volume and subvolume if *volume* and *subvolume* are omitted.

Guidelines

- The file specified with the START option must be referenced in *restore-files*.

- Specify the file set like any other file set in a RESTORE command. You can use the asterisk wild card only to represent a whole subvolume name or file ID, such as:

  $*volume*.*subvolume*.*

  or

  $*volume*.*.*

  but not

  $*volume*.*subvolume*.AA*

- The START option works the same as the START qualifier in a qualified file set.

Example

To begin the RESTORE process at the file MAC2 on the current subvolume:

```
1> RESTORE $TAPE, *.*.*, START MAC2
```

TAPEDATE

The TAPEDATE option determines the date used for the last modification and last open timestamps of the files that are being restored.

```
TAPEDATE [ ON | OFF ]
```

ON

sets the last modification and last open timestamps of the restored files to the values they had when the files were backed up. ON is the default for this option.

OFF

changes the last modification and last open timestamps of the restored files to the time of the RESTORE process.

Examples

- To display the timestamps for a file backed up on September 10, 2001 and restored on October 12, 2001 with the TAPEDATE ON option:

```
$BOOKS1.SHARONF.NOVEL
DATA MODIF:  10 Sep 2001, 13:13
CREATION DATE:  12 Oct 2001, 16:10
LAST OPEN:  10 Sep 2001, 13:13
```

- To display the timestamps for a file restored on October 12, 2001 with the TAPEDATE OFF option:

```
$BOOKS1.SHARONF.NOVEL
DATA MODIF:  12 Oct 2001, 16:10
```

```
CREATION DATE:  12 Oct 2001, 16:10
LAST OPEN:  12 Oct 2001, 16:10
```

The creation date for the file is changed to the restore date whether or not the TAPEDATE option is used.

TARGET

The TARGET option specifies the disk device where the data is to be restored. This option is for use in volume mode only.

```
TARGET $ldev [ -P | -M ]
```

ldev

specifies the disk (by logical device number) where the disk image from tape is to be restored.

-P
-M

specifies whether the primary (-P) or mirror (-M) half of the mirrored pair is to be restored. If neither -P or -M is specified, the disk image is restored to both halves of the logical mirrored pair.

Guidelines

- If the TARGET option is not specified, the disk image is restored to the disk volume that is named on the backup tape.
- SQL objects can be dependent on files in multiple volumes. Before restoring files to a different volume, consider the dependencies of SQL objects that might be affected. For more information, see the *SQL/MP Installation and Management Guide*.
- If you use the TARGET and VOLUMEMODE options to restore data to a disk that is different from the original source disk, and both the target disk and the source disk are on the same system, you must use the RENAME option. Otherwise, error 66 is returned when an SCF START command is issued to the restored disk. To prevent the problem, use the RENAME option:

```
RESTORE $tape,VOLUMEMODE,*,TARGET $lev-for-$data1,RENAME
   $data1
```

TURNOFFAUDIT

The TURNOFFAUDIT option specifies that files audited by TMF when they were backed up are not to be audited when they are restored. You must use the AUDITED option with the TURNOFFAUDIT option.

```
TURNOFFAUDIT
```

For more information about restoring audited files, see AUDITED on page 7-7.

VERIFY

The VERIFY option verifies the data integrity of the files on disk.

```
VERIFY
```

Guideline

Using a RESTORE process with the VERIFY option takes longer to complete because RESTORE checks file labels and data on the disk after it writes the tapes to disk.

VERIFYTAPE

The VERIFYTAPE option verifies the volume label, file label, and data records of the files on tape.

```
VERIFYTAPE
```

Guideline

You must use the LISTONLY option with the VERIFYTAPE option.

VOL

The VOL option restores files to a new disk volume name. This option applies only to Enscribe files.

```
VOL [\node.][$new-vol.]new-subvol |

    [\node.]$new-vol[.new-subvol]
```

node

is the new node (system) name.

new-vol

is the new volume name.

new-subvol

is the new subvolume name.

Guidelines

- If you specify the VOL option, you cannot specify the MAP NAMES option.
- The VOL option has limitations in handling secondary partitions and alternate-key files. For more information, see MAP NAMES on page 7-18.
- Specify the name of the remote node for *node* to restore files to a remote node.
- Specify a destination volume whose disk-process type is different from the disk-process format of the files on tape to convert files using the VOL option.
- If you specify *new-subvol*, you can restore files to only one subvolume.

△ **Caution.** If you specify a new volume or subvolume name with the VOL option and the same file name is given to two or more files that are being restored, RESTORE replaces the file each time it encounters the file name. Thus, only the last occurrence of the duplicate file name is restored. For more information about duplicate file names, see PARTONLY on page 7-29.

VOLUMEMODE

The VOLUMEMODE option directs RESTORE to operate in volume mode instead of file mode.

```
VOLUMEMODE, { $volume | * }
```

volume

specifies the disk volume that is to be restored from tape. When specified, *volume* must match the disk volume name that is on the tape.

*

specifies that any disk volume on the tape is to be restored. The asterisk matches any disk volume name on the tape.

Guidelines

- The disk volume you specify to be restored must be on the current node (system). That is, you cannot do a remote volume-mode restore.
- Any volume-mode BACKUP/RESTORE request to a SMF virtual disk results in this error message: “Volume Mode BACKUP/RESTORE is incompatible with SMF virtual disks, use File Mode.”

- If you try a volume-mode backup from a disk with a capacity greater than 2 GB, you cannot restore it to an operating system version prior to D30, and this message is generated:

  ```
  The source disk has a capacity of more than 2 GB, which
  cannot be restored to a pre-D30 system.
  ```

- For more information, see Restoring in Volume Mode on page 7-62.

Completion Information

RESTORE returns a completion code to the process that started it. The completion code indicates how successful the RESTORE process ran. Its primary purpose is to support batch processing by allowing conditional execution of subsequent processes based on the success of the RESTORE process. If TACL started the RESTORE process, the completion code is stored in a TACL variable (:_COMPLETION), where you can examine it using TACL functions. TACL also displays the completion codes at the home terminal, unless the RESTORE process completes normally without any warnings or errors. For more information about completion codes and batch processing, see the *TACL Programming Guide*.

In addition to a completion code, RESTORE also returns a subsystem identifier (TANDEM.75.*version*), where *version* is the release number (for example, C30). The subsystem identifier is also stored in the TACL variable (:_COMPLETION).

Table 7-2. Completion Codes Returned by RESTORE and the Operating System

Code	Description
0	A normal voluntary termination. The process completed normally with no errors or warnings.
1	A normal, voluntary termination with warnings. All the files were copied, but warnings occurred on one or more files. (Warnings that are not file-specific do not result in this code.)
2	An abnormal, voluntary termination with errors. RESTORE could not copy one or more files because of errors on the tape or disk. RESTORE skips the files that cannot be restored.
3	A premature, voluntary termination with fatal errors. RESTORE terminated itself because of a fatal error or an invalid user command. The files might not all be restored.
4	The RESTORE process never got started. This completion code is generated by the TACL process executing RESTORE. TACL returns a process creation error in the *terminationinfo* field.
5	The RESTORE process calls ABEND on itself. This code is the default completion code for the ABEND procedure.

RESTORE Guidelines and Examples

Note. The current node in a RESTORE process is where the utility is running. HP recommends that you run RESTORE on the same node where the files were backed up.

Viewing the Contents of a Tape

To display the contents of a tape without physically restoring files to disk:

- To verify the tape on drive $TAPE1 and list the files without writing the tape to disk:

  ```
  1> RESTORE $TAPE1,*.*.*, VERIFYTAPE, LISTONLY
  ```

  RESTORE displays a listing at the home terminal similar to:

```
File RESTORE Program - T9074D46 - (07SEP98)          System: \GLOBE
Copyright Tandem Computers Incorporated 1981-1998
Drives: $TAPE1
System: G06  Tape Version: 3
Backup options: NO AUDITED, BLOCKSIZE 8, NO IGNORE, NO OPEN, NO PARTONLY
Restore (list only) time: 19Dec1998 16:13 Backup time: 30Jul1998 18:04 Page: 1

tape: 1  OBC Code       EOF        Last modif    Owner RWEP  Type Rec Bl

 BMEMO         101      2126   05Nov1998 14:56   8,44  OOOO
 CMEMO         101     36442   19Nov1998 18:48   8,44  OOOO
 INDEX         101     61616   19Nov1998 19:02   8,12  CUUU
 PREFACE       101      9262   19Nov1998 19:05   8,12  CUUU
 SEC1          101     26374   19Nov1998 19:06   8,12  CUUU
 SEC2          101    214030   19Nov1998 19:09   8,12  CUUU
 SEC3FIGS      101      6580   19Nov1998 19:22   8,12  CUUU
 SEC4          101    338605   23Nov1998 11:13   8,12  CUUU
 SEC5          101    191016   23Nov1998 14:17   8,12  CUUU
 SEC5FIGS      101      3410   18Nov1998 18:27   8,12  CUUU
 SEC6          101     55418   19Nov1998  9:55   8,12  CUUU
 SEC7          101     75420   19Nov1998 18:34   8,12  CUUU
 SEC8          101     54444   18Nov1998 16:44   8,12  CUUU
 SYNTAX        101      7790   05Nov1998 18:38   8,12  CUUU
 TITLE         101      3348   19Nov1998 15:06   8,12  CUUU

Summary Information
Files restored = 16  Files not restored = 0
```

For a description of the fields in the LISTONLY output, see LISTALL on page 3-22.

- To list the files at the home terminal in volume $DATA from the tape mounted on $TAPE2 without verifying them or writing them to disk:

  ```
  1> RESTORE $TAPE2, $DATA.*.*, LISTONLY
  ```

- To list the files in volume $DATA from the tape mounted on $TAPE3 at the terminal $MYTERM, and leave the final reel in the tape set online at the end of the listing, so the operator can start a RESTORE process without remounting the tape:

  ```
  1> RESTORE /OUT $MYTERM / $TAPE3, $DATA.*.*, NOUNLOAD &
  1> &LISTONLY
  ```

Restoring Enscribe Files

For any RESTORE process that involves Enscribe files:

- If an object file being restored has the PROGID attribute set, the file retains the PROGID setting only if the owner restores the file, and the restore program is running under the super ID (255, 255). The object file being restored is licensed if the program is running under the super ID (255, 255).
- If an object file being restored is a licensed file, the file loses its license if a user other than the super ID restores the file.
- The report listings from RESTORE and BACKUP display the significant options that were used when the utility was run.

  For example, use of the DETAIL option is not reported because that option has no significant impact on a BACKUP or RESTORE process. In general, the options in the RESTORE command must match those used in the original BACKUP command. These options are particularly significant:

  - The PARTONLY option must be the same for a BACKUP and RESTORE process if you are restoring partitioned files. That is, if the PARTONLY ON option does not appear in the listing, you cannot include the PARTONLY ON option in your RESTORE command.
  - The AUDITED and OPEN BACKUP options indicate whether the tape might contain audited files or files that were open at the time of the BACKUP process.
  - Backup tapes created using the PARTIAL BACKUP option are restored with the REBUILD option. Since REBUILD implies KEEP, any files on the disk before the REBUILD process begins are retained.

  If the RESTORE process does not complete normally, the file is marked corrupt and should be purged. The file system rejects normal OPENs of corrupt files.

- If the state of FLTrustFlags for an object file being restored is ME or SHARED, FLTrustFlags for the file is set to OFF if a user other than the super ID restores the file.

  Note. The FLTrustFlags attribute is supported only on systems running H-series RVUs or J-series RVUs.

Examples

- To restore all files on the backup tape except those on the volume $SYSTEM:

```
1> RESTORE $TAPE2,*.*.* EXCLUDE $SYSTEM.*.*, KEEP, &
1> &LISTALL
```

  The KEEP option preserves the disk file if it has the same name as a file on the backup tape. That is, the file is not restored from the tape.

- To restore all files on tape that have the volume and subvolume name $STORE2.ACCTSRCV to the subvolume ACCTSRCV on the volume $FINANCE:

```
1> RESTORE $TAPE, $STORE2.ACCTSRCV.*, &
1> &VOL $FINANCE.ACCTSRCV, LISTALL
```

  Any file on $FINANCE.ACCTSRCV that has the same name as a file on the backup tape is purged and the backup copy is restored. The names of all specified files are listed on the default output file or device:

- To have RESTORE begin restoring files when it encounters the file DATA.MYFILE, without restoring files that precede MYFILE in subvolume DATA, and leaving the tape loaded and online after the RESTORE operation:

```
1> RESTORE $TAPE3,*.* START DATA.MYFILE, LISTALL,  &
1> &NOUNLOAD
```

  Subvolumes that precede DATA are not restored, but subvolumes that follow DATA are restored.

- To restore the files using the modification and last open timestamps of the restore time, use the TAPEDATE OFF option:

```
1> RESTORE $TAPE1, *.*.*, LISTALL, TAPEDATE OFF
```

- To restore all files and file partitions for the volume $STORE1. It does not restore partitions on other volumes, for a BACKUP tape created with the PARTONLY ON option:

```
1> VOLUME $STORE1
2> RESTORE $TAPE, *.*, PARTONLY ON, LISTALL
```

- To restore all file partitions on $SYSTEM that have primary partitions on the volume $STORE1, for a BACKUP tape created without the PARTONLY option:

```
1> VOLUME $SYSTEM
2> RESTORE $TAPE3, *.*, PARTOF $STORE1, LISTALL
```

- To restore all files whose owner is SOFTWARE.BOGEY:

```
1> RESTORE $TAPE1, *.*.* WHERE OWNER =SOFTWARE.BOGEY
```

- To restore the files and preserve the NOPURGEUNTIL date of the Enscribe files:

```
1> RESTORE $TAPE1, $DEV.*.*, LISTALL, NOPURGEUNTIL, VOL
$DATA, NOUNLOAD
```

Restoring SQL Files

These types of SQL files can be explicitly named in *restore-files*, or restored using the FROM CATALOG qualifier: base tables, catalog tables, indexes on base tables and catalog tables, and SQL shorthand views.

△ **Caution.** BACKUP is normally used as a secondary recovery method for SQL objects; the TMF recovery process is used as the primary method. An SQL object recovered with RESTORE might be inconsistent with the current catalog description of the object.

For more information about restoring SQL objects, including guidelines for restoring and moving individual objects and entire databases, and renaming objects, see the *SQL/MP Installation and Management Guide* and *SQL/MX Installation and Mangement Guide*.

- When you use PARTONLY OFF and INDEXES IMPLICIT:
  - When a base table is restored to disk, any SQL protection views defined on the table are restored automatically. Any SQL shorthand views that depend on the table are not restored unless they are explicitly named in the *restore-files*.
  - When a base table is restored, any constraints associated with the table are restored automatically.
  - When a base table, index, or SQL shorthand view is restored, any comments associated with these objects are restored automatically.

    Note. SQL tables can be restored only if the program file from RESTORE is licensed. If you use DSM/SCM to install RESTORE, licensing is automatic. If you do not use DSM/SCM, to license the RESTORE program file use the FUP LICENSE command.

  - An SQL protection view is automatically restored when its base table is named in *restore-files*. An SQL protection view cannot be restored independent of the base table on which the protection view is dependent.
- BACKUP and RESTORE do not work with remote SQL tables if SQL is not installed on the local node.
- All SQL programs, as well as all programs that use tables, indexes, or views, that you restore must be recompiled. To automatically recompile all SQL programs

when they are restored, use the SQLCOMPILE ON option in your RESTORE command.

△ **Caution.** It is critical that the CATALOGS and MAP NAMES mapping lists are complete even when recovering an individual partition:

- If the CATALOGS option is specified, every associated catalog must be mapped.
- If the MAP NAMES option is specified, every partition must be mapped.

If the maps are incomplete or incorrect, severe catalog inconsistencies for the object will result. These inconsistencies are quite difficult to repair, so make sure to avoid them.

- To restore an SQL object to a different location, use the MAP NAMES option. If the location is on a different system, you must specify both the old and new system names. For example:

  ```
  MAP NAMES (\oldsys.subvol.filename to
  \newsys.subvol.filename)
  ```

- RESTORE cannot directly recover a catalog. TMF recovery methods protect SQL catalogs. Catalog tables are audited so that they can be archived with TMF and recovered with either TMF autorollback or rollforward.
- If the SQL catalog (where SQL objects are registered) does not exist, use the AUTOCREATECATALOG ON option in your RESTORE command. An SQL catalog is created, and RESTORE inserts appropriate entries with the first instance of an SQL object.
- To register SQL objects in a different SQL catalog during a restore than they were originally registered in, use the CATALOGS option. If the different catalog is on a different system, use a fileset that describes the destination fileset, not the original.
- SQL catalogs backed up with the SQLCATALOGS ON option can be restored as normal SQL tables with a file code of 0.
- RESTORE purges the existing object on disk before it begins writing files from the backup tape. If an error occurs during the RESTORE process and the INDEXES IMPLICIT option was used, RESTORE purges the restored SQL components. If the INDEXES EXPLICIT or PARTONLY ON options were used, RESTORE purges the restored partition or index and leaves the base table invalid. If an error occurs when RESTORE attempts to validate objects towards the end of the operation, the restored objects are left invalid.

  In these situations, you might have to fix inconsistencies in SQL tables:

  - All of the components were not restored.
  - The security or owner ID is mismatched. The components (partitions) of a base table must have the same security and owner.
  - Partitions are inconsistent with the indexes.
  - Indexes are inconsistent with the underlying table.

- Constraints are mismatched with the underlying table.
- Physical attributes of partitions are mismatched.

- The state of restored SQL files depends on the particular RESTORE options that were used during the BACKUP and RESTORE processes. The default options of BACKUP and RESTORE are:

  ```
  PARTONLY OFF, INDEXES IMPLICIT
  ```

  If only the default options are used, these statements are true:

  - Sets of logically related objects such as partitions and indexes are backed up and restored together.
  - The default RESTORE process purges any existing objects with identical names.
  - At the completion of the RESTORE process, the restored objects are in a valid state. That is, the objects are now accessible for normal use such as running SQL queries, and so on.

- When you use the PARTONLY option with SQL files:
  - If the existing disk file is an SQL table partition and the PARTONLY option is ON, only the partition is purged. The other partitions and the indexes on the table are preserved. If the PARTONLY option is OFF, an SQL DROP TABLE command is applied to the table, which causes the entire partitioned table and all indexes defined on the table to be dropped. Any SQL program that depends on the table is marked invalid.
  - If the existing disk file is an SQL index partition and the PARTONLY option is ON, only that partition of the index is purged. If the PARTONLY option is OFF, an SQL DROP INDEX command is applied to the index, which causes all partitions of the index to be dropped. Any SQL program that depends on the index is marked invalid.
  - If the existing disk file is an SQL view definition, an SQL DROP VIEW command is applied to the view whether the PARTONLY option is ON or OFF. For more information about the effects of the SQL DROP TABLE, DROP INDEX, and DROP VIEW commands, see the *SQL/MP Reference Manual*.
  - RESTORE PARTONLY ON operations on SQL objects involve timestamp updates and partition array synchronization to partitions not included in the *restore-files* parameter. All logically related table and index partitions, views, collations, and SQL programs are affected. Error messages appear when required elements are not yet present or are unavailable for update. Temporary, corrupt file labels might be created for missing table and index partitions. Some severe situations, such as missing nodes, can result in partitions being left offline.
  - The final RESTORE PARTONLY ON session performs steps that cannot complete successfully until all index and table partitions are recovered. These

include placing partitions online, linking indexes to the table partitions, and verification. Ideally, the primary partition of the base table should be the last piece recovered to disk though, through special handling, RESTORE can handle most situations where this is not the case.

Examples: Restoring SQL Files

The first four examples correspond to the Example on page 3-55:

- To restore a shorthand view, the view must have been backed up explicitly. In this example, the shorthand view is restored to a different volume from the one on which it was backed up; its underlying table and index were previously moved to a different volume from the one on which they were backed up. The MAP NAMES option moves the files and resets the pointers to maintain consistency in the SQL object.

```
1> RESTORE $TAPE, $A.A.SVIEW, AUDITED, LISTALL &
1> &MAP NAMES ($A.A.* TO $D.D.*,              &
1> &$B.B.INDX TO $E.E.INDX),                  &
1> &CATALOGS ($C.CAT FOR $E.*.*, $D.CAT FOR $D.*.*)
```

- To explicitly restore index table $B.B.INDX:

```
2> RESTORE $TAPE1, $B.B.INDX, AUDITED, LISTALL &
2> &INDEXES EXPLICIT
```

- To explicitly restore the secondary partition $D.D.PART:

```
3> RESTORE $TAPE2, $D.D.PART, LISTALL, AUDITED, &
3> &PARTONLY ON
```

- To perform the RESTORE process, when disk volumes $A and $B can contain both Enscribe and SQL files:

```
4> RESTORE $TAPE, ($A.*.*, $B.*.*), &
4> &CATALOGS ($A.CAT FOR $A.*.* ,     &
4> &$B.CAT FOR $B.*.*), AUDITED, LISTALL
```

  This command has these results:

  - The tape files are copied to disk if they have the volume name $A or $B. All partitions, protection views, and indexes associated with files that have volume name $A or $B are restored.
  - The newly copied disk files are given the same names as the ones on the tape. All the new disk files have the volume name $A or $B.
  - Existing disk files whose names match the file names generated by RESTORE are purged.
  - Any SQL object that is restored to $A.*.* will be registered in the catalog $A.CAT. Similarly, any SQL object that is restored to $B.*.* will be registered in the catalog $B.CAT.
  - An output listing is displayed on the terminal. This listing includes the names of all the disk files that were successfully restored, the names of disk files that were not restored because of errors, and the error messages for the disk files that were not generated.

- To include the MAP NAMES option in the previous example and restore the files to another volume:

```
1> RESTORE $TAPE, ($A.*.*, $B.*.*),&
1> &MAP NAMES ($A.*.* TO $C.*.*, $B.*.* TO $D.*.*),&
1> &CATALOGS ($C.CAT FOR $C.*, $D.CAT for $D.*),&
1> &LISTALL, AUDITED
```

  This command produces these results:

  - The tape files are copied to disk if they have the volume name $A or $B. All partitions, protection views, and indexes associated with files that have the volume name $A or $B are restored.
  - The newly generated disk files are all given the volume name $C or $D. Their subvolume names and file IDs are the same as the tape files from which they were generated.
  - The newly generated SQL files are described in catalog $C.CAT and $D.CAT.
  - Existing disk files on volumes $A and $B are preserved because the newly generated disk files all have the volume name $C or $D.
  - An output listing is displayed on the terminal.

- To restore a partitioned file that was backed up on system \A to system \B and registers the file in catalog $B.CAT:

```
1> RESTORE $TAPE, \A.$A.*.*
1> MAP NAMES (\A.$A.*.* TO \B.$B.* *, &
1> $A.*.*   TO $B.*.*),&
1> CATALOGS ($B.CAT FOR $B.*.*) &
1> AUDITED, LISTALL
```

Restoring Existing Disk Files

When restoring files that already exist on disk, if the name of a file to be restored has the same name as a file that exists on disk, the action taken by RESTORE depends on the file type and the options that you specify in your RESTORE command:

- If you specify the KEEP option, the file that exists on disk is preserved and the file on tape is not restored.
- If you specify the PARTONLY ON option, only the existing disk files with the same names as the files explicitly named in the *restore-files* parameter of the RESTORE command are purged. Specifying the PARTONLY OFF option purges existing disk files with the same names as the files on tape, and all objects logically related to the existing disk files (such as secondary partitions and indexes).
- If the existing disk file is a primary partition and the PARTONLY option is ON, only that partition is purged. If PARTONLY is OFF, the entire partitioned file is purged.

Restoring Using Name Mapping

You can use the MAP NAMES and CATALOG(S) options to:

- Move interrelated files such as SQL objects from one volume or node to another
- Within limits, transfer files backed up on one network to another

To understand how the MAP NAMES option works when you restore files to a different node or network, you must know how BACKUP and RESTORE handle file names.

When you restore these multinode files, RESTORE searches the network for a node that matches the backup node information. If you are restoring to the node on which the files were backed up or to another node on the same network, you need not be concerned about file names or destinations. RESTORE restores each file to its original location. However, if you must restore files to a different network, or if you must restore SQL objects to a new location, you might need to perform name mapping.

The CATALOG[S] option is necessary with the MAP NAMES option if a different catalog is to be used during RESTORE. A complete mapping scheme eliminates errors if AUTOCREATECATALOG is OFF, and unexpected registration locations if it is ON.

△ **Caution.** It is critical that the CATALOGS and MAP NAMES mapping lists are complete even when recovering an individual partition:

- If the CATALOGS option is specified, every associated catalog must be mapped.
- If the MAP NAMES option is specified, every partition must be mapped.

If the maps are incomplete or incorrect, severe catalog inconsistencies for the object will result. These inconsistencies are quite difficult to repair, so make sure to avoid them.

Restoring Files From Multiple Systems

If you try to restore a file set that originally existed on multiple system nodes, RESTORE searches the current network for a system that matches the file node of any remote files on the BACKUP tape.

If you are restoring to the system on which the files were backed up (or to another system on the same network), you do not need to be concerned about the file names or destination for remote files. RESTORE restores the remote files back to their original locations. Because the local files do not contain the file node information, the local files will be restored on the system that is running RESTORE.

Restoring Files to a Different Network

If you run RESTORE on a system that is not connected to the network where the BACKUP session was run, several outcomes are possible:

- If all the files on the BACKUP tape are in the local format, the files are restored to the local system. For more information on backup formats, see BACKUP Guidelines and Examples on page 3-45.

- If any files on the BACKUP tape are in the remote format, RESTORE attempts to restore the files to the system that has the same system node number as the remote file. For example:

  1. Run RESTORE with the LISTONLY option:

     ```
     RESTORE $TAPE, ( *.*.* ), LISTONLY
     ```

     This lets you determine how node numbers and node names are correlated on the destination network.

     If the source node number exists in the destination network, the output from the LISTONLY option displays the matching node name in the destination network. For example, the files are listed as:

     ```
     $VOL.SUBVOL
     TInd1
     \B.$VOL.SUBVOL
     Table1
     ```

     If the node number does not exist on the destination network, the node name is replaced with "\??" in the output from the LISTONLY option. For example, the files are listed as:

     ```
     $VOL.SUBVOL
     TInd1
     \??.$VOL.SUBVOL
     Table1
     ```

  2. After running RESTORE with the LISTONLY option once, run it again, but use the appropriate QFS in RESTORE command along with a MAP NAMES option to specify the destination node names.

     For example, you can restore the files from Step 1 with the command:

     ```
     RESTORE $TAPE, ( \B.$VOL.SUBVOL.TABLE1 ),&
     MAP NAMES ( \B.$VOL.SUBVOL.* TO $NEW.SUB.* )
     ```

     You cannot restore the files of a remote file set if the system node is offline. RESTORE cannot resolve the system node number of the remote files.

Qualified File Sets

If you attempt to restore files from a BACKUP tape, you need to remember the file set syntax. A file set is specified in the form:

```
[[[ \node.]$volume.]subvolume.]file-id
```

- A file set can contain wild-card characters in the volume, subvolume, and file-id fields. A file set with the asterisk in all three positions (*.*.*) instructs the RESTORE to restore all files on the tape, including both the local and remote files.
- A file set with a dollar sign in the volume location ($*.*.*) instructs RESTORE to restore all of the local files on the tape. Any files stored on tape in the remote node format will not be restored or listed using this format. For example, if you have a

tape with a remote file-set from \B, the QFS “$*.*.*” will not be restored because the QFS of “$VOL.*.*” designates a file set that is local to the RESTORE system:

```
RESTORE $TAPE, ( $*.*.* )
```

You receive this message:

```
\A
Files not found - Error 2013
```

If this situation occurs, use “*.*.*” with an appropriate MAP NAMES option to restore all of the files to the local node:

```
RESTORE $TAPE, ( *.*.* ), MAP NAMES ( *.*.* TO $NEW.SUB.*
)
\A.$NEW.SUB
Table1
TInd1
```

You could also use the remote node name for the file-set list (if you know it). Use the node name in the file set (\ node.$*.*.*) to instruct RESTORE to restore all of the files from that node:

```
RESTORE $TAPE, ( \B.$VOL.*.* ),
                MAP NAMES ( \B.$VOL.*.* TO $NEW.SUB.* )
\A.$NEW.SUB
Table1
```

Moving SQL Objects

You can use the MAP NAMES option to move SQL objects to another volume or node. However, be careful to define the MAP NAMES and CATALOGS parameters correctly for the dependent tables, indexes, views, and programs. An incorrect or incomplete mapping scheme can leave the objects invalid or cause the RESTORE process to fail. The MAP NAMES mapping list must be complete even for recovering an individual partition. Every partition contains a complete list of all other partitions and must be completely mapped for the SQL object to function.

To restore backup files $A.*.* to a different volume, $C.*.*, and register the files in the catalog $C.CAT:

```
1> RESTORE $TAPE, $A.*.*, &
1> &MAP NAMES ($A.*.* TO $C.*.*) &
1> &CATALOGS ($C.CAT FOR C.*.*)
```

The CATALOGS option does not use the original file names; instead, it uses the new file names specified by the MAP NAMES option because the MAP NAMES option is processed before the CATALOGS option.

The MAP NAMES option also applies to the names and definitions of restored SQL views. Both the SQL view name and the SQL table name in the SQL view definition are changed according to the rules regarding the MAP NAMES option. An SQL protection view and its related base table must reside on the same subvolume and catalog; a shorthand view can reside in a separate subvolume from its base table and catalog.

When you use the MAP NAMES option to restore a shorthand view to a volume that does not contain the related base table, include the base table name in the MAP NAMES syntax. This method ensures that the shorthand view can find the base table. For example, to move the shorthand view $A.SH.VIEW, which refers to the base table $B.VIEW.TAB, to volume $C:

```
RESTORE $TAPE, $A.SH.VIEW, &
&MAP NAMES ($A.SH.VIEW TO $C.SH.VIEW,
            ($B.VIEW.TAB TO $C.VIEW.TAB)
```

This example moves the secondary partitions of a file from multiple nodes onto one node. Data was backed up using this command:

```
BACKUP $TAPE, ($B.B.B, $D.B.B) WHERE SQL, AUDITED
```

where:

- \A.$B.B.B is the primary partition of an SQL table that has secondary partitions on \C.$C.B.B and \D.$D.B.B, and...
- \A.$D.B.B is another SQL object unrelated to \A.$B.B.B

The file names are stored on tape in the form:

```
$B.B.B , \C.$C.B.B , \D.$D B.B , $D.B.B
```

To restore these SQL objects to different volumes on node \A, the command might be:

```
RESTORE $TAPE, *.*.* WHERE SQL,  &
&MAP NAMES ($B.*.* TO $F.*.*,
            \C.$C.*.* TO $F1.*.*,
            \D.$D.*.* TO $F2.*.*,
            $D.*.* TO $F3.*.*), AUDITED
```

The file names for secondary partitions \C.$C.B.B and \D.$D.B.B are stored in remote format, so that is how they must be identified in the MAP NAMES option. Instead of restoring these partitions to their source nodes, the MAP NAMES option causes them to be restored to different volumes on \A. The MAP NAMES option also directs the files \D.$D.B.B and \A.$D.B.B to different locations on node \A to avoid a file-name conflict.

Restoring or Converting DP1 and DP2 Files

During a file-mode RESTORE process (not volume mode), you can use RESTORE to convert files on tape from the DP1 disk-process format to DP2. RESTORE automatically converts files to match the disk-process type of the destination volume. You can also restore some files to DP2 volumes and other files to DP1 volumes in a system that has volumes of both disk-process types.

Files are restored to the volume and subvolume that match their file names on tape. At backup time, the files on tape can be given different volume and subvolume names with BACKUP's VOL option. Also, to specify a different destination volume and subvolume for the restored files, include the VOL or MAP NAMES options in the RESTORE command.

You do not specify disk conversion during the RESTORE process; RESTORE automatically converts the files if needed:

- When RESTORE converts files, it creates temporary disk files on the current subvolume which, in the destination-file format, have names that begin with ZZRA. If you terminate the RESTORE process early, the current subvolume might contain some of these files. To regain disk space, purge any remaining ZZRA files. If disk space is inadequate, use the SCRATCHVOL option.
- You can use RESTORE to convert key-sequenced files. For DP2 files, index and data blocks are the same size. When you convert a key-sequenced file from DP1 to DP2, RESTORE changes the index-block size if it is not equal to the data-block size. RESTORE does not change the block size for entry-sequenced or relative files.
- If RESTORE converts an entry-sequenced DP1 file that does not have a DP2-compatible block size (for example, 1536, 2560, 3072, or 3584 bytes), the block size is rounded up to a DP2-compatible size (2048 or 4096 bytes). This change causes record addresses to change in the file. A warning message says that you must perform a FUP LOADALTFILE on all the alternate-key files of this primary file.
- Key-sequenced files sometimes become larger during conversion. To avoid file-system error 45, use the EXT option to increase the destination file size.

Restoring From Labeled-Tapes

You can perform labeled-tape BACKUP and RESTORE processes using either file mode or volume mode. BACKUP, RESTORE, and BACKCOPY send a labeled-tape request through a CLASS TAPE DEFINE to the labeled-tape server process ($ZSVR). It passes messages about labeled-tape operations to the operator console ($0) and the MEDIACOM console (if one exists).

Create the DEFINE the same way you create DEFINEs for ANSI and IBM labeled tapes. For information on DEFINEs or labeled tape processing on NonStop systems, including instructions on using DEFINEs and how to use the CLASS TAPE DEFINE for labeled tape processing, see the *Guardian User's Guide*.

Unlike TAPE DEFINEs for ANSI and IBM tapes, you can use only a subset of the CLASS TAPE attributes for a RESTORE process. For a summary of supported attributes, see Table 7-3 on page 7-58. For a complete description of the DEFINE attributes, see Appendix E, CLASS TAPE DEFINEs.

You can specify the same DEFINE attributes for both a labeled-tape RESTORE and labeled-tape BACKUP process. However, the only attributes used are displayed in Table 7-3. The other tape attributes in the DEFINE are ignored. This lets you use the same DEFINE for a labeled tape BACKUP process and a subsequent labeled-tape RESTORE process without having to redefine the tape attributes.

Table 7-3. DEFINE Attributes Used With RESTORE

Attribute	Use	Description
DEVICE	Optional	Specifies the tape drive to use
FILEID	Optional	Specifies the name of the tape set (file) and must match FILEID on the tape label
GEN	Optional	Indicates that this file is part of a generation group and must match GEN value on the tape label
LABELS	Required	Must be specified as BACKUP, IBMBACKUP, or BYPASS
MOUNTMSG	Optional	Specifies a comment message to be displayed to the operator
SYSTEM	Optional	Specifies the node (system) where all tapes must be mounted
VERSION	Optional	Specifies a version within one generation and must match VERSION number on the tape label
VOLUME	See Note	Specifies the volume ID of the tape

Note. Though the VOLUME attribute is required for CLASS TAPEDEFINEs and optional for CLASS TAPECATALOGDEFINEs, you cannot specify it for LABELS BYPASS DEFINEs. For more information on CLASS TAPECATALOGDEFINEs, see the *DSM/Tape Catalog User's Guide*.

In your DEFINE for a labeled-tape RESTORE process, specify one of:

LABELS BACKUP	Identifies the format of the tapes to be read for the RESTORE process as NonStop BACKUP labeled tapes. Labeled backup tapes written by BACKUP always have an ANSI label with a B in the Label-Standard-Version field in VOL1, and the files are in tape format 3.
LABELS IBMBACKUP	Identifies the format of the tapes to be read for the RESTORE process as IBM labeled tapes. The IBM labeled tapes always have an IBM standard volume label in the Label-Standard Version field in VOL1, and the files are in tape format 3.
LABELS BYPASS	A special purpose DEFINE attribute that lets you recover from parity errors you might have received while attempting a RESTORE process using a NonStop labeled backup tape.

For the RESTORE process, use the VOLUME attribute to specify the volume ID of a single labeled backup tape or volume IDs of multiple labeled backup tapes.

Using Scratch Tapes

The RESTORE utility is unable to use scratch tapes because it reads existing data. You can use either BACKUP or IBM (Backup) tapes for the RESTORE process, and you must specify the VOLUME attribute for any labeled tape RESTORE processes.

Using the LABELS BYPASS DEFINE Attribute

Use the LABELS BYPASS DEFINE attribute only with NonStop BACKUP or IBM labeled backup tapes. You cannot read ANSI labeled tapes using the LABELS BYPASS DEFINE attribute.

The LABELS BYPASS attribute lets you:

- Restore files even after you have received a parity error using NonStop BACKUP or IBM labeled tapes.
  - If a parity error occurs when RESTORE is attempting to read the label of a NonStop BACKUP or IBM labeled tape, the automatic volume recognition (AVR) process fails and the tape is automatically unloaded.
  - To recover from a parity error on volume one of a labeled-tape RESTORE process, use the LABELS BYPASS attribute in a DEFINE. In a subsequent RESTORE process, the LABELS BYPASS attribute lets RESTORE skip the read process of the BACKUP labeled tape. RESTORE only reads the beginning-of-volume label (VOL1) and the beginning-of-file-section label group (HDR1, HDR2) to identify the tape.
  - Any request for a labeled-tape operation in LABELS BYPASS mode goes to the operator console ($0) and the MEDIACOM console (if one exists) to receive operator permission. To permit the request and release the tape drive for use in LABELS BYPASS mode, the operator must issue the appropriate MEDIACOM ACCEPT command.
- Restore multiple tape sets that consist of both labeled and unlabeled NonStop BACKUP tapes by using the LABELS BYPASS DEFINE attribute. This type of process causes additional prompts to be issued by RESTORE.
- Mount the tape for a RESTORE process without having to specify the tape volume ID in your DEFINE, if you restore a NonStop BACKUP labeled tape.

Examples

To create a DEFINE and use it in a RESTORE command:

- To use the DEFINE =MYBACKUP to specify that the labeled backup tapes NY082 and NY004 be used for the RESTORE process (if you use the VOLUME attribute, you must specify tape volume IDs in the order the tapes were written):

```
1> SET DEFMODE ON
2> ADD DEFINE =MYBACKUP, CLASS TAPE, LABELS BACKUP, &
2> &VOLUME (NY082,NY004)
3> RESTORE =MYBACKUP, *.*.*, LISTALL, TAPEDATE
```

  The two labeled tapes specified in this first example were written by the BACKUP utility, which means the files are in tape format 3, and the tape label includes a B in the Label-Standard-Version field.

- To restore a labeled tape without using automatic volume recognition (AVR):

```
1> ADD DEFINE =LAZY, CLASS TAPE, LABELS BYPASS, &
1> &DEVICE $TAPE1
2> RESTORE =LAZY, *.*.*, LISTALL, TAPEDATE
```

  $ZSVR (the labeled tape server process) sends a message to the operator console and the MEDIACOM console (if one exists) to request the release of the tape drive for use in LABELS BYPASS mode. The operator must issue the MEDIACOM ACCEPT command to release (or open) the tape drive. MEDIACOM is only used if the BLPCHECK option is set for the tape drive. Tape volume IDs cannot be specified when LABELS BYPASS is used.

For more information on labeled tape operations, see the *DSM/Tape Catalog Operator Interface (MEDIACOM) Manual*.

This example shows how to find out information on backup sessions and how to use ADD DEFINE and RESTORE commands:

- To find out which backup sessions have the file you want to restore, use the MEDIACOM INFO DISKFILE command. This example lists two backups on the same physical tape:

```
1> info diskfile \CAUNI3.$FOX082.AUG1999.cusinfo,
Silo1_FileCat ,detail

File Catalog       \CAUNI3.SILO1_FILECAT
Tape File          W00007
Generation         1
Version            0
Time Archived      18 Dec 2000, 18:40:35
Physical Copy      1
Logical Copy       1
Disk File          \CAUNI3.$FOX082.AUG1999.CUSINFO
Disk File Status   VALID
Open               FALSE
Broken             FALSE
RF Needed          FALSE
Corrupt            FALSE
Audited            FALSE
File Code          963
File Type          ENSCRIBE
EOF                8192
Last Mod Time      14 Jul 2000,  9:53:21

   Volume Catalog  \CAUNI3.SILO1_VOLCAT
   Pool Name       WEEKLY_APPENDABLE
   Tape Name       005907
   Volume Index    1
   Tape Set ID     005907

File Catalog       \CAUNI3.SILO1_FILECAT
Tape File          W00007
Generation         2
Version            0
```

```
Time Archived      19 Dec 2000,  8:07:20
Physical Copy      1
Logical Copy       1
Disk File          \CAUNI3.$FOX082.AUG1999.CUSINFO
Disk File Status   VALID
Open               FALSE
Broken             FALSE
RF Needed          FALSE
Corrupt            FALSE
Audited            FALSE
File Code          963
File Type          ENSCRIBE
EOF                8192
Last Mod Time      14 Jul 2000,  9:53:21

   Volume Catalog  \CAUNI3.SILO1_VOLCAT
   Pool Name       WEEKLY_APPENDABLE
   Tape Name       005907
   Volume Index    1
   Tape Set ID     005907

2 disk files returned.
```

- To restore the file from the first backup session, use the ADD DEFINE. You must specify the GENERATION and VERSION. Otherwise, their default values would be for the latest backup performed.

```
2> add define =DEF1, class tapecatalog, pool
WEEKLY_APPENDABLE, volcat
\CAUNI3.SILO1_VOLCAT, gen 1, version 0, device $tape0,filecat
\CAUNI3.SILO1_FILECAT, FILEID W00007, USE IN

3> INFO DEFINE =DEF1,DETAIL

   Define Name        =DEF1
   CLASS              TAPECATALOG
   LABELS             ANSI
   CATALOG            ON
   USE                IN
   FILEID             W00007
   GEN                1
   VERSION            0
   FILECAT            \CAUNI3.SILO1_FILECAT
   VOLCAT             \CAUNI3.SILO1_VOLCAT
   POOL               WEEKLY_APPENDABLE
   DEVICE              \CAUNI3.$TAPE0
```

- To restore the file to disk volume $FOX004, use the RESTORE command:

```
4>RESTORE =DEF1, $FOX082.AUG1999.CUSINFO, LISTALL,
SOURCEDATE, VOL $FOX004
```

Restoring in Volume Mode

In volume mode, RESTORE copies an entire tape to a disk volume. Only the super ID (255,255) can run a volume-mode RESTORE process.

When performing a volume-mode RESTORE process:

- The disk volume must be in the DOWN state.
- Do not restore an outdated tape. The effect is similar to using SCF to bring up an old version of a volume. Therefore, the same considerations for the SCF START command apply. For more information, see the *SCF Reference Manual for the Storage Subsystem*.
- When restoring to a mirrored disk, neither mirror should be up during the process. If only one mirror is being restored, the other mirror should be in the DOWN or HARD DOWN state. When restoring to only one mirror, RESTORE checks that the other mirror is in the DOWN or HARD DOWN state, and terminates the RESTORE process if this is not the case.
- Do not use volume mode to restore the system disk ($SYSTEM), because the operation requires both mirrors to be DOWN which is an unreasonable state for $SYSTEM. Do not back up $SYSTEM in volume mode unless there is a good reason. If you must back up $SYSTEM in volume mode, do so only after carefully considering that the disk must be restored under a different name.
- RESTORE can restore a volume-mode backup tape to any disk volume that has the same or greater capacity.
- The disk process type of the target disk in a RESTORE process must be the same as the disk process type of the disk image on tape.
- If you want to restore a volume-mode BACKUP to an upgraded disk, you must first issue an SCF START command to the new target disk to update its internal tables. For example:
  - A volume-mode backup was performed on a device with subtype XX.
  - You need to issue an SCF START command on the new target volume, then an SCF STOP command.
  - Now you can perform a volume-mode RESTORE to the new target volume (on a device with subtype YY).
- Microcode and bootstrap sections are deleted from the disk image during a volume-mode RESTORE process. These special disk image sections are specific to the hardware configuration of the disk at the time of the backup and might not apply to the restored disk. Generally, these sections are found only on $SYSTEM and should not present problems for any other disks.
- Do not use volume mode to compress a volume. No disk compression is performed by volume-mode BACKUP or RESTORE processes.

- SQL catalog consistency during a volume-mode RESTORE process is only assured if catalogs and the files they describe exist on the same volume.
- You must use the RENAME option if you use the TARGET and VOLUMEMODE options to restore data to a disk that is different from the original source disk (and both the target disk and the source disk are on the same system). Otherwise, error 66 is returned when an SCF START command is issued to the restored disk. By using the RENAME option, you can prevent the problem:

  ```
  RESTORE $tape,VOLUMEMODE,*,TARGET $lev-for-$data1,RENAME
     $data1
  ```

- Any volume-mode BACKUP/RESTORE request to an SMF virtual disk results in the error message:

  ```
  Volume Mode BACKUP/RESTORE is incompatible with SMF
  virtual disks, use File Mode.
  ```

- If you attempt a volume-mode backup from a disk with a capacity greater than 2 GB, you will be unable to restore it to an operating system version prior to D30, and this message is generated:

  ```
  The source disk has a capacity of more than 2 GB, which
  cannot be restored to a pre-D30 system.
  ```

Example

This example displays the sequence of commands used in a volume-mode RESTORE process.

Note. The target disk must be in the DOWN state before a volume-mode RESTORE command can be executed. In a file-mode RESTORE process, the target disk remains UP.

Note. Any volume-mode BACKUP/RESTORE request to an SMF virtual disk results in the error message: “Volume Mode BACKUP/RESTORE is incompatible with SMF virtual disks, use File Mode.”

In this example, $DATA is the target disk to be restored. $DATA is configured as a DP2 mirrored volume on logical device $9 and is currently UP.

Before entering any commands, you need to know the disk process type of the backup tape. For a volume-mode RESTORE process, the target disk and the disk image on tape must have the same disk process type.

1. Check the current state of the disk:

   ```
   1> SCF
   STATUS DISK $DATA
   ```

2. Verify that there are no new bad sectors on the specified disk drive. This is a normal procedure to perform before you bring down the disk for the RESTORE process.

   ```
   INFO DISK $DATA, BAD
   ```

If you find any unspared defective sectors:

a. Use the SCF CONTROL DISK, SPARE command to correct the situation.

b. Recheck the status of both halves of the mirrored volume using the INFO DISK command. You must find no new defects before continuing.

3. Bring down all paths of the disk to be restored. RESTORE requires that both halves of a mirrored volume be in the DOWN state to perform a volume-mode RESTORE.

```
STOP DISK $9
```

4. Ensure that all paths of the disk volume are DOWN:

```
INFO DISK $DATA, DETAIL
```

5. Exit SCF.

6. Issue the volume-mode RESTORE command:

```
3> RESTORE $TAPE4, VOLUMEMODE, *, TARGET $9-M
```

7. Check the state of the disk after the RESTORE process has finished. All paths to the disk should still be DOWN.

```
4> SCF
INFO DISK $DATA, DETAIL
```

8. Issue the appropriate SCF command to bring both halves of the mirrored disk UP:

```
5> SCF
START DISK $DATA
```

A

BACKCOPY, BACKUP, and RESTORE Messages

These messages are produced by the BACKCOPY, BACKUP, or RESTORE utilities. When a message is displayed at your terminal, it is preceded by either *ERROR* or *WARNING*.

49

```
access violation
```

Cause. The version of BACKUP or RESTORE in use is not licensed.

Effect. SQL files cannot be backed up.

Recovery. Use a licensed version of BACKUP or RESTORE.

1070

```
The process's program file is not licensed.
```

Cause. The version of BACKUP or RESTORE in use is not licensed.

Effect. SQL files cannot be backed up.

Recovery. Use a licensed version of BACKUP or RESTORE.

1500

```
tape-drive: Tape error nnn (procedure).
```

Cause. A file-system error occurred when the utility tried to back up, copy, or restore a file.

Effect. The utility terminates due to an error in *procedure*.

Recovery. For corrective action for the error indicated by *nnn* and rerun the utility, see the *Guardian Procedure Errors and Messages Manual*.

2003

```
Syntax error (Invalid file name).
```

Cause. An invalid format was used for the name of the tape drive. A device name begins with a dollar sign ($) followed by one through seven alphanumeric characters. The first character must be a letter.

Effect. The utility terminates.

Recovery. Reenter the command using a valid tape drive name.

2012

```
Internal error. error-description. optional-additional-text
```

Cause. An internal error occurred caused by *error-description*.

Effect. The utility terminates and will usually be followed by a TRACE which should be supplied together with the complete error text and the exact command entered when submitting a case for investigation.

Recovery. For corrective action for the error indicated in the message, see the *Guardian Procedure Errors and Messages Manual*.

2013

```
File not dumped : nnn
```

Cause. BACKUP encountered a disk with no files that met the backup criteria.

Effect. The utility terminates.

Recovery. For corrective action for the error number indicated by *nnn*, see “File-System Errors” in the *Guardian Procedure Errors and Messages Manual*.

3*nnn*

```
File aborted filename : [read | write | open] error
```

Cause. A file-system error occurred when the utility tried to read, write, or open a file. The number 3 in the error number indicates that this is a disk error. The *nnn* parameter is the file-system error number.

Effect. The utility terminates.

Recovery. For corrective action for the error indicated by *nnn* and rerun the utility, see the *Guardian Procedure Errors and Messages Manual*.

6000

```
Syntax error (Invalid fileset).
```

Cause. An invalid format was used for the file-set name. The proper file-set format is:

```
[[[ \node.]$volume.]subvolume.]file-id
```

Effect. The utility terminates.

Recovery. Reenter the command using a valid file set.

6011

```
Syntax error (Invalid filename mapping): fileset must be
restricted i.e '*'in place of a system/vol/subvol/filename.
```

Cause. The `TO` clause of the `MAP NAMES` option supports "*" when it is the only character in the `system/vol/subvol/filename` section. Partial name changes are not supported. Each file-name section can either be preserved intact ("*") or changed to a user-specified name (no embedded wildcards).

Effect. The utility terminates.

Recovery. Reenter the command using a valid file-set mapping.

7000

```
File skipped.  Cannot convert this individual partition since
records must be redistributed.  Must convert entire
partition.
```

Cause. BACKUP or RESTORE tried to adjust the extent sizes of individual partitions when converting them from one disk-process type to another. The extent sizes could not be adjusted so that the records of relative and entry-sequenced files could remain in the same partition.

Effect. The file is skipped, and BACKUP or RESTORE proceeds to the next file.

Recovery. The entire partitioned file must be converted in one process. Back up the source file again without specifying the BACKUP PARTONLY ON option. Or you can restore each partition to the same disk-process type (that is, no conversion) and then back up the reconstructed partitioned file without specifying the BACKUP PARTONLY ON option.

7001

```
File aborted.  File label for this file is longer than the
tape block size.
```

Cause. The tape file label describing the attributes of the file contains more information than could be recorded in one tape record of the size specified by the BLOCKSIZE.

Effect. BACKUP skips the file and proceeds to the next file.

Recovery. Specify a larger BLOCKSIZE and retry the process on the file.

7002

```
Cannot use the PARTOF option on a tape recorded with the
PARTONLY option.
```

Cause. An attempt was made to restore a tape that was created using the PARTONLY ON option. The PARTOF and PARTONLY parameters are incompatible.

Effect. RESTORE terminates, and no files are restored.

Recovery. Reissue the command, specifying a valid option mix.

7003

```
PARTONLY was specified and the tape was not recorded with
PARTONLY.  No partitions restored.
```

Cause. The command specified the PARTONLY ON option, but the tape was not created using the PARTONLY ON option of BACKUP.

Effect. No partitions are restored.

Recovery. Re-create the tape, by using the PARTONLY ON option for BACKUP and RESTORE, or delete the PARTONLY ON option from the RESTORE command.

7004

```
PARTONLY was not specified and the tape was recorded with
PARTONLY.  No partitions restored.
```

Cause. The RESTORE command did not specify the PARTONLY ON option, but the tape was created using the PARTONLY ON option of BACKUP.

Effect. No partitions are restored.

Recovery. Re-create the tape by using the PARTONLY ON option for BACKUP and RESTORE, or add the PARTONLY ON option to the RESTORE command.

7005

```
Audited file skipped.
```

Cause. The utility tried to back up a file audited by TMF, but the AUDITED option was not included in the command.

Effect. BACKUP or RESTORE skips the audited file and continues processing the file set.

Recovery. To back up or restore audited files, include the AUDITED option in the command. Rerun the utility, specifying the audited files that were skipped. To restore audited files, see AUDITED on page 3-13 for BACKUP, or on page 7-7 for RESTORE.

7006

```
Audit turned off.
```

Cause. The TURNOFFAUDIT option was specified.

Effect. An audited file on tape was restored as a nonaudited file.

Recovery. Informational message only; no corrective action is needed.

7007

```
Privileged license turned off.
```

Cause. The privileged license was turned off.

Effect. None.

Recovery. Informational message only; no corrective action is needed.

7008

```
PROGID turned off.
```

Cause. The program ID (PROGID) option was turned off.

Effect. None.

Recovery. Informational message only; no corrective action is needed.

7009

```
Rebuild file not found on tape(s).
```

Cause. A request was made to REBUILD a file set, but the file set was not found on the tape.

Effect. RESTORE terminates.

Recovery. Informational message only; no corrective action is needed.

7010

```
Could not make file audited (Error n).
```

Cause. An attempt to restore a file that was previously audited by TMF was successful, but the file is not currently audited. This can occur when the file is restored, TMF has not been started, or TMF has not been configured.

Effect. The file is restored, and RESTORE continues execution.

Recovery. Audit the file using TMF.

7011

```
File is audited.
```

Cause. The file that was backed up was audited by TMF.

Effect. None.

Recovery. Informational message only; no corrective action is needed.

7012

```
Disk data error encountered (Error n, Address b, Length l).
```

Cause. BACKUP encountered an error when trying to read the file data.

Effect. BACKUP skips the file and proceeds to the next file.

Recovery. Contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7013

```
Data lost (Error n Address b, Length l).
```

Cause. BACKUP encountered an error when trying to read data. The IGNORE option was specified.

Effect. None of the data could be read. Because the IGNORE option was specified, all zeros are written to tape for the data.

Recovery. Contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7014

```
Data bad (Error n, Address b, Length l).
```

Cause. BACKUP encountered an error trying to read data. The IGNORE option was specified.

Effect. Because the IGNORE option was specified, BACKUP writes the bad data to tape.

Recovery. Contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7015

```
File aborted.  End-of-file shrank during the backup.
```

Cause. The BACKUP OPEN option was specified, and the end-of-file was modified by another process.

Effect. BACKUP skips the file and proceeds to the next file.

Recovery. Make sure the system is not under heavy use before backing up files. The files can be open for writing, but they should not be updated (actively) while BACKUP is running.

7016

```
Premature end-of-file.
```

Cause. The end-of-file changed during the backup.

Effect. BACKUP skips the file and proceeds to the next file.

Recovery. Contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7017

```
File open for write.
```

Cause. The file being copied by BACKUP was open for write, and the OPEN option was specified.

Effect. BACKUP copies the file to tape.

Recovery. Informational message only; no corrective action is needed.

7018

```
Audited file open for write.
```

Cause. A file was open for write when it was backed up and audited by TMF.

Effect. BACKUP copies the file to tape.

Recovery. Informational message only; no corrective action is needed.

7019

```
File was open for write during backup.
```

Cause. The file that was restored was open for write when it was backed up.

Effect. RESTORE continues.

Recovery. Informational message only; no corrective action is needed.

7020

```
File skipped.  Lock encountered in file and user specified
skip.
```

Cause. BACKUP was run with the MSGONLOCK option specified, and a locked file or record was encountered. User instructed BACKUP to skip the file.

Effect. BACKUP skips the locked file and proceeds with the next file.

Recovery. Informational message only; no corrective action is needed.

7021

```
Tape version of the BACKUP tape is not compatible with the
version of RESTORE.
```

Cause. An attempt was made to restore a tape that was created using a newer version of BACKUP.

Effect. RESTORE terminates.

Recovery. Use a newer version of RESTORE.

7022

```
Command line is empty and the in file was not specified.
```

Cause. The command line was empty, and no input file was specified.

Effect. The utility terminates.

Recovery. Reenter the command and specify the parameters.

7023

```
EXT parameter not applied when converting an individual
partition.
```

Cause. New extents were specified when attempting to convert individual partitions.

Effect. BACKUP or RESTORE converts the file with the old extent sizes.

Recovery. New extents can be specified only when converting an entire partitioned file.

7024

```
EXT parameter not applied when converting unstructured
partitioned files.
```

Cause. Although the EXT option was used, the extent-size specification was ignored because extent sizes for unstructured files must remain the same.

Effect. BACKUP or RESTORE converts the file with the old extent sizes.

Recovery. Informational message only; no corrective action is needed.

7025

```
File skipped.  File does not have alternate key file
specified with ALTFILE.
```

Cause. The ALTFILE option was used to specify an alternate-key file that does not exist for the file.

Effect. BACKUP or RESTORE skips the file and proceeds to the next file.

Recovery. Informational message only; no corrective action is needed.

7026

```
File skipped.  File does not have secondary partition
specified with PART.
```

Cause. A partition was specified with the PART option, but the file does not have a secondary partition.

Effect. BACKUP or RESTORE skips the file and proceeds to the next file.

Recovery. Informational message only; no corrective action is needed.

7027

```
No files written on tape.
```

Cause. BACKUP was unable to write any files to tape because of previous errors.

Effect. BACKUP terminates.

Recovery. Create a new BACKUP tape after correcting any errors.

7028

```
File aborted (Error nnn).
```

Cause. A file-system error occurred.

Effect. BACKUP or RESTORE skips the file and proceeds to the next file.

Recovery. For the appropriate corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

7029

```
Volume aborted (Error nnn).
```

Cause. A file-system error occurred.

Effect. BACKUP or RESTORE skips the file and proceeds to the next file.

Recovery. For the appropriate corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

7030

```
Secondary partition not found on tape.
```

Cause. RESTORE could not find the requested secondary partition of the partitioned file on tape. The tape has an invalid format. There are three possible reasons:

- BACKUP was stopped in the middle of a BACKUP process, resulting in an incomplete BACKUP tape. The “Bad Tape Format” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape or RESTORE is not checking correctly due to a software error.

Effect. RESTORE terminates.

Recovery. If the invalid format is the result of one of the first two causes, the files cannot be restored. If you suspect the problem is the result of the third cause, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7031

```
Volume aborted.  Cannot convert files when PARTIAL is
specified.
```

Cause. An attempt was made to convert files using the PARTIAL option.

Effect. No more files from the volume are processed, and BACKUP proceeds to the next volume.

Recovery. Reenter the command without specifying PARTIAL.

7032

```
This tape can only be restored with TNS/II(B10 or later) or
RESTORE2 (B00).
```

Cause. The tape was created by BACKUP2.

Effect. BACKUP execution continues.

Recovery. Use B10 RESTORE or B00 RESTORE2.

7033

```
This tape can only be restored with TNS/II RESTORE (B41, C00
or later).
```

Cause. The tape was created by BACKUP2.

Effect. BACKUP execution continues.

Recovery. Use B41 RESTORE.

7034

```
The alternate key files of this entry sequenced file will
have to be reloaded using FUP LOADALTFILE.
```

Cause. The file conversion caused the record addresses to change, making the record addresses stored in the alternate keys invalid.

Effect. BACKUP or RESTORE converts the file.

Recovery. Reload the alternate-key files by using the FUP LOADALTFILE command.

7035

```
Old format tape -- no checksums.
```

Cause. The tape was created using BACKUP with the OLDFORMAT option or with an old version of BACKUP that does not write checksums. Consequently, RESTORE cannot perform checksum verification.

Effect. RESTORE continues.

Recovery. Informational message only; no corrective action is needed.

7036

```
File aborted.  Bad DP2 structured block encountered on tape.
```

Cause. An invalid DP2 structured block was encountered when attempting to RESTORE a file. This error occurs only when the DP2 file on tape is being converted to a DP1 file on disk.

Effect. RESTORE skips the file and proceeds to the next file.

Recovery. The DP2 file must be repaired before it can be converted. Either the file must be restored to a DP2 volume and repaired, or the original DP2 file must be repaired and backed up again.

7037

```
File skipped.  File open for write during backup.  Must use
OPEN option to restore.
```

Cause. An attempt was made to restore a file that was open for write when it was backed up.

Effect. RESTORE skips the file and continues.

Recovery. Use the OPEN option and reissue the RESTORE command.

7038

```
Break detected - operation terminated.
```

Cause. The Break key was pressed, and the IN file is a terminal. The Break key is monitored only while the IN file is being read. Otherwise, the Break key is ignored by BACKUP and RESTORE and results in a command interpreter prompt.

Effect. BACKUP or RESTORE terminates.

Recovery. Reissue the BACKUP or RESTORE command.

7039

```
File aborted.  Too much alternate key and/or partition
information for DP1.
```

Cause. The DP2 file has too many alternate keys and files or too many partitions for conversion to a DP1 file.

Effect. BACKUP or RESTORE skips the file and proceeds to the next file.

Recovery. Eliminate alternate keys, files, or partitions in the DP2 file. Otherwise, the file cannot be converted.

7040

```
File aborted.  Cannot back up an entire partitioned file that
has remote secondary partitions unless tape version is 2 or
greater.
```

Cause. A partitioned file with a remote secondary partition was specified in the file set, but the tape format was not 2 or greater.

Effect. BACKUP skips the file and proceeds to the next file.

Recovery. Use the DP2FORMAT option to generate DP2 format tapes (if necessary).

7041

```
File aborted.  Not all the partitions of this entire file
have the same volume type (DP1/DP2).
```

Cause. The disk-process type of one or more partitions was not the same.

Effect. BACKUP or RESTORE skips the file and proceeds to the next file.

Recovery. Move the partitions so that the disk-process types are the same.

7042

```
File aborted.  Bad structured file on tape.
```

Cause. RESTORE encountered an invalid structured file while trying to convert it.

Effect. RESTORE skips the file and proceeds to the next file.

Recovery. Contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7043

```
Entire partitioned file aborted.
```

Cause. An entire partitioned file was terminated. This message is preceded by an error message indicating that a file was terminated.

Effect. BACKUP or RESTORE skips the entire partitioned file and proceeds to the next file.

Recovery. Correct the error that caused the partition to be terminated, and retry the operation on the file.

7044

```
File not dumped (Error nnn).
```

Cause. A file-system error occurred when BACKUP was trying to copy the file.

Effect. BACKUP skips the offending file and continues processing the rest of the file set.

Recovery. For corrective action for the error number indicated, see the *Guardian Procedure Errors and Messages Manual* and then rerun BACKUP.

7045

```
File not restored (Error nnn).
```

Cause. A file-system error occurred when RESTORE was trying to copy the file to disk.

Effect. RESTORE skips the offending file and continues processing the rest of the file set.

Recovery. For corrective action for the error number indicated, see the *Guardian Procedure Errors and Messages Manual*, then rerun RESTORE.

7046

```
Secondary partition not found.
```

Cause. BACKUP of a partitioned file whose secondary partition volume is down.

Effect. BACKUP of the file fails.

Recovery. Bring up the volume on which the secondary partition is present.

7047

```
File aborted.  Parity error encountered on tape file label.
```

Cause. A parity error was encountered on the tape file label.

Effect. RESTORE skips the file and proceeds to the next file.

Recovery. The file cannot be restored if the parity error occurs on any tape drive.

7048

```
File aborted.  Checksum error encountered on tape file label.
```

Cause. Checksum error encountered on the tape file label.

Effect. RESTORE skips the file and proceeds to the next file.

Recovery. The file cannot be restored if the checksum error occurs on any tape drive.

7049

```
Tape parity error encountered on data block(Bad data byte
offset in file: b,Expected length: e, Actual length: a).
```

Cause. The IGNORE option was specified in the RESTORE command, and a parity error was encountered on a data block.

Effect. RESTORE continues to process the file.

Recovery. The data at the specified location is probably corrupted and requires repair.

7050

```
File aborted.  Tape parity error encountered on data block.
```

Cause. A parity error was encountered on a data block.

Effect. RESTORE skips the file and proceeds to the next file.

Recovery. Use the IGNORE option with RESTORE to force restoration of the file with the bad data block. The restored file will need to be repaired (if possible). Otherwise, it will be necessary to back up the file again.

7052

```
At least one alternate key file has a different volume type
(DP1/DP2).
```

Cause. One or more alternate-key files has a different volume type.

Effect. The files are restored.

Recovery. To make the primary-key file usable, use the FUP ALTER command to change the alternate-key file names so they reference volumes (files) that have the same disk-process type.

7053

```
At least one alternate key file name references a volume that
does not exist
```

Cause. A reference was made to a nonexistent volume by one or more alternate-key file names.

Effect. The file is restored.

Recovery. To make the primary-key file complete, use the FUP ALTER command to change the alternate-key file names so they see existing volumes (files).

7054

```
File skipped.  Primary partition would have the same volume
name as one of the secondary partitions.
```

Cause. The primary partition volume name is the same as one or more of the secondary partitions.

Effect. BACKUP or RESTORE skips the partitioned file and continues processing the fileset.

Recovery. Reenter the command using the PART or VOL parameter to specify a different volume name for the partitions.

7055

```
File skipped.  Duplicate secondary partition name.
```

Cause. Two or more secondary partitions of a partitioned file have the same name.

Effect. BACKUP or RESTORE skips the partitioned file and continues processing.

Recovery. Use the PART parameter to specify a different volume name for the secondary partition.

7057

```
At least one secondary partition file name references a
volume that does not exist.
```

Cause. When restoring an individual primary partition, a reference was made to a nonexistent volume by one or more secondary-partition file names.

Effect. The file is restored.

Recovery. To make the partitioned file usable, use the FUP ALTER command to change the partition volume names to see volumes (files) that exist.

7058

```
File aborted.  The data of this DP1 structured file is in DP2
format.
```

Cause. DP2 structured file data was put in a DP1 file.

Effect. RESTORE skips the file and proceeds to the next file.

Recovery. Investigate the cause of the invalid file and eliminate the possibility of its recurrence.

7060

```
File skipped.  Record length of DP2 key-sequenced source file
is too large for a DP1 file.
```

Cause. The record length of the DP2 source file was too long for conversion to a DP1 file.

Effect. BACKUP or RESTORE skips the file and proceeds to the next file.

Recovery. Informational message only; no corrective action is needed.

7061

```
Error n on temporary file used for making the directory for
the system disk.
```

Cause. A file-system error occurred in the temporary file used for making the system disk image that eventually is copied to the system image tape (SIT).

Effect. BACKUP terminates.

Recovery. For appropriate corrective action, see “File-System Errors” in the *Guardian Procedure Errors and Messages Manual*. Correct the error and rerun the SYSGEN phase of the installation process.

7062

```
Unable to back up all the files for the System Image tape.
```

Cause. One of the files to be recorded on the system image tape (SIT) for SYSGEN was terminated.

Effect. BACKUP terminates.

Recovery. Correct the problem that caused the file to terminate and rerun SYSGEN.

7063

```
This System Image tape is for a different volume type
(DP1/DP2) System Disk.  The OSIMAGE will not boot from the
current volume.
```

Cause. The operating system contained in the OSIMAGE file could not be booted from the current volume.

Effect. RESTORE execution continues.

Recovery. The system image tape (SIT) is built for a specific disk-process type, and the OSIMAGE tape can only be booted on the correct disk process. Move the files to the correct disk process, and retry the boot.

7064

```
Volume aborted.  Cannot convert files when REBUILD is
specified.
```

Cause. An attempt was made to convert files using the REBUILD option.

Effect. No more files from the volume are processed, and RESTORE proceeds to the next volume.

Recovery. Reenter the command without specifying REBUILD.

7066

```
Tape checksum error encountered in the data block.
```

Cause. Checksum error encountered reading the tape.

Effect. Warning message only.

Recovery. Informational message only; no corrective action is needed.

7069

```
Volume aborted.  Temporary conversion volume does not exist.
Use SCRATCHVOL parameter to specify a volume or make sure the
default volume exists.
```

Cause. When attempting to convert files, either the current default volume or the indicated scratch volume was nonexistent.

Effect. BACKUP or RESTORE terminates.

Recovery. Use the SCRATCHVOL parameter to specify a volume or (if the default volume is nonexistent) select an existing volume using the command interpreter VOLUME command.

7073

```
File is corrupt and will not be dumped.
```

Cause. The specified file was marked corrupt, and the IGNORE option was not specified to BACKUP.

Effect. BACKUP skips the file and proceeds to the next file.

Recovery. A corrupted file generally is not recoverable.

7075

```
Secondary partition partition-name is corrupt and will not be
dumped.
```

Cause. The specified secondary partition was marked corrupt, and the IGNORE option was not specified to BACKUP.

Effect. BACKUP skips the entire partitioned file and proceeds to the next file.

Recovery. A corrupted file generally is not recoverable.

7077

```
File is corrupt.
```

Cause. The specified file was marked corrupt.

Effect. The corrupt file is backed up, but the corrupt flag remains set.

Recovery. Informational message only; no corrective action is needed.

7078

```
File is marked broken.
```

Cause. BACKUP of a broken file with the IGNORE option.

Effect. BACKUP file is marked broken.

Recovery. None.

7079

```
Original source file was corrupt.
```

Cause. The specified file was marked corrupt on tape.

Effect. The corrupt file is restored, but the corrupt flag remains set.

Recovery. Informational message only; no corrective action is needed.

7080

```
Original source file was marked broken.
```

Cause. RESTORE of a broken file that is backed up with the IGNORE option.

Effect. RESTORE file is marked broken.

Recovery. None.

7081

```
Original source file was corrupt.  Will not restore.
```

Cause. The specified file was marked corrupt on tape, and the IGNORE option was not specified.

Effect. RESTORE skips the file without restoring it.

Recovery. A corrupt file generally is not recoverable.

7083

```
Secondary partition partition-name is corrupt.
```

Cause. The specified secondary partition was marked corrupt on the tape.

Effect. The corrupt file is restored, but the corrupt flag remains set.

Recovery. Informational message only; no corrective action is needed.

7088

```
Safeguard information cannot be included using this tape
format.  Do not use OLDFORMAT or DP1FORMAT if you want
Safeguard information to be backed up.
```

Cause. A tape format was specified that does not let Safeguard information be included on the tape.

Effect. Files that have Safeguard access lists are backed up, but the Safeguard information is not included with the files.

Recovery. Do not use the OLDFORMAT or DP1FORMAT options if you want Safeguard information to be backed up.

7089

```
Safeguard information for this file cannot be backed up
because the Safeguard Monitor is inaccessible.
```

Cause. The specified file is protected by the Safeguard product, but the Safeguard monitor was not accessible.

Effect. The file is backed up, but the associated Safeguard information is not included with the file.

Recovery. Repeat the BACKUP operation when the Safeguard monitor is running.

7090

```
Safeguard information for this file cannot be restored
because the Safeguard Monitor is inaccessible.
```

Cause. The specified file on tape included Safeguard information, but the Safeguard monitor was not accessible.

Effect. The file is restored, but the Safeguard information is not restored. The file is not protected by the Safeguard product.

Recovery. Repeat the RESTORE operation when the Safeguard monitor is running.

7091

```
Safeguard information for this file cannot be backed up
because it is too long when combined with the file label.
```

Cause. BACKUP could not accommodate both the file label and the Safeguard information.

Effect. BACKUP continues, but Safeguard information is not included.

Recovery. None.

7092

```
Safeguard protection has been turned off for this file.
```

Cause. Either the NOSAFEGUARD option was specified, or Safeguard is not running on the system.

Effect. Safeguard information is not backed up or restored.

Recovery. Remove the NOSAFEGUARD option or start Safeguard on the system.

7093

```
Safeguard protection for this file cannot be restored because
of a Safeguard server error nnn.
```

Cause. The specified file on tape was protected by the Safeguard product, but file-system error *nnn* occurred.

Effect. The file is restored, but the Safeguard protection is ignored.

Recovery. For error *nnn* corrective action, see the *Guardian Procedure Errors and Messages Manual*.

7094

```
The tape block size has been rounded down to nn.
```

Cause. BACKUP rounded down the tape block size.

Effect. BACKUP continues.

Recovery. Informational message only; no corrective action is needed.

7095

```
This is a volume mode Backup tape.  Volume mode command
syntax must be used to restore this tape.
```

Cause. An attempt was made to restore a volume-mode BACKUP tape without using volume-mode RESTORE syntax.

Effect. RESTORE terminates.

Recovery. Retry the operation using volume-mode RESTORE syntax.

7103

```
cannot use the PARTOF option to restore SQL objects.
```

Cause. RESTORE of SQL objects using the PARTOF option.

Effect. RESTORE of the object fails.

Recovery. Use the PARTONLY option to restore the specific partitions.

7105

```
Attempt to purge an SQL object while restoring an Enscribe
file.
```

Cause. The Enscribe file has the same name as an existing SQL object.

Effect. File is not restored.

Recovery. Purge the SQL object before performing the RESTORE operation.

7109

```
Too many files specified for the system disk directory.
```

Cause. The number of files specified was too large for one of these reasons:

- The number of files specified would produce a directory extent that would exceed the largest possible extent.
- The number of files specified would cause the system tables to exceed the size of the disk.

Effect. BACKUP ends abnormally.

Recovery. Change the SYSTEM_VOLUME_DIRECTORY_SIZE of the SYSGEN ALLPROCESSORS paragraph so that the size does not exceed 5000. If this does not help, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7120

```
Start file set not in the main file set list.
```

Cause. The file specified in the START option is not in the *backup-files* or *restore-files*.

Effect. BACKUP or RESTORE terminates.

Recovery. Reissue the command, making sure the START file is referenced in the *backup-files* or *restore-files*.

7121

```
Tape boot at beginning of System Image tape not copied
```

Cause. A BACKCOPY operation is being performed on a system image tape.

Effect. The tape boot record is not copied to the destination tape; BACKCOPY continues.

Recovery. Informational message only; no corrective action is needed.

7122

```
Unable to copy all files
```

Cause. File errors caused some files to be skipped in the BACKCOPY operation.

Effect. Files are missing on the destination tape.

Recovery. Reissue the command to see if the file errors are transitory.

7123

```
File exists on disk and is not being replaced
```

Cause. The RESTORE KEEP option is in use.

Effect. Tape files with names identical to files on the destination disk are not restored.

Recovery. Informational message only; no corrective action is needed.

7124

```
Secondary partition contains illegal file attributes
```

Cause. Secondary partitions cannot be backed up because of an invalid file attribute.

Effect. The entire partitioned file is skipped.

Recovery. Informational message only; no corrective action is needed.

7125

```
File not contained on tape due to partial backup
```

Cause. The BACKUP PARTIAL option is in use.

Effect. Files that have not been modified since the date specified with the PARTIAL option are not backed up.

Recovery. Informational message only; no corrective action is needed.

7126

```
REBUILD was not specified and the tape was recorded using the
PARTIAL parameter
```

Cause. The RESTORE REBUILD option was not used to restore a tape created with the BACKUP PARTIAL option.

Effect. Tape not restored.

Recovery. Reissue the command with the REBUILD option.

7127

```
PARTONLY option specified is incompatible
```

Cause. The PARTONLY option was used with incompatible options.

Effect. BACKUP or RESTORE terminates.

Recovery. Reissue the command without the incompatible options. In a BACKUP command, PARTONLY ON cannot be used with the INDEXES IMPLICIT option or with the PART option if extent sizes are specified. In a RESTORE command, PARTONLY ON cannot be used with the EXT or INDEXES IMPLICIT options.

7128

```
Unable to determine file attributes for qualification
```

Cause. The file-set qualifier contains invalid file attributes.

Effect. BACKUP or RESTORE terminates.

Recovery. Reissue the command with valid file attributes. (See Section 3, BACKUP).

7129

```
Non-existent catalog xxx has been created
```

Cause. The RESTORE AUTOCREATECATALOG option is in use.

Effect. A new catalog is created.

Recovery. Informational message only no corrective action is needed.

7130

```
This tape can only be restored with TNS/II RESTORE (C00 or
later).
```

Cause. The tape was created by BACKUP2.

Effect. BACKUP execution continues.

Recovery. Use C00 or a later version of RESTORE.

7131

```
This tape cannot be restored using the MYID option of
RESTORE.
```

Cause. The MYID option cannot be used because the NOMYID option was used when the backup tape was made.

Effect. RESTORE terminates.

Recovery. Do not use the MYID option.

7132

```
The subvolume has been defaulted and the volume was not.
```

Cause. The specified file included a volume name but not a subvolume name. Subvolume defaulting will not be supported in future releases.

Effect. BACKUP or RESTORE continues.

Recovery. Informational message only no corrective action is needed.

7138

```
OUT file cannot be created due to security violation error.
```

Cause. Attempt to create a file with an unauthorized user ID.

Effect. The file is not created.

Recovery. Create the file using an authorized user ID or assign write permission to the user ID.

7139

```
The backup files may be opened for write because the
SHAREOPEN option was used for Backup.  This may result in
inconsistent file data on the backup tape.
```

Cause. The SHAREOPEN option was used in the BACKUP command, which gives other processes read/write access to the backup files during the BACKUP operation. This option presents some risk to data integrity because files could be modified while being backed up.

Effect. BACKUP or RESTORE continues.

Recovery. Informational message only no corrective action is needed.

7141

```
Unable to obtain MYPIN and processor information.
```

Cause. An EXCEED^MYPINERROR has occurred.

Effect. BACKUP or RESTORE continues.

Recovery. Informational message only no corrective action is needed.

7142

```
CATALOGFILES option was used in the Backup command while
CATALOG OFF was specified in the DEFINE.  The files will not
be cataloged in the DSM/TC database.
```

Cause. A conflict exists between the DEFINE attribute and the command line parameter.

Effect. The files are not cataloged.

Recovery. Informational message only no corrective action is needed.

7143

```
The collation object already exists on disk and its
definition is equivalent to the one on tape.
```

Cause. The collation object is identical to the one on the disk.

Effect. None.

Recovery. Informational message only no corrective action is needed.

7144

```
This tape can only be restored with RESTORE (D30 or later).
```

Cause. A D30 version of this product (or later) is required because the BLOCKSIZE is larger than 28 KB.

Effect. BACKUP or BACKCOPY continues.

Recovery. Informational message only no corrective action is needed.

7145

```
This tape was backed up using BLOCKSIZE larger than 28 but
the following tape drive does not support large transfers:
nnnn.
```

Cause. The tape drive does not support large blocks.

Effect. RESTORE or BACKCOPY terminates.

Recovery. Use a tape drive that supports a larger BLOCKSIZE.

7146

```
The following tape drive does not support large transfers:
nnnn.  A value of 28 will be used for BLOCKSIZE.
```

Cause. The tape drive does not support a BLOCKSIZE larger than 28 KB.

Effect. BACKUP continues, but it uses a smaller BLOCKSIZE.

Recovery. Informational message only no corrective action is needed.

7147

```
Files created and stored via OSS and SQL/MX objects are not
supported.
```

Cause. An attempt was made to back up a file that is not a Guardian file.

Effect. All files in ZYQ* subvolumes are skipped.

Recovery. Informational message only no corrective action is needed.

7157

```
The NOSQLDATA option is active. No SQL data can be recovered
using this BACKUP tape.
```

Cause. A BACKUP command was entered using the NOSQLDATA option.

Effect. All data contained in SQL files is skipped during the backup, so no SQL data can be recovered from this tape using RESTORE.

Recovery. Informational message only no corrective action is needed.

7158

```
The REMOTEIOSIZE cannot exceed BLOCKSIZE. It has been rounded
down to nn.
```

Cause. BACKUP rounded down the REMOTEIOSIZE.

Effect. BACKUP continues.

Recovery. Informational message only; no corrective action is needed.

7159

```
The NOSQLDATA option is active. No SQL data is being
recovered from this tape.
```

Cause. A RESTORE command was entered using the NOSQLDATA option.

Effect. All data contained in SQL files is skipped during the RESTORE. An empty version of each SQL object on tape is created at the target location.

Recovery. Informational message only; no corrective action is needed.

7251

```
Disk volume diskname is in the diskstate.  It must be in the
DOWN state.
```

Cause. A disk volume was in a state other than down.

Effect. BACKUP or RESTORE terminates.

Recovery. Bring down the disk volume and retry the operation.

7252

```
Not a volume mode BACKUP tape.
```

Cause. An attempt was made to RESTORE a tape that was not a volume-mode BACKUP tape.

Effect. RESTORE terminates.

Recovery. Mount the correct tape and retry the operation.

7253

```
The capacity of the target disk must be greater than or equal
to the size of the disk image on tape.
```

Cause. An attempt was made to perform a RESTORE, but the capacity of the target disk was insufficient.

Effect. RESTORE terminates.

Recovery. Specify a target disk that has sufficient space and retry the operation.

7254

```
Target disk process type must match the disk process type of
the source disk on tape.
```

Cause. An attempt was made to perform a RESTORE, but the target disk-process type did not match the disk-process type on tape.

Effect. RESTORE terminates.

Recovery. Specify a target disk with the same disk process type as the disk image on tape.

7255

```
Name of the volume on tape does not match the volume name
specified.
```

Cause. An attempt was made to RESTORE a tape, but the name of the volume on the tape did not match the specified volume name.

Effect. RESTORE terminates.

Recovery. Retry the operation with the correct volume name or specify the “*” wild card to match any volume.

7257

```
The disk does not have mirror (-M).
```

Cause. BACKUP or RESTORE encountered a disk that did not have a mirror.

Effect. BACKUP or RESTORE terminates.

Recovery. Retry the operation without specifying a mirror.

7258

```
The disk is a mirrored pair and the primary and mirror
volumes are not identical.  The valid primary or mirror
volume must be determined and specified.
```

Cause. BACKUP encountered a mirrored disk pair, but the primary and mirror volumes were not identical.

Effect. BACKUP terminates.

Recovery. Determine and specify the valid primary or mirror volume and retry the operation.

7259

```
Bad disk format or disk has been partially formatted.
```

Cause. BACKUP or RESTORE encountered a disk that has a faulty format or that has been only partially formatted.

Effect. BACKUP or RESTORE terminates.

Recovery. Fix the disk format or change the disk name and retry the operation.

7260

```
Bad disk volume label.
```

Cause. BACKUP encountered a faulty disk volume label.

Effect. BACKUP terminates.

Recovery. Correct the disk volume label and retry the operation.

7264

```
The subtype of the disk on tape (subtype number) is not known
by (this version of) this program.
```

Cause. An attempt was made to RESTORE a tape, but the disk subtype on the tape was unknown to the version of RESTORE being used.

Effect. RESTORE terminates.

Recovery. None.

7266

```
One or more boot strap sections have not been restored to the
target volume (Use PUP REPLACEBOOT).
```

Cause. A RESTORE was performed, but one or more bootstrap sections were not restored.

Effect. RESTORE continues.

Recovery. Use the PUP REPLACEBOOT command or the SCF CONTROL DISK, REPLACEBOOT command to replace the boot strap.

7267

```
The 3106 microcode section has not been restored to the
target volume (Use PUP REPLACEBOOT).
```

Cause. A RESTORE was performed, but the 3106 microcode section was not restored.

Effect. RESTORE continues.

Recovery. Use the PUP REPLACEBOOT command or the SCF CONTROL DISK, REPLACE BOOT command to replace the microcode section.

7268

```
The disk Free Space Table is broken (an invalid entry was
detected).
```

Cause. An invalid entry was detected in the disk free space table.

Effect. BACKUP terminates.

Recovery. If necessary, use the WHOLEDISC option to back up the volume. For error 52 corrective action, see the *Guardian Procedure Errors and Messages Manual*.

7269

```
When restoring a disk image to only one mirror of a mirror
pair, the other mirror must be down or hard-down.
```

Cause. An attempt was made to restore a disk image to one mirror of a mirrored pair, but the other mirror was not down.

Effect. RESTORE terminates.

Recovery. Bring the other mirror into a down or hard-down state and retry the operation.

7270

```
Volume mode backup and restore can only be performed under
the SUPER.SUPER user ID.
```

Cause. An attempt was made to perform a volume-mode BACKUP or RESTORE with a user ID other than the super ID (255,255).

Effect. BACKUP or RESTORE terminates.

Recovery. Log on as the super ID and retry the operation.

7271

```
The audit record cannot be sent to the Audit Collector due to
file-system-name.
```

Cause. A file-system error occurred during the attempt to send an audit record to the Safeguard product during a volume-mode backup.

Effect. BACKUP continues.

Recovery. For more information, see the *Guardian Procedure Errors and Messages Manual*.

7274

```
Trust flag turned off.
```

Cause. The state of FLTrustFlags for the file being restored is set to OFF.

Effect. None.

Recovery. Informational message only; no corrective action is needed.

Note. Message 7274 appears only on systems running H-series RVUs or J-series RVUs.

7500

```
Not positioned to the beginning of the tape.
```

Cause. An attempt was made to run BACKUP or RESTORE before positioning the tape to the beginning of the tape.

Effect. BACKUP or RESTORE terminates.

Recovery. Reenter the command after positioning the tape to the beginning.

7501

```
7 track tape drive cannot be used for BACKUP/RESTORE.  Must
be a 9 track drive.
```

Cause. An attempt was made to run BACKUP or RESTORE with a 7-track tape drive.

Effect. BACKUP or RESTORE terminates.

Recovery. Use a 9-track tape drive.

7502

```
Bad tape format (Bad length for a DP1 tape file label).
```

Cause. An attempt was made to restore a tape, but a BACKUP tape in the proper format could not be found. There are three possible reasons:

- BACKUP was stopped in the middle of a backup operation, resulting in an incomplete BACKUP tape. The “Bad Tape Format” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape or RESTORE is not checking correctly due to a software error.

Effect. RESTORE terminates.

Recovery. If the invalid format is the result of one of the first two causes, the files cannot be restored. If you suspect the problem is a result of the third cause, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7503

```
Bad tape format (Bad section length in DP2 tape file label).
```

Cause. An attempt was made to restore a tape, but a BACKUP tape in the proper format could not be found. There are three possible reasons:

- BACKUP was stopped in the middle of a backup operation, resulting in an incomplete BACKUP tape. The “Bad Tape Format” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape or RESTORE is not checking correctly due to a software error.

Effect. RESTORE terminates.

Recovery. If the invalid format is the result of one of the first two causes, the files cannot be restored. If you suspect the problem is a result of the third cause, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7504

```
Bad tape format (Bad length for a DP2 tape file label).
```

Cause. An attempt was made to restore a tape, but a BACKUP tape in the proper format could not be found. There are three possible reasons:

- BACKUP was stopped in the middle of a backup operation, resulting in an incomplete BACKUP tape. The “Bad Tape Format” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape or RESTORE is not checking correctly due to a software error.

Effect. RESTORE terminates.

Recovery. If the invalid format is the result of one of the first two causes, the files cannot be restored. If you suspect the problem is a result of the third cause, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7505

```
Bad tape format (End-of-file encountered where file label or
ending volume label expected).
```

Cause. An attempt was made to restore a tape, but a BACKUP tape in the proper format could not be found. There are three possible reasons:

- BACKUP was stopped in the middle of a backup operation, resulting in an incomplete BACKUP tape. The “Bad Tape Format” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape or RESTORE is not checking correctly due to a software error.

Effect. RESTORE terminates.

Recovery. If the invalid format is the result of one of the first two causes, the files cannot be restored. If you suspect the problem is a result of the third cause, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7508

```
Bad tape format (Record found that is not a file label or
ending volume label).
```

Cause. An attempt was made to restore a tape, but a BACKUP tape in the proper format could not be found. There are three possible reasons:

- BACKUP was stopped in the middle of a backup operation, resulting in an incomplete BACKUP tape. The “Bad Tape Format” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape or RESTORE is not checking correctly due to a software error.

Effect. RESTORE terminates.

Recovery. If the invalid format is the result of one of the first two causes, the files cannot be restored. If you suspect the problem is a result of the third cause, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7509

```
Bad tape format (Ending volume label does not match beginning
volume label).
```

Cause. An attempt was made to restore a tape, but a BACKUP tape in the proper format could not be found. There are three possible reasons:

- BACKUP was stopped in the middle of a backup operation, resulting in an incomplete BACKUP tape. The “Bad Tape Format” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape or RESTORE is not checking correctly due to a software error.

Effect. RESTORE terminates.

Recovery. If the invalid format is the result of one of the first two causes, the files cannot be restored. If you suspect the problem is a result of the third cause, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7510

```
Bad tape format (No files on tape)
```

Cause. An attempt was made to restore a tape, but a BACKUP tape in the proper format could not be found. There are three possible reasons:

- BACKUP was stopped in the middle of a backup operation, resulting in an incomplete BACKUP tape. The “Bad Tape Format” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape or RESTORE is not checking correctly due to a software error.

Effect. RESTORE terminates.

Recovery. If the invalid format is the result of one of the first two causes, the files cannot be restored. If you suspect the problem is a result of the third cause, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7511

```
Bad tape format (Bad continuation indicator for ending volume
label).
```

Cause. An attempt was made to restore a tape, but a BACKUP tape in the proper format could not be found. There are three possible reasons:

- BACKUP was stopped in the middle of a backup operation, resulting in an incomplete BACKUP tape. The “Bad Tape Format” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape or RESTORE is not checking correctly due to a software error.

Effect. RESTORE terminates.

Recovery. If the invalid format is the result of one of the first two causes, the files cannot be restored. If you suspect the problem is a result of the third cause, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7512

```
Bad tape format (Bad continuation indicator for beginning
volume label).
```

Cause. An attempt was made to restore a tape, but a BACKUP tape in the proper format could not be found. There are three possible reasons:

- BACKUP was stopped in the middle of a backup operation, resulting in an incomplete BACKUP tape. The “Bad Tape Format” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape or RESTORE is not checking correctly due to a software error.

Effect. RESTORE terminates.

Recovery. If the invalid format is the result of one of the first two causes, the files cannot be restored. If you suspect the problem is a result of the third cause, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7513

```
Bad tape format (End-of-file encountered where a data block
was expected).
```

Cause. An attempt was made to restore a tape, but a BACKUP tape in the proper format could not be found. There are three possible reasons:

- BACKUP was stopped in the middle of a backup operation, resulting in an incomplete BACKUP tape. The “Bad Tape Format” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape or RESTORE is not checking correctly due to a software error.

Effect. RESTORE terminates.

Recovery. If the invalid format is the result of one of the first two causes, the files cannot be restored. If you suspect the problem is a result of the third cause, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7514

```
Tape drive does not support density selection.
```

Cause. The BACKUP DENSITY option was specified, but the tape drive does not implement density selection.

Effect. BACKUP proceeds. Tape drive uses physical density setting.

Recovery. Informational message only; no corrective action is needed.

7515

```
Bad tape format (End-of-file encountered where beginning
volume label expected).
```

Cause. An attempt was made to restore a tape, but a BACKUP tape in the proper format could not be found. There are three possible reasons:

- BACKUP was stopped in the middle of a backup operation, resulting in an incomplete BACKUP tape. The “Bad Tape Format” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape or RESTORE is not checking correctly due to a software error.

Effect. RESTORE terminates.

Recovery. If the invalid format is the result of one of the first two causes, the files cannot be restored. If you suspect the problem is a result of the third cause, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7516

```
Unable to read beginning volume label(Parity error).
```

Cause. A parity error was encountered when the beginning volume label was read. Possible causes are:

- The density used to record the tape and the current density setting on the tape drive are different.
- The tape-drive heads are dirty.
- The tape drive in use needs realignment.
- The tape drive that wrote the tape needs realignment.

Effect. BACKUP or RESTORE unloads the tape and prompts for a reel.

Recovery. Verify the density and try again. If the parity error persists, type STOP at the prompt. Try other tape drives. If parity errors occur on a number of tape drives, the tape cannot be restored.

7518

```
End of tape encountered near the beginning of the tape.
```

Cause. BACKUP encountered the end-of-tape near the beginning of the tape. Either the tape is too short, or there is a hardware problem in the tape drive, causing it to report the end-of-tape prematurely.

Effect. BACKUP terminates.

Recovery. Use a longer tape. If this is not the cause of the problem, have the tape drive checked.

7519

```
Tape verify encountered end-of-file where beginning/ending
volume label expected.
```

Cause. Either the VERIFYTAPE or VERIFYREEL option was used with BACKUP or RESTORE (LISTONLY). The end-of -file was encountered where either the beginning or ending volume label was expected.

Effect. BACKUP or RESTORE terminates.

Recovery. Contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7520

```
Tape verify encountered ending volume label where a file
label was expected.
```

Cause. Either the VERIFYTAPE or VERIFYREEL option was used with BACKUP or RESTORE (LISTONLY). The ending volume label was encountered where a file label was expected.

Effect. BACKUP or RESTORE terminates.

Recovery. Contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms

- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7523

```
Tape verify found different number of data blocks than were
supposed to have been written.
```

Cause. Either the VERIFYTAPE or VERIFYREEL option was used with BACKUP or RESTORE (LISTONLY). An inconsistency exists between the number of data blocks requested and the number written.

Effect. BACKUP or RESTORE terminates.

Recovery. Contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7524

```
Tape verify found file label problem.
```

Cause. Either the VERIFYTAPE or VERIFYREEL option was used with BACKUP or RESTORE (LISTONLY). A problem was encountered with the file label.

Effect. BACKUP or RESTORE terminates.

Recovery. Have the tape drive serviced.

7525

```
Tape verify found data block problem.
```

Cause. Either the VERIFYTAPE or VERIFYREEL option was used with BACKUP or RESTORE (LISTONLY). A problem was encountered with the data block.

Effect. BACKUP or RESTORE terminates.

Recovery. Have the tape drive serviced.

7526

```
Tape verify found beginning volume label doesn't match what
was supposed to have been written.
```

Cause. Either the VERIFYTAPE or VERIFYREEL option was used with BACKUP or RESTORE (LISTONLY). The beginning volume label did not match what was expected.

Effect. BACKUP or RESTORE terminates.

Recovery. Contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7527

```
System image tape too short.
```

Cause. A tape reel that was too short was specified to create a System Image Tape (SIT).

Effect. BACKUP terminates.

Recovery. The SIT must fit on one reel. Mount a longer tape reel, and reissue the command.

7528

```
Not a BACKUP tape (bad label identifier).
```

Cause. An attempt was made to restore a tape, but the label could not be read.

Effect. RESTORE terminates.

Recovery. Check that you are using the correct tape or create a new BACKUP tape. Then retry the RESTORE operation.

7529

```
Not a BACKUP tape.
```

Cause. BACKUP requested a previous reel, and the reel mounted was not a BACKUP tape.

Effect. BACKUP unloads the tape and prompts for the correct reel.

Recovery. Mount the correct tape and press the Return key.

7530

```
Wrong tape mounted.
```

Cause. For BACKUP, the tape mounted was not the requested tape. For RESTORE, the wrong reel of a multiple reel BACKUP tape set was mounted.

Effect. BACKUP waits for the operator response. RESTORE prompts again for the next reel.

Recovery. For BACKUP, mount the correct tape and press Return. To terminate the backup operation, type STOP and press Return. For RESTORE, mount the correct reel and press Return or check the tape density.

7532

```
Bad tape sequence number value (Expected sequence number: e,
Actual sequence number: a).
```

Cause. BACKUP or RESTORE encountered a BACKUP tape record with an invalid sequence number. Three possible reasons are:

- BACKUP was stopped in the middle of a backup operation, resulting in an incomplete BACKUP tape. The “Bad Tape Sequence Number” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape, or BACKUP or RESTORE is not checking correctly due to a software error.

Effect. BACKUP or RESTORE terminates.

Recovery. If the invalid sequence number is a result of one of the first two causes, the files cannot be restored. If you suspect the problem is a result of the third cause, contact your service provider and provide all relevant information:

- The information in this message

- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7533

```
Out of sequence tape record (Expected sequence number: e,
Actual sequence number: a).
```

Cause. BACKUP or RESTORE encountered a BACKUP tape record that was not in correct order. Three possible reasons are:

- BACKUP was stopped in the middle of a backup operation, resulting in an incomplete BACKUP tape. The “Bad Tape Format” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape, or BACKUP or RESTORE is not checking correctly due to a software error.

Effect. BACKUP or RESTORE terminates.

Recovery. If the invalid format is a result of one of the first two causes, the files cannot be restored. If you suspect the problem is a result of the third cause, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7534

```
Not from the same BACKUP tape set.
```

Cause. An attempt was made to back up a tape that is not part of the same BACKUP tape set as the other reel(s).

Effect. BACKUP or RESTORE unloads the tape and prompts for the correct reel.

Recovery. Mount the correct BACKUP tape, and press Return at the prompt.

7535

```
Bad tape format (Too many data blocks found for file on
tape).
```

Cause. An attempt was made to restore a tape, but a BACKUP tape in the proper format could not be found. Three possible reasons are:

- BACKUP was stopped in the middle of a backup operation, resulting in an incomplete BACKUP tape. The “Bad Tape Format” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape or RESTORE is not checking correctly due to a software error.

Effect. RESTORE terminates.

Recovery. If the invalid format is the result of one of the first two causes, the files cannot be restored. If you suspect the problem is a result of the third cause, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7536

```
Bad tape format (Data block too long for file).
```

Cause. An attempt was made to restore a tape, but a BACKUP tape in the proper format could not be found. Three possible reasons are:

- BACKUP was stopped in the middle of a backup operation, resulting in an incomplete BACKUP tape. The “Bad Tape Format” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape or RESTORE is not checking correctly due to a software error.

Effect. RESTORE terminates.

Recovery. If the invalid format is the result of one of the first two causes, the files cannot be restored. If you suspect the problem is a result of the third cause, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7537

```
Bad tape format (Data block too short for file).
```

Cause. An attempt was made to restore a tape, but a BACKUP tape in the proper format could not be found. Three possible reasons are:

- BACKUP was stopped in the middle of a backup operation, resulting in an incomplete BACKUP tape. The “Bad Tape Format” was detected when unwanted or meaningless information was encountered at the interruption point on the tape.
- The BACKUP tape was corrupted in some way.
- Either BACKUP produced an invalid tape or RESTORE is not checking correctly due to a software error.

Effect. RESTORE terminates.

Recovery. If the invalid format is the result of one of the first two causes, the files cannot be restored. If you suspect the problem is a result of the third cause, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

7542

```
Unable to turn on tape buffering due to insufficient I/O
buffer space (file system error 33).
```

Cause. The tape process reported it was unable to allocate space for buffered mode.

Effect. BACKUP or RESTORE processing continues in unbuffered mode. The rate at which the tape is read or written might be lower than expected.

Recovery. If this message appears frequently, check memory use on the node where the tape process is running. If necessary, reconfigure the system to distribute memory use more evenly or add more physical memory.

7750

```
Number expected.
```

Cause. A required number was missing.

Effect. BACKUP or RESTORE terminates.

Recovery. Reenter the command with the required number.

7751

```
Invalid BLOCKSIZE.  Must be in the range 2 . . 30 and even.
```

Cause. An invalid data record size was specified.

Effect. BACKUP or RESTORE terminates.

Recovery. Make sure your BLOCKSIZE is an even integer value from 2 through 30, inclusive.

7752

```
Invalid system disk directory size.
```

Cause. While attempting to create a system image tape, the directory size specified to SYSGEN in the SYSTEM_VOLUME_DIRECTORY_SIZE option included too many files for the disk.

Effect. BACKUP terminates.

Recovery. Specify a smaller directory size to SYSGEN and rerun SYSGEN.

7753

```
Keyword too long.
```

Cause. The specified keyword was too long.

Effect. BACKUP, BACKCOPY, or RESTORE terminates.

Recovery. Check and reenter the command with the right keyword.

7754

```
Missing parameter.
```

Cause. The command did not contain a required parameter.

Effect. BACKUP, BACKCOPY, or RESTORE terminates.

Recovery. Check the syntax, and reenter the command with the required parameter.

7755

```
Comma expected.
```

Cause. The command line was missing an expected comma.

Effect. BACKUP, BACKCOPY, or RESTORE terminates.

Recovery. Reenter the command with the required comma.

7757

```
Keyword expected.
```

Cause. A required keyword was omitted.

Effect. BACKUP, BACKCOPY, or RESTORE terminates.

Recovery. Check and reenter the command with the right keyword.

7758

```
Unrecognized keyword.
```

Cause. An invalid keyword was used.

Effect. BACKUP, BACKCOPY, or RESTORE terminates.

Recovery. Check and reenter the command with the right keyword.

7759

```
Invalid slack value.  Must be in the range 0 . . .99.
```

Cause. An invalid slack value was specified.

Effect. BACKUP or RESTORE terminates.

Recovery. Make sure the slack value is in the range 0 through 99.

7760

```
Must be local to the system that BACKUP/RESTORE is running
on.
```

Cause. An attempt was made to restore a remote file.

Effect. RESTORE terminates.

Recovery. You can only restore files that are local to the system on which RESTORE resides.

7761

```
Invalid START file set.  '*' wildcard only allowed in
subvolume or name field.
```

Cause. An attempt was made to indicate the START file set by using the wild-card symbol (*) in the node name.

Effect. BACKUP terminates.

Recovery. Use the wild-card symbol (*) only in the subvolume or name field.

7762

```
Start file set not in the main file set list.
```

Cause. An attempt was made to run RESTORE, but the START file set was not in the main fileset list specified to RESTORE.

Effect. RESTORE terminates.

Recovery. Reissue the command, specifying the correct START file set.

7763

```
Unrecognized month name.
```

Cause. An invalid month name was used.

Effect. BACKUP or RESTORE terminates.

Recovery. Correct the command by using only the first three letters of the month name, and retry the command.

7764

```
Bad date/time (Year must be 1975 or beyond).
```

Cause. The date was entered with a year prior to 1975.

Effect. BACKUP or RESTORE terminates.

Recovery. Correct the date or time and reissue the command.

7765

```
Bad date/time.
```

Cause. The date was entered in an incorrect format.

Effect. BACKUP or RESTORE terminates.

Recovery. Correct the date or time in the format *ddmmmyyyy,hh:mm* and reissue the command.

7766

```
Colon between hour and minute expected.
```

Cause. The time was entered without a colon between the hour and minute.

Effect. BACKUP or RESTORE terminates.

Recovery. Reenter the command with the required colon.

7767

```
Required system image parameters missing.
```

Cause. SYSGEN started BACKUP to create a system image tape but failed to provide one or more of the parameters that BACKUP requires for this operation, probably because incompatible versions of SYSGEN and BACKUP are being used.

Effect. BACKUP terminates without writing a system image tape.

Recovery. Run SYSGEN again, either directly or through Install, after ensuring that the SYSGEN phase of the installation process starts a compatible version of BACKUP.

7768

```
Not allowed to specify PARTOF and PARTONLY together.
```

Cause. An attempt was made to restore individual partitions from a tape that contains entire partitioned files and was created with the PARTONLY option.

Effect. RESTORE terminates.

Recovery. Reenter the command without the PARTOF option. PARTOF and PARTONLY are incompatible.

7769

```
Bad density name.  Expecting NRZI, PE, or GCR.
```

Cause. An invalid density name was specified.

Effect. BACKUP or RESTORE terminates.

Recovery. Reenter the command with a valid density name (NRZI, PE, or GCR).

7770

```
Bad density number.  Expecting 800, 1600, or 6250.
```

Cause. An invalid density number was specified.

Effect. BACKUP or RESTORE terminates.

Recovery. Reenter the command with a valid density number.

7771

```
Only a single volume can be backed up when VOL is specified.
```

Cause. An attempt was made to dump more than one volume by using the VOL parameter.

Effect. BACKUP terminates.

Recovery. Restrict the backup to one volume.

7772

```
Only a single volume and subvolume can be dumped when a
subvolume is specified with VOL.
```

Cause. An attempt was made to dump more than one volume and subvolume by using the VOL parameter.

Effect. BACKUP terminates.

Recovery. Restrict the BACKUP to one subvolume.

7774

```
Not allowed to specify BLOCKSIZE or REMOTEIOSIZE with
OLDFORMAT.
```

Cause. An attempt was made to use BLOCKSIZE or REMOTEIOSIZE with the OLDFORMAT option.

Effect. BACKUP terminates.

Recovery. Reenter the command without specifying the BLOCKSIZE or REMOTEIOSIZE option.

7775

```
Not allowed to specify VERIFYTAPE/VERIFYREEL with OLDFORMAT.
```

Cause. An attempt was made to use either VERIFYTAPE or VERIFYREEL with the OLDFORMAT option.

Effect. BACKUP terminates.

Recovery. Old format tapes do not contain checksums and sequence numbers to verify. Reenter the command without specifying the VERIFYTAPE or VERIFYREEL option.

7776

```
Not allowed to specify both VERIFYTAPE and VERIFYREEL.
```

Cause. An attempt was made to use both the VERIFYTAPE and VERIFYREEL options.

Effect. BACKUP terminates.

Recovery. Reenter the command, specifying only one of the options.

7777

```
More than one tape format option specified.
```

Cause. An attempt was made to specify more than one tape format option (DP1FORMAT, DP2FORMAT, OLDFORMAT).

Effect. BACKUP terminates.

Recovery. Correct the tape option, and reenter the command.

7778

```
VERIFYTAPE is only allowed when LISTONLY is specified.
```

Cause. A file-set list was specified along with VERIFYTAPE.

Effect. RESTORE terminates.

Recovery. RESTORE has two modes. In the first mode, you do not specify a file set list, and the files on tape are listed out. In this mode, VERIFYTAPE can be specified to cause all on-tape data blocks to be read and verified. In the second mode, you can specify a file-set list without VERIFYTAPE, and all files are then restored and verification is performed.

7779

```
'(' expected.
```

Cause. An opening parenthesis “(” was expected.

Effect. BACKUP, BACKCOPY, or RESTORE terminates.

Recovery. Insert the required parenthesis, and reenter the command.

7780

```
')' expected.
```

Cause. A closing parenthesis “)” was expected.

Effect. BACKUP, BACKCOPY, or RESTORE terminates.

Recovery. Insert the required parenthesis, and reenter the command.

7781

```
Bad alternate file number.
```

Cause. An invalid alternate-file number was specified.

Effect. BACKUP or RESTORE terminates.

Recovery. Reenter the command with an alternate-file number in the range 0 through 255.

7782

```
Bad partition file number.
```

Cause. An invalid partition file number was specified.

Effect. BACKUP or RESTORE terminates.

Recovery. Reenter the command with a partition file number in the range 1 through 15.

7783

```
Bad extent size.
```

Cause. An invalid extent size was requested.

Effect. BACKUP or RESTORE terminates.

Recovery. Reenter the command with an extent size in the range 1 through 65535.

7784

```
PART parameter specified more than once for this secondary
partition.
```

Cause. The PART parameter was used more than once with the same partition number.

Effect. BACKUP or RESTORE terminates.

Recovery. Reenter the command with only one PART parameter.

7785

```
ALTFILE parameter specified more than once for this alternate
key file.
```

Cause. The ALTFILE parameter was specified more than once with the same alternate-file number.

Effect. BACKUP terminates.

Recovery. Each ALTFILE parameter must be for a different alternate file.

7786

```
Volume name has already been specified with a previous PART
parameter.
```

Cause. A volume name was specified that had already been specified in a PART parameter.

Effect. BACKUP or RESTORE terminates.

Recovery. Reenter the command without specifying the volume name a second time.

7787

```
File name has already been specified with a previous ALTFILE
parameter.
```

Cause. An attempt was made to issue a file name using the ALTFILE parameter, but the file name was already specified.

Effect. BACKUP or RESTORE terminates.

Recovery. Reissue the command, specifying a different file name in the ALTFILE parameter.

7789

```
Not allowed to specify new extent sizes with EXT or PART when
PARTONLY or PARTOF specified.
```

Cause. An attempt was made to change the extent size of an individual partition that was being converted.

Effect. BACKUP or RESTORE terminates.

Recovery. Extent sizes can be changed only when an entire structured partitioned file is being converted. Reenter the command without specifying EXT or PART.

7790

```
Partial date and time after the current date and time.
```

Cause. The *partial-dump-date* parameter of the PARTIAL parameter was not before the current date and time.

Effect. BACKUP terminates.

Recovery. Correct the partial date and time, and reenter the command.

7791

```
Must be a disk file name.
```

Cause. A name other than a disk-file name was used.

Effect. BACKUP, BACKCOPY, or RESTORE terminates.

Recovery. Reenter the command with a disk-file name.

7792

```
Too many file sets.
```

Cause. The total number of file sets specified for *fileset-list* and *not-fileset-list* exceeded the storage BACKUP allows.

Effect. BACKUP or RESTORE terminates.

Recovery. Reenter the command, specifying fewer file sets.

7793

```
START file set for RESTORE cannot have any wildcards.
```

Cause. In RESTORE, the START option must specify a fully qualified file-set name; wild-card characters are not allowed.

Effect. RESTORE fails and terminates.

Recovery. Reissue the command without wild-card characters in the START option.

7794

```
Not allowed to specify PARTOF and PART together.
```

Cause. An attempt was made to use PARTOF with the PART option.

Effect. RESTORE terminates.

Recovery. Reenter the command, specifying either PARTOF or PART.

7795

```
3 character alphabetic month name expected.
```

Cause. The month name in the BACKUP PARTIAL option is not in the required three-character alphabetic format.

Effect. BACKUP fails and terminates.

Recovery. Specify month using three characters and reissue the command. For example, JAN, FEB, MAR, AUG, SEP, DEC.

7796

```
NOT fileset list has already been specified.
```

Cause. The NOT file-set list was specified more than once in the BACKUP or RESTORE command.

Effect. BACKUP or RESTORE fails and terminates.

Recovery. Specify a file-set list for the NOT option that does not conflict with a file-set specified in another option and reissue the command.

7797

```
Only allowed to restore a single volume when PARTOF is
specified and a new volume is specified with VOL.
```

Cause. When the VOL option is used with the PARTOF option in RESTORE, only one volume can be specified in *restore-files*.

Effect. RESTORE fails and terminates.

Recovery. Reissue the command with only one volume name specified in *restore-files*.

7799

```
Volume is not accessible or does not exist.
```

Cause. An invalid volume name or an inaccessible volume was specified for the VOL option in the RESTORE command.

Effect. RESTORE fails and terminates.

Recovery. Reissue the command without the offending volume name.

7801

```
The MYID option is not allowed in this copy of RESTORE.
```

Cause. This older version of RESTORE does not support the MYID option.

Effect. RESTORE fails and terminates.

Recovery. Reissue the command without the MYID option or use a later version of RESTORE.

7802

```
Bad tape mode name.  Expecting STARTSTOP or STREAM.
```

Cause. An invalid TAPEMODE specification was used in the BACKUP command.

Effect. BACKUP fails and terminates.

Recovery. Specify either STARTSTOP or STREAM with the TAPEMODE option and reissue the BACKUP command.

7803

```
Not allowed to specify DENSITY more than once.
```

Cause. The DENSITY option was specified two or more times in the BACKUP command.

Effect. BACKUP fails and terminates.

Recovery. Specify the DENSITY option only once and reissue the command.

7807

```
Expecting either ON or OFF or nothing (the default is ON).
```

Cause. An incorrect parameter was specified for a RESTORE command option.

Effect. RESTORE fails and terminates.

Recovery. Specify either ON, OFF, or no parameter for the option and reissue the RESTORE command. Specifying no parameter is the same as specifying ON.

7808

```
Expecting either EXPLICIT or IMPLICIT.
```

Cause. An invalid parameter for the INDEXES option was specified in the BACKUP/RESTORE command.

Effect. BACKUP/RESTORE fails and terminates.

Recovery. Specify either EXPLICIT or IMPLICIT with the INDEXES option and reissue the command.

7809

```
VOL and MAP NAMES options are mutually exclusive.  Must use
only one of these.
```

Cause. The VOL and MAP NAMES options cannot be used in a single RESTORE command.

Effect. RESTORE fails and terminates.

Recovery. Specify either VOL or MAP NAMES, but not both, and reissue the command.

7810

```
Only one file renaming option can be specified.
```

Cause. The RENAME and MAP NAMES options cannot be used in a single RESTORE command

Effect. RESTORE fails and terminates.

Recovery. Specify either RENAME or MAP NAMES, but not both, and reissue the command.

7811

```
PART/ALTFILE and MAP NAMES options are mutually exclusive.
Must use only one of these.
```

Cause. The PART and MAP NAMES options cannot be used in a single RESTORE command, and the ALTFILE and MAP NAMES options cannot be used in a single RESTORE command.

Effect. RESTORE fails and terminates.

Recovery. Reissue the command without the offending option.

7812

```
Not allowed to specify PARTONLY ON and INDEXES IMPLICIT
together.
```

Cause. The PARTONLY ON and INDEXES IMPLICIT options cannot be used in a single RESTORE command.

Effect. RESTORE fails and terminates.

Recovery. Reissue the command without the offending option.

7813

```
USE CATALOG has already been specified.
```

Cause. The CATALOG option was specified more than once in a RESTORE command.

Effect. RESTORE fails and terminates.

Recovery. Specify the CATALOG option only once and reissue the command.

7814

```
START cannot be specified as a qualifier and an option.
```

Cause. START can be specified either as a qualifier or as a separate option in a BACKUP or RESTORE command, but not both.

Effect. BACKUP or RESTORE fails and terminates.

Recovery. Specify the START option only once and reissue the command.

7815

```
Expecting a complete fileset, *.*.*.
```

Cause. BACKCOPY expects a complete file set, *.*.*, to be specified.

Effect. BACKCOPY fails and terminates.

Recovery. Specify *.*.* in the BACKCOPY command and reissue the command.

7816

```
PARTONLY and MAP NAMES cannot be used together without KEEP.
```

Cause. The PARTONLY and MAP NAMES options cannot be used in a single RESTORE command unless you also use the KEEP option.

Effect. RESTORE fails and terminates.

Recovery. Reissue the command either without these two options together or with the KEEP option added.

7818

```
Expecting a SYSNN subvolume where NN is a 2 digit octal
integer.
```

Cause. SYSGEN should invoke BACKUP from a SYS*nn* subvolume.

Effect. BACKUP fails and terminates.

Recovery. Restart the SYSGEN phase with a SYS*nn* subvolume specified.

7819

```
Not allowed to specify TAPEMODE more than once.
```

Cause. The TAPEMODE option was specified more than once in a RESTORE command.

Effect. RESTORE fails and terminates.

Recovery. Specify TAPEMODE only once and reissue the command.

7820

```
Not allowed to specify more than one system.
```

Cause. BACKUP cannot process files from more than one node (system).

Effect. BACKUP fails and terminates.

Recovery. Reissue the command with only one node specified in *backup-files*.

7824

```
PARTONLY and CATALOGS cannot be used together without KEEP.
```

Cause. You cannot use the PARTONLY and CATALOGS options in a single RESTORE command unless you also use the KEEP option.

Effect. RESTORE fails and terminates.

Recovery. Reissue the command without these two options together or with the KEEP option added.

7831

```
Invalid REMOTEIOSIZE. Must be in the range 2 .. 52 and even.
```

Cause. An invalid number was entered for REMOTEIOSIZE in a BACKUP command.

Effect. BACKUP fails and terminates.

Recovery. Reissue the command specifying an even number from 2 to 52.

7832

```
PARTONLY must be set when setting SQLTAPEPARTARRAY.
```

Cause. The SQLTAPEPARTARRAY option was specified in a RESTORE command, but PARTONLY was not specified.

Effect. RESTORE fails and terminates.

Recovery. Reissue the RESTORE command including the PARTONLY option.

7833

```
The NOSQLDATA and SQLCATALOGS parameters are mutually
exclusive.
```

Cause. A BACKUP or RESTORE command was issued including both the NOSQLDATA and SQLCATALOGS option, which cannot be used in the same command.

Effect. BACKUP or RESTORE fails and terminates.

Recovery. Reissue the BACKUP or RESTORE command using only one of these options.

7834

```
The NOUNLOAD and NOREWIND parameters are mutually exclusive.
```

Cause. This error could be caused by:

- The NOUNLOAD and NOREWIND options were both specified for BACKUP.
- The NOUNLOAD and NOREWIND options were both specified for RESTORE.
- The NOUNLOADIN and NOREWINDIN options were both specified for BACKCOPY.
- The NOUNLOADOUT and NOREWINDOUT options were both specified for BACKCOPY.

Effect. The utility fails and terminates.

Recovery. Reissue the command with the correct options.

7835

```
Expecting either ON, OFF, PARTIAL, or nothing (the default is
ON).
```

Cause. An incorrect parameter in the PARTONLY option was specified in the RESTORE command.

Effect. RESTORE fails and terminates.

Recovery. Specify either ON, OFF, PARTIAL, or no parameter for the PARTONLY parameter, then reissue the RESTORE command. Specifying no parameter is the same as specifying ON.

7836

```
Invalid CATALOGFILES NUMBER. Must be in the range 1...20000
```

Cause. A RESTORE command was issued including both the NOSQLDATA and SQLCATALOGS option, which cannot be used in the same command.

An invalid CATALOGFILES NUMBER (value of *every-n-files*) was specified.

Effect. BACKUP terminates.

Recovery. Ensure that CATALOGFILES NUMBER is an integer value in the range 1 through 20,000.

7900

```
Target must be specified by a logical device number
```

Cause. In a volume-mode RESTORE operation, the parameter specified with the TARGET option must be a logical device number.

Effect. RESTORE fails and terminates.

Recovery. Specify a valid device number for the TARGET option and reissue the command.

8000

```
Tape DEFINE define is not CLASS TAPE.
```

Cause. A DEFINE was specified that was not CLASS TAPE.

Effect. BACKUP or RESTORE terminates.

Recovery. Specify a CLASS TAPE DEFINE.

8001

```
Tape DEFINE define is not LABELS BACKUP.
```

Cause. A DEFINE was specified that was not LABELS BACKUP.

Effect. BACKUP terminates.

Recovery. Specify a LABELS BACKUP DEFINE.

8002

```
Tape DEFINE define is not LABELS BACKUP or LABELS BYPASS.
```

Cause. A DEFINE was specified that was not LABELS BACKUP or LABELS BYPASS.

Effect. RESTORE terminates.

Recovery. Specify a LABELS BACKUP or LABELS BYPASS DEFINE.

8003

```
File names for the two parallel tape copies are not
different.
```

Cause. The same file name was specified for both copies of a parallel backup.

Effect. BACKUP terminates.

Recovery. Specify different file names when making a parallel backup.

8004

```
One of the two parallel copies is labeled but the other is
unlabeled.
```

Cause. A tape drive was specified as one of the copies of a parallel backup (implying unlabeled), and a labeled tape DEFINE was specified for the other copy.

Effect. BACKUP terminates.

Recovery. Specify two tape drives to make an unlabeled parallel backup or two labeled tape DEFINEs to make a labeled parallel backup.

8005

```
The SYSTEM attribute of the tape DEFINE define specifies a
system, node, that either does not exist or is inaccessible
because all paths to it are down.
```

Cause. A node was specified with a DEFINE that either does not exist or is unavailable.

Effect. BACKUP or RESTORE terminates.

Recovery. Specify a node that exists and is available.

8006

```
The DEVICE attribute of the tape DEFINE define specifies a
drive, drive, whose system either does not exist or is
inaccessible because all paths to it are down.
```

Cause. A node was specified with a DEFINE that either does not exist or is unavailable.

Effect. BACKUP or RESTORE terminates.

Recovery. Specify a node that exists and is available.

8007

```
The SYSTEM attribute of the tape DEFINE define specifies a
system, node, that is on a pre-C00 operating system.
```

Cause. A node was specified with a DEFINE that is not running a C00.00 or later RVU.

Effect. BACKUP or RESTORE terminates.

Recovery. Specify a node running on a C00.00 or later RVU.

8008

```
The DEVICE attribute of the tape DEFINE define specifies a
drive, drive, whose system is on a pre-C00 operating system.
```

Cause. A node was specified with a DEFINE that is not running a C00.00 or later version of the NonStop operating system.

Effect. BACKUP or RESTORE terminates.

Recovery. Specify a node running on a C00.00 or later version of the NonStop operating system.

8009

```
The SYSTEM attribute of the tape DEFINE define specifies a
system,node, that does not have labeled tape support
configured.
```

Cause. A node was specified with a DEFINE that does not have labeled-tape support configured.

Effect. BACKUP or RESTORE terminates.

Recovery. Specify a node that was configured with labeled-tape support.

8010

```
The DEVICE attribute of the tape DEFINE define specifies a
drive, drive, whose system does not have labeled tape support
configured.
```

Cause. A node was specified with a DEFINE that does not have labeled-tape support configured.

Effect. BACKUP or RESTORE terminates.

Recovery. Specify a node that was configured with labeled-tape support.

8011

```
The tape DEFINE define specifies labeled tape processing and
labeled tape support is not configured.
```

Cause. The local node does not have labeled-tape support configured.

Effect. BACKUP or RESTORE terminates.

Recovery. Specify a node that was configured with labeled-tape support.

8012

```
Duplicate volume ID found in the VOLUME attribute of the tape
DEFINE define.
```

Cause. A labeled tape volume ID was specified more than once in the VOLUME attribute's list of volume IDs.

Effect. BACKUP or RESTORE terminates.

Recovery. Specify a unique list of labeled tape volume IDs.

8013

```
Tape volume IDs for the two parallel copies are not unique.
```

Cause. A labeled tape volume ID was specified in both VOLUME attribute lists of a parallel backup.

Effect. BACKUP terminates.

Recovery. Specify lists of labeled tape volume IDs for the two parallel copies that are unique for the whole parallel backup operation.

8014

```
The attribute attribute was specified for the tape DEFINE
define and it is not allowed.
```

Cause. A DEFINE attribute was specified that is not allowed.

Effect. BACKUP or RESTORE terminates.

Recovery. Specify a DEFINE that has attributes from this list: DENSITY, EXPIRATION, LABELS, MOUNTMSG, OWNER, RETENTION, SYSTEM, TAPEMODE, and VOLUME.

8015

```
VOLUME SCRATCH specified for input in the tape DEFINE define.
```

Cause. SCRATCH was specified instead of a list of volume IDs.

Effect. RESTORE terminates.

Recovery. Specify a DEFINE with the list of volume IDs to use.

8016

```
Value for the tape DEFINE attribute DENSITY conflicts with
the DENSITY parameter.
```

Cause. The density value specified with the BACKUP DENSITY parameter is different than the value specified with the tape DEFINE attribute DENSITY.

Effect. BACKUP terminates.

Recovery. Either specify the same density value for the BACKUP DENSITY parameter and the tape DEFINE attribute DENSITY, or omit one of these specifications.

8025

```
Tape record is too short.  Length: rec-len.
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape.
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8026

```
Tape record length is not even (Length: rec-len).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape.
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8027

```
Tape record checksum is bad.
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8028

```
Tape record type is unknown.
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8029

```
Tape record is out of sequence. (Expected sequence
number:expected-seq-num, actual sequence number: actual-seq-
num).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8030

```
Tape record header length is bad (Record length in
header:header-rec-len, actual record length:actual-rec-len).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8031

```
Tape record length is bad (Record length: rec-len, expected
length: expected-rec-len, record type:rec-type).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8032

```
Tape record length is less than the minimum allowed (Record
length:rec-len, minimum length: min-rec-len,record type: rec-
type).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8033

```
Tape file label is bad on tape-device.
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

This error should not appear on systems using T9074 with the AEF SPR or newer versions of BACKUP and RESTORE. Instead, you should get a more specific error between 8035 and 8040.

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If not, identify the problem file. If the problem exists on another tape containing the identical file, it is probably a software problem.

If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms (including the fileset specified in BACKUP or RESTORE, and the last file successfully dumped or recovered before the failure)
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8034

```
Tape end volume label is bad.
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads

- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8035

```
Tape file label is bad: total length < minimum length on
tape-device.
```

Cause. This error is most likely a software problem in BACKUP or RESTORE, but could also be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If not, identify the problem file. If the problem exists on another tape containing the identical file, it is probably a software problem.

If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms (including the fileset specified in BACKUP or RESTORE, and the last file successfully dumped or recovered before the failure)
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8036

```
Tape file label is bad: total length > maximum length on
tape-device.
```

Cause. This error is most likely a software problem in BACKUP or RESTORE but could also be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If not, identify the problem file. If the problem exists on another tape containing the identical file, it is probably a software problem.

If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms (including the fileset specified in BACKUP or RESTORE, and the last file successfully dumped or recovered before the failure)
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8037

```
Tape file label is bad: total length is an odd number on
tape-device.
```

Cause. This error is most likely a software problem in BACKUP or RESTORE but could also be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives

- A BACKUP being stopped in the middle, resulting in an incomplete tape

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If not, identify the problem file. If the problem exists on another tape containing the identical file, it is probably a software problem.

If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms (including the fileset specified in BACKUP or RESTORE, and the last file successfully dumped or recovered before the failure)
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8038

```
Tape file label is bad: current length > total length on
tape-device.
```

Cause. This error is most likely a software problem in BACKUP or RESTORE but could also be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If not, identify the problem file. If the problem exists on another tape containing the identical file, it is probably a software problem.

If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms (including the fileset specified in BACKUP or RESTORE, and the last file successfully dumped or recovered before the failure)

- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8039

```
Tape file label is bad: first record is not full on tape-
device.
```

Cause. This error is most likely a software problem in BACKUP or RESTORE but could also be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If not, identify the problem file. If the problem exists on another tape containing the identical file, it is probably a software problem.

If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms (including the fileset specified in BACKUP or RESTORE, and the last file successfully dumped or recovered before the failure)
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8040

```
Tape file label is bad: length is negative on tape-device.
```

Cause. This error is most likely a software problem in BACKUP or RESTORE but could also be caused by:

- Corruption of the tape media
- Dirty tape drive heads

- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If not, identify the problem file. If the problem exists on another tape containing the identical file, it is probably a software problem.

If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms (including the fileset specified in BACKUP or RESTORE, and the last file successfully dumped or recovered before the failure)
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8050

```
Tape format is bad (Unexpected tape record type found.
Expected record type: expected-rec-type,actual record type:
actual-rec-type).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8051

```
Tape format is bad (File encountered is not from the same
tape set).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8052

```
Tape format is bad (File is out of sequence. Expected
sequence number: expected-seq-num, actual sequence number:
actual-seq-num).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8053

```
Tape format is bad (Variable length termination record was
found in a fixed length file).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8054

```
Tape format is bad (Actual length of the variable length file
did not match the expected value).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8055

```
Tape format is bad (Short block in the middle of the file).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8056

```
Tape format is bad (End volume label is not from this tape
set).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8057

```
Tape format is bad (File label sequence number in end volume
label is incorrect. Expected sequence number: expected-seq-
num, actual sequence number: actual-seq-num).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads

- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8058

```
Tape format is bad (File continuation mismatch in end volume
label).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8059

```
Tape format is bad (File record position mismatch in end
volume label).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8060

```
Tape format is bad (File label sequence number in begin
volume label is incorrect. Expected sequence number:
expected-seq-num, actual sequence number: actual-seq-num).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8061

```
Tape format is bad (File continuation mismatch in begin
volume label).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8062

```
Tape format is bad (File record position mismatch in begin
volume label).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8063

```
Tape format is bad (An EOF mark was found where an end volume
label was expected).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8064

```
Tape format is bad (An EOF mark was found where a file label
or an end volume label was expected).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8065

```
Tape format is bad (An EOF mark was found where a file label
continuation record was expected).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8066

```
Tape format is bad (An EOF mark was found where a data block
record was expected).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8067

```
Tape format is bad (Variable length termination record was
not on the same tape as the last data block of the file).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8068

```
Tape format is bad (An EOF mark that terminates a file was
not found).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8069

```
Tape format is bad (Two tape EOFs were not found at the end
of the tape).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8070

```
Tape format is bad (An end volume label was found where a
file label was expected).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8071

```
Tape format is bad (A file label was not found).
```

Cause. This error could be caused by:

- Corruption of the tape media
- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

8072

```
Tape format is bad (End file label length is bad).
```

Cause. This error could be caused by:

- Corruption of the tape media

- Dirty tape drive heads
- Faulty tape drives
- A BACKUP being stopped in the middle, resulting in an incomplete tape
- A software problem in BACKUP or RESTORE

Effect. BACKUP or RESTORE terminates.

Recovery. Investigate whether a corrupt tape or the tape drive is causing the problem. If you suspect that the error is due to a software problem, contact your service provider and provide all relevant information:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as EMS logs, trace files, and a processor dump

If your operation procedures require contacting the GCSC, add your system number and the numbers and versions of all related products.

8100

```
This program does not understand Archive tape format version
version (version-name).
```

Cause. The tape was made with a release of BACKUP that is incompatible with the release of RESTORE being used to restore it.

Effect. RESTORE terminates.

Recovery. Use a version of RESTORE that is compatible with *version*.

8101

```
An EOF mark was found where an Archive volume label was
expected.
```

Cause. An invalid tape format was discovered.

Effect. RESTORE rewinds and unloads the tape and prompts for a tape.

Recovery. Check the tape and remount it. If the problem persists, see the cause and recovery for error 8025 on page A-73.

8102

```
Both Archive begin volume label sets were bad.
```

Cause. Duplicate volume label sets were both bad due to parity or checksum errors.

Effect. RESTORE rewinds and unloads the tape and prompts for a tape.

Recovery. If the drive used has a manual density switch, make sure that it is set to the correct density and try again. If the problem persists, see the cause and recovery for error 8025 on page A-73.

8103

```
The tape was not positioned at the load point.
```

Cause. A back space operation detected that the tape was not positioned at the load point.

Effect. BACKUP or RESTORE rewinds and unloads the tape and prompts for a tape.

Recovery. Reload the tape. If the problem persists, have the tape drive serviced.

8106

```
Unable to recognize the tape.
```

Cause. The tape could not be recognized. This could have been caused by mounting a tape not produced by BACKUP or by parity or checksum errors or a corrupted tape.

Effect. RESTORE rewinds and unloads the tape and prompts for a tape.

Recovery. If the drive used has a manual density switch, make sure that it is set to the correct density and try again. If the problem persists, see the cause and recovery for error 8025 on page A-73.

8107

```
Found a regular labeled tape.
```

Cause. A labeled tape (but not a BACKUP labeled tape) was mounted when a tape produced by BACKUP was expected.

Effect. RESTORE rewinds and unloads the tape and prompts for the correct tape.

Recovery. Mount the correct tape.

8108

```
Found a pre-C00 TMF tape.
```

Cause. A TMF tape with a version prior to C00.00 was mounted when a tape produced by BACKUP was expected.

Effect. RESTORE rewinds and unloads the tape and prompts for the correct tape.

Recovery. Mount the correct tape.

8110

```
Tape is not an Archive tape.
```

Cause. An Archive tape was expected, but not found.

Effect. RESTORE rewinds and unloads the tape and prompts for the correct tape.

Recovery. Mount the correct tape.

8111

```
A TMF Archive tape was not expected.
```

Cause. A TMF Archive tape was mounted instead of an Archive tape produced by BACKUP.

Effect. RESTORE rewinds and unloads the tape and prompts for the correct tape.

Recovery. Mount the correct tape.

8112

```
Tape is not from the same tape set.
```

Cause. A tape from a different tape set was mounted.

Effect. RESTORE rewinds and unloads the tape and prompts for the correct tape.

Recovery. Mount the correct tape.

8113

```
Wrong tape mounted (Found tape #tape-num, expected tape
#expected-tape-num).
```

Cause. The wrong tape from a multitape tape set was mounted.

Effect. RESTORE rewinds and unloads the tape and prompts for the correct tape.

Recovery. Mount the correct tape.

8115

```
An EOF mark was not found after the Archive volume label set.
```

Cause. An invalid tape format was discovered.

Effect. RESTORE rewinds and unloads the tape and prompts for the correct tape.

Recovery. Check the tape and remount it. If the problem persists, see the cause and recovery for error 8025 on page A-73.

8116

```
Bad length for standard tape header label.
```

Cause. An invalid tape format was discovered.

Effect. RESTORE rewinds and unloads the tape and prompts for the correct tape.

Recovery. Check the tape and remount it. If the problem persists, see the cause and recovery for error 8025 on page A-73.

8117

```
Standard tape header label not found.
```

Cause. An invalid tape format was discovered.

Effect. RESTORE rewinds and unloads the tape and prompts for the correct tape.

Recovery. Check the tape and remount it. If the problem persists, see the cause and recovery for error 8025 on page A-73.

8118

```
Standard tape header label is out of sequence.
```

Cause. An invalid tape format was discovered.

Effect. RESTORE rewinds and unloads the tape and prompts for the correct tape.

Recovery. Check the tape and remount it. If the problem persists, see the cause and recovery for error 8025 on page A-73.

8119

```
File-system error n occurred on the first I/O operation
attempted.
```

Cause. The first I/O operation attempted on a tape encountered a file-system error.

Effect. BACKUP rewinds and unloads the tape and prompts for a tape.

Recovery. For corrective action for the error number *n*, see the *Guardian Procedure Errors and Messages Manual*.

8120

```
Labeled tape mounted for input is a scratch tape.
```

Cause. A tape that has been labeled or relabeled and has no data was mounted for reading.

Effect. RESTORE rewinds and unloads the tape and prompts for another tape.

Recovery. Mount the correct tape.

8121

```
Tape parity errors and/or tape record checksum errors were
found on the beginning volume labels.
```

Cause. Parity errors, checksum errors, or both occurred on the beginning volume labels. Parity and checksum errors might result because the beginning of a tape is worn out or a tape drive is dirty.

Effect. RESTORE rewinds and unloads the tape and prompts for a tape.

Recovery. If the drive used has a manual density switch, make sure that it is set to the correct density and try again. If the problem persists, see the cause and recovery for error 8025 on page A-73.

8122

```
Tape parity error(s) and/or tape record checksum error(s)
occurred on the beginning volume labels. Error recovery was
able to recognize the tape.
```

Cause. Parity or checksum errors occurred on some of the beginning volume labels. Parity and checksum errors might result because the beginning of a tape is worn out or a tape drive is dirty.

Effect. RESTORE continues. Because duplicate volume labels were written, error recovery was able to recognize the tape usage of the unaffected volume labels.

Recovery. After RESTORE is finished, investigate the tape and the tape drive.

8123

```
An EOF mark was found where begin volume labels were
expected.
```

Cause. An invalid tape format was discovered.

Effect. RESTORE rewinds and unloads the tape and prompts for a tape.

Recovery. Check the tape and remount it. If the problem persists, see the cause and recovery sections for error 8025 on page A-73.

8126

```
Attempt to open a labeled tape failed due to error number.
Evaluate the error and remount the same/different tape on the
same/different tape drive.
```

Cause. A tape mount error was discovered during a labeled tape operation.

Effect. BACKCOPY, BACKUP, or RESTORE tries to recover from the error until the operation succeeds or until a file-system error occurs. The error *number* is one of:

- 120, the data parity error, which often indicates that the tape drive needs cleaning or that the tape is unusable
- 190, undefined device error

Recovery. If the error number is 120, clean the tape drive or try mounting a different tape. If the error number is 190, mount the tape on a different tape drive.

8150

```
The data read from both parallel copies did not match.
```

Cause. The data records read from each parallel copy during tape verification were not equal. The checksums and tape sequence numbers for each record were correct, yet the data was not the same.

Effect. BACKUP terminates.

Recovery. Contact your service provider and provide:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your operation procedures require contacting the GCSC, add your system number and the numbers and versions of all related products.

8151

```
Premature end of tape.
```

Cause. The end of tape marker is very close to the beginning of tape marker or the tape drive is faulty.

Effect. BACKUP terminates.

Recovery. Use a longer tape or check the tape drive as appropriate.

8152

```
Tape density selection not supported.
```

Cause. The DENSITY parameter was specified and the tape drive does not support programmatic density selection.

Effect. BACKUP continues.

Recovery. To use this parameter, use a drive capable of programmatic density selection.

8153

```
Tape streaming selection not supported.
```

Cause. The TAPEMODE streaming parameter was specified and the tape drive does not support streaming.

Effect. BACKUP continues.

Recovery. To make use of the parameter, use a drive capable of streaming.

8154

```
During repositioning an EOF mark was not found where it was
expected.
```

Cause. While repositioning the tape, to overwrite a terminated file, an EOF mark was not found where it was expected.

Effect. BACKUP terminates.

Recovery. See the cause and recovery sections of error 8025 on page A-73.

8155

```
During repositioning an EOF mark was found where a record was
expected.
```

Cause. While repositioning the tape to overwrite a terminated file, an EOF mark was found where a record was expected.

Effect. BACKUP terminates.

Recovery. See the cause and recovery sections of error 8025 on page A-73.

8156

```
A parity error occurred on the first end volume label.
```

Cause. A parity error was detected on one of the two end volume labels. The end of the tape is particularly vulnerable to parity errors. For example, tapes with reflective strips cause a bump in the tape; sudden stopping at the end causes tape stretching; and so on. The tape format has duplicate end volume labels in hope that at least one of them will be readable.

Effect. RESTORE proceeds.

Recovery. After the restore is finished, the tape and tape drive should be investigated.

8157

```
A parity error occurred on the second end volume label.
```

Cause. A parity error was detected on one of the two end volume labels. The end of the tape is particularly vulnerable to parity errors. For example, tapes with reflective strips cause a bump in the tape; sudden stopping at the end causes tape stretching; and so on. The tape format has duplicate end volume labels in hope that at least one of them will be readable.

Effect. RESTORE proceeds.

Recovery. After the restore is finished, the tape and tape drive should be investigated.

8158

```
The labeled tape has already been used in this tape set.
```

Cause. A labeled tape already written was mounted again.

Effect. BACKUP terminates.

Recovery. Mount unique labeled tapes for the whole operation.

8159

```
The labeled tape has already been used in this tape set for
the other copy.
```

Cause. A labeled tape already written for the other copy was mounted again.

Effect. BACKUP terminates.

Recovery. Mount unique labeled tapes for the whole parallel backup operation.

8160

```
The file being aborted began on one of the earlier tapes.
That tape will have to be remounted.
```

Cause. A file already on tape that began on a previous tape is being terminated. That tape must be remounted so the next file can be written over the terminated file.

Effect. BACKUP prompts for a previous tape.

Recovery. Find the previous tape specified and mount it.

8163

```
A bad Archive standard labeled tape volume label was found.
```

Cause. Tape verification by BACKUP detected that the Archive standard labeled-tape volume label contained an error.

Effect. BACKUP terminates.

Recovery. See the cause and recovery sections of error 8025 on page A-73.

8164

```
A bad Archive tape volume label was found.
```

Cause. Tape verification by BACKUP detected that the Archive tape volume label contained an error.

Effect. BACKUP terminates.

Recovery. See the cause and recovery sections of error 8025 on page A-73.

8165

```
File-system error n occurred on operation operation.
```

Cause. A file-system error occurred on the tape drive.

Effect. BACKUP or RESTORE terminates.

Recovery. For corrective action for the error number indicated by *n*, see the *Guardian Procedure Errors and Messages Manual*.

8166

```
Unable to turn on tape buffering due to insufficient I/O
buffer space (file-system error n).
```

Cause. The tape process reported that it was unable to allocate space for buffered mode.

Effect. BACKUP and RESTORE proceed in unbuffered mode. The rate at which the tape is read or written might be lower than expected.

Recovery. If this occurs frequently, memory utilization in the tape process processor should be examined. If necessary, reconfigure the system to distribute memory utilization more evenly or add more physical memory.

8167

```
Unable to Open/Close/Write/Communicate with Media Catalog
Manager (file-system error n).
```

Cause. DSM/TC was unavailable.

Effect. BACKUP process aborts.

Recovery. Restart BACKUP or investigate why DSM/TC is unavailable.

8168

```
Unable to start TMF Transaction for Media Catalog.
```

Cause. DSM/TC was unable to start a TMF transaction.

Effect. BACKUP aborts.

Recovery. Investigate why TMF is unable to start the transaction.

8169

```
File Catalog specified does not exist.
```

Cause. DSM/TC was unable to access the file catalog.

Effect. BACKUP aborts.

Recovery. Investigate why the DSM/TC file catalog is unavailable.

8170

```
SQL error-message.
```

Cause. DSM/TC reported an SQL error.

Effect. BACKUP aborts.

Recovery. Investigate the SQL error. If the SQL error reported to BACKUP is either 8041 or -8041, use the define =_SQL_CMP_TIMEOUT. (The define should be map class. The value of the file attribute is not important. It only needs to exist to disable the SQLCOMP timeout feature.) An example of adding the define is:

```
Add define =_SQL_CMP_TIMEOUT, class map, file dummy
```

B DCOM/DSAP Messages

These error messages can appear on your terminal in response to a Disk Space Analysis Program (DSAP) or Disk Compression Program (DCOM) command.

```
COMMAND ERROR:  command
```

Cause. A syntax error was encountered in a DSAP or DCOM command.

Effect. DSAP or DCOM stops.

Recovery. Rerun the command with the corrected syntax.

```
COMPRESSION IS BEING TERMINATED
```

Cause. DCOM encountered an invalid sector on the disk during the DCOM operation.

Effect. DCOM issues the appropriate error messages and then terminates.

Recovery. Use the SCF INFO DISK, BAD command on the affected disk to obtain the bad sector address, and then perform a CONTROL DISK, SPARE command before restarting DCOM.

```
ERROR: To display disks > 18GB, use the NEWFORMAT option
```

Cause. NEWFORMAT option not specified.

Effect. DSAP returns error.

Recovery. Rerun the command with correct syntax.

```
filename CREATE ERROR:  nnn
```

Cause. DCOM tried to perform a CREATE on the file name but encountered an error.

Effect. DCOM continues its compression activities. It ignores this file, working around it, until it reaches the maximum number of extent moves specified.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

```
{ PRIMARY } DISK HAS nn UNSPARED SECTORS
{ MIRROR  }
```

Cause. DSAP or DCOM encountered at least one bad sector on the specified disk.

Effect. When DSAP encounters an invalid sector, it issues this error message and continues processing. When DCOM encounters an invalid sector, it stops after issuing the message and an additional message, “COMPRESSION IS BEING TERMINATED” (unless the IGNOREBADSECTORS option was specified).

Recovery. Not needed for DSAP. Before restarting DCOM, use the SCF INFO DISK, BAD command on the affected disk to obtain the bad sector address, then perform a CONTROL DISK, SPARE command before restarting DCOM.

```
DIRECTORY COPY ERROR:  nnn
```

Cause. DSAP or DCOM was trying to perform a directory copy.

Effect. DSAP or DCOM stops.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

```
DIRECTORY FILE NAMES NOT IN ASCENDING ORDER
-- filename-1 FOLLOWS filename-2
```

Cause. The file names in the disk directory were not in ascending order. The directory is bad.

Effect. DSAP or DCOM stops.

Recovery. The procedure for fixing the directory is complex. Contact your service provider and provide all relevant information as follows:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

```
DIRECTORY OPEN ERROR:  nnn
```

Cause. DSAP or DCOM tried to open the directory file.

Effect. DSAP or DCOM stops.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

```
DIRECTORY WORKFILE CLOSE/UNMAPPING ERROR:  nnn
```

Cause. DSAP or DCOM tried to close (or unmap) the work file.

Effect. DSAP or DCOM stops.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

```
DIRECTORY WORKFILE CREATION/OPEN/MAPPING ERROR: nnn
```

Cause. DSAP or DCOM tried to open (or create or map) the work file.

Effect. DSAP or DCOM stops.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual* for file-system errors, or the SEGMENT_ALLOCATE_ of the *Guardian Procedure Calls Reference Manual* for guidance on proper recovery.

```
ERROR:  BUFFER OVERFLOWED.
FILE SIZE DISTRIBUTION NOT BEING PRODUCED
```

Cause. DSAP was run with the FILESIZE option, and the buffer overflowed due to insufficient space.

Effect. The file-size distribution report is not created.

Recovery. None required.

```
EXCHANGE EXTENT ERROR:  nnn
```

Cause. DCOM failed to exchange extents during an extent move operation.

Effect. DCOM cancels the extent exchange but continues to run, working around the unmoved extents.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

```
EXTENT MOVE LOGIC ERROR
```

Cause. DCOM detected an error in the internal logic of the program.

Effect. DCOM cancels the most recent extent move but continues to run, working around the unmoved extents.

Recovery. None.

```
FATAL DISK ERROR:  nnn
```

Cause. DCOM detected a fatal disk error. For example, the disk was down.

Effect. DCOM stops.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

```
FINAL EXTENT DISTRIBUTION REPORT ERROR:  nnn
```

Cause. DCOM encountered an error while trying to produce the final extent distribution report.

Effect. DCOM stops.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

```
FREE SPACE CHANGED DURING DIRECTORY COPY
SPACE ALLOCATIONS ARE TOO FREQUENT
EXTENT CHECK CANNOT BE PERFORMED
```

Cause. DSAP tried to copy the disk directory, but due to high disk activity it could not get a copy without intervening changes (by other users).

Effect. DSAP terminates the space allocation. If you specified the EXTENTCHECK option, DSAP terminates.

Recovery. Perform DSAP on a busy disk at a time when disk activity is lower.

```
FREE SPACE TABLE READ ERROR:  nnn
```

Cause. DCOM detected an error while trying to read the free space table.

Effect. DCOM stops.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

```
ILLEGAL FILE NAME IN DIRECTORY FOLLOWING:  filename
```

Cause. The DSAP or DCOM disk directory had an invalid entry following *filename*. The condition can arise if the directory is bad.

Effect. DSAP stops.

Recovery. The procedure for fixing the directory is complex. Contact your service provider and provide all relevant information as follows:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

```
ILLEGAL USER
```

Cause. The DSAP USER option was specified with a user ID that does not exist.

Effect. DSAP stops.

Recovery. Rerun DSAP with a valid user specified.

```
INCONSISTENT FILE SIZE IN LABEL OF: filename
```

Cause. An invalid DP2 file label occurred, with cause unknown.

Effect. DSAP terminates.

Recovery. Contact your service provider to investigate the cause and provide all relevant information as follows:

- The information in this message
- Description of the problem and accompanying symptoms
- Supporting documentation such as Event Management Service (EMS) logs, trace files, and a processor dump, if applicable

If your local operation procedures require contacting the GCSC, supply your system number and the numbers and versions of all related products as well.

Do not attempt to access *filename* with DSAP. After the service provider investigates, FUP DUP *filename* to a new name, purge *filename*, then rename the new file to the old *filename*.

```
$volume IS NOT A LOCAL DISK VOLUME
```

Cause. The volume specified in the DSAP or DCOM command was not a local disk volume.

Effect. DSAP or DCOM stops.

Recovery. Check the syntax or use the SCF INFO DISK command to verify which volumes exist locally.

```
filename OPEN ERROR:  nnn
```

Cause. DCOM tried to perform an OPEN on the *filename* but encountered an error.

Effect. DCOM continues its compression activities. It ignores this file, working around it, until it reaches the maximum number of extent moves specified.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

```
PHYSICAL I/O ERROR:  nnn ON { MIRROR  } DISK
                            { PRIMARY }
```

Cause. DSAP or DCOM failed while trying to perform a physical I/O operation to the mirror (or primary) disk.

Effect. DSAP or DCOM stops.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

```
PLEASE ELIMINATE BAD SECTORS USING THE PUP SPARE COMMAND
```

Cause. DSAP or DCOM detected that the disk contained unspared bad sectors.

Effect. If this message appears during a DCOM run, DCOM terminates without performing any compression. If this message appears during a DSAP operation, DSAP issues the appropriate error messages and continues processing.

Recovery. It is best to spare the defective sectors. You can specify IGNOREBADSECTORS and rerun DCOM, but use this procedure only with caution.

```
READ ERROR:  nnn
```

Cause. DCOM encountered an error while doing a read during an extent move.

Effect. DCOM cancels that extent move.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

```
filename SETMODE ERROR:  nnn
```

Cause. DCOM tried to perform a SETMODE on the *filename* but encountered an error.

Effect. DCOM continues its compression activities. It ignores this file, working around it, until it reaches the maximum number of extent moves specified.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

```
THIS DISK VERSION IS NOT SUPPORTED
```

Cause. An attempt was made to run DCOM or DSAP on an unsupported disk volume version.

Effect. DCOM or DSAP stops.

Recovery. Run DCOM or DSAP on another volume.

```
THIS IS A DP1 VOLUME {DCOM | DSAP} IS BEING TERMINATED
```

Cause. An attempt was made to run DCOM or DSAP on a disk that is using DP1, which is not supported on D-series, G-series, H-series, or J-series RVUs.

Effect. DCOM or DSAP stops.

Recovery. Run DCOM or DSAP on another volume.

```
THIS PROGRAM REQUIRES A NONSTOP II SYSTEM WITH A
{ C10+ } VERSION OF THE OPERATING SYSTEM
{ D00 OR SUBSEQUENT }
{ D30 OR SUBSEQUENT }
```

Cause. An attempt was made to run DSAP or DCOM on a NonStop 1+ system or a version of the NonStop operating system earlier than C00.00.

Effect. The utility (DSAP or DCOM) specified in the run command is not started.

Recovery. You cannot use this version of DSAP or DCOM to analyze disk space use or compress files with the current operating system. If you need to compress files, see the discussion of consolidating disk free space in the system operator's guide for your system. Or upgrade your system to meet the requirements of DSAP and DCOM.

```
UNABLE TO {  OPEN  } SYSDCOM.RECOVERY FILE,
          { CREATE }
          { SECURE }
ERROR: nnn
```

Cause. DCOM encountered an error while trying to open (or create or secure) the file SYSDCOM.RECOVERY.

Effect. DCOM stops. If, however, the message is `UNABLE TO OPEN` and the *nnn* is error 12 (FILE IN USE), DCOM might already be running, preventing a second DCOM from running simultaneously. In this case, only the second DCOM stops.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

```
option:  UNRECOGNIZED OPTION
```

Cause. The specified *option* was incorrect for the DSAP or DCOM command entered.

Effect. DSAP or DCOM stops.

Recovery. Check the syntax of the command and reenter it correctly.

```
USER DETAIL REPORT ERROR:  nnn
```

Cause. An error was encountered when DSAP tried to produce the user detail report.

Effect. DSAP stops.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

```
USER SUMMARY REPORT ERROR:  nnn
```

Cause. An error was encountered when DSAP tried to produce the user summary report.

Effect. DSAP stops.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

```
WARNING: '$volume.SYSDCOM.RECOVERY' NOT CREATED BY
         DCOM
```

Cause. DCOM encountered a file named SYSDCOM.RECOVERY that it did not create. Someone else might have created a file by that name.

Effect. DCOM follows this message with a request that you purge the current file.

Recovery. Purge the current file.

```
WORK VOLUME SPECIFIED TWICE
```

Cause. WORKFILE *$volume* appeared more than once in a DSAP or DCOM command.

Effect. DSAP or DCOM stops.

Recovery. Reenter the command correctly.

```
filename WRITE ERROR:  nnn
```

Cause. DCOM tried to perform a WRITE on the *filename* but encountered an error.

Effect. DCOM continues its compression activities. It ignores this file, working around it, until it reaches the maximum number of extent moves specified.

Recovery. For corrective action for the error number indicated by *nnn*, see the *Guardian Procedure Errors and Messages Manual*.

```
ERROR: To display disks > 72 GB, use the TERAFORM option
```

Cause. The TERAFORM option is not specified.

Effect. DSAP returns an error.

Recovery. Rerun the command with the correct syntax.

C PAK/UNPAK Messages

This section describes the error messages that can arise when you use PAK or UNPAK. In addition to their own messages, PAK and UNPAK can display any BACKUP or RESTORE messages during their processing.

```
num files created
```

Cause. The -split option was used, and PAK created *num* files.

Effect. None.

Recovery. Informational message; no corrective action is necessary.

```
'after' command too long
```

Cause. The length of the parameter to the -after option is larger than 255.

Effect. The program stops with completion code 2. No archive is created.

Recovery. Use a shorter command and try again.

```
Archive does not exist
```

Cause. The archive file given in the UNPAK command does not exist.

Effect. The program stops with completion code 2. No files are extracted.

Recovery. Correct the command and try again.

```
Archive file is too small - specify a larger extent
```

Cause. PAK reaches the maximum file size while writing to the archive.

Effect. The program stops with completion code 2. The created archive file is purged.

Recovery. Resubmit the command with larger extent size (-ext option), larger maxextents (-max), or multifile option (-split) or reduce the number or size of files to be compressed.

```
Archive version: num
```

Cause. UNPAK has read the header of an archive file and reports the version of the file. The supported version numbers are 1 and 2.

Effect. The program continues.

Recovery. None. This is not an error if the version is 1 or 2.

```
BACKUP issued a READ or WRITEREAD
```

Cause. The BACKUP process issued an unexpected READ operation BACKUP should only write data to the PAK process.

Effect. The program abends. A saveabend file is possibly created. The archive is created but is probably invalid.

Recovery. This is an internal error. Report it to your service provider.

```
Comment string too long: string
```

Cause. The string specified in the -comment option is too long. The maximum length is 255 characters not including the surrounding quotes.

Effect. The program stops with completion code 2. No archive is created.

Recovery. Correct the syntax and try again.

```
Compressed bytes: num-bytes
```

Cause. PAK has created an archive. *num-bytes* is the number of compressed data written to the archive file.

Effect. None.

Recovery. Informational message; no corrective action is necessary.

```
Compression failed; error code error
```

Cause. The ZLIB library reports the error code *error* while compressing data.

Effect. The program stops with completion code 2. The created archive file is purged.

Recovery. This is an internal error. Report it to your service provider.

```
Decompression failed; error code error (address address;
count count)
```

Cause. The ZLIB library reports the error code *error* while uncompressing data. The *address* value is the offset of the compressed data in the archive and *count* is the number of bytes to be uncompressed.

Effect. The program stops with completion code 2. The archive extraction is interrupted.

Recovery. This probably indicates an invalid or corrupted archive file. Ensure that the archive has been created successfully with PAK and transferred without problems.

```
Decryption failed; code error
```

Cause. The CBCDESDECODE function returned error code *error*.

Effect. The program abends. Some files might have been extracted.

Recovery. This is an internal error. Report it to your service provider.

```
Encryption faild; code error
```

Cause. The CBCDESENCODE function returned error code *error*.

Effect. The program abends. The archive was created but is probably invalid.

Recovery. This is an internal error. Report it to your service provider.

```
Error error opening file 'filename'
```

Cause. The file-system error *error* was encountered when opening the archive file *filename*.

Effect. The program stops with completion code 2. The created archive file is purged.

Recovery. For an explanation and possible recovery, use the ERROR command or see the *Guardian Procedure Errors and Messages Manual*.

```
Error error when writing to file filename
```

Cause. The file-system error *error* was encountered when writing to archive file *filename*.

Effect. The program stops with completion code 2. The created archive file is purged.

Recovery. For an explanation and possible recovery, use the ERROR command or see the *Guardian Procedure Errors and Messages Manual*.

```
Error - 2013,Fileset not dumped (error 11)
```

Cause. There was an attempt to compress an empty file set, or the file set is not available. Error 11: The file is not in the directory or the record is not in the file or the specified tape file is not on a labeled tape.

Effect. No archive file is created

Recovery. Check if the file set is empty or already exists.

```
ERROR-3012  File aborted (Read error 12).
```

Cause. The specified file is in use.

Effect. No archive file is created.

Recovery. Close the exclusively opened file and resubmit the command.

```
Error 43 (disk is full) when writing to file filename
```

Cause. File-system error 43 was encountered when writing to archive file *filename*. Not enough disk space is available.

Effect. The program stops with completion code 2. The created archive file is purged.

Recovery. Select another disk for the archive file, free some disk space on this disk, or use a smaller extent size (-ext option).

```
Error error purging file 'filename'
```

Cause. PAK cannot purge the temporary file because of file-system error *error*. If *error* is 580, the extent size or maxextents values is too large, making the archive file size exceed 4 GB.

Effect. The program stops with completion code 2. The archive is created, but it might be invalid.

Recovery. If *error* is 580, change the extent size or maxextents so the archive file is less than 4 GB. For other error numbers, use the ERROR command. For an explanation and possible recovery, see the *Guardian Procedure Errors and Messages Manual*.

```
Executing command 'command'
```

Cause. A self-extracting file is starting a TACL process to execute the command specified with the -after option.

Effect. A temporary file is created, and a TACL process is started.

Recovery. None.

```
File creation failed: Error error, Filename filename
```

Cause. The file-system error *error* was encountered when trying to create the archive file *filename*.

Effect. The program stops with completion code 2. If the -split option was used, any archive files already created are purged.

Recovery. For an explanation and possible recovery, use the ERROR command or see the *Guardian Procedure Errors and Messages Manual*.

```
File creation failed: Error 10 (File already exists),
Filename filename
```

Cause. File-system error 10 occurred when trying to create the archive file *filename*. An archive file with the name specified in the PAK command already exists.

Effect. The program stops with completion code 2. The archive file is not created. If the -split option was given, the archive files possibly already created are purged.

Recovery. Use another file name or use the -purge option if you want to purge the old file.

```
File format not recognized
```

Cause. The archive file given in the UNPAK command does not look like a valid archive. It does not start with the letters PAK, or the header address in a self-extracting archive is larger than the file size.

Effect. The program stops with completion code 2. No files are extracted.

Recovery. Use a correct file name or re-create the archive.

```
File purge failed: Error error, Filename filename
```

Cause. The -purge option was used, and file-system error *error* was encountered when trying to purge the archive file *filename*.

Effect. The program stops with completion code 2. If the -split option was used, any archive files already created are purged.

Recovery. For an explanation and possible recovery, use ERROR command or see the *Guardian Procedure Errors and Messages Manual*.

```
FILE_CLOSE_ failed: error
```

Cause. The call to FILE_CLOSE_ failed after sending the startup message to the BACKUP or RESTORE process. *error* is the file-system error retrieved with FILE_GETINFO_.

Effect. The program abends. Possibly a saveabend file is created. No archive is created, and no files are extracted.

Recovery. This is an internal error. Report it to your service provider.

```
FILE_GETINFO_ failed: error
```

Cause. The call to FILE_GETINFO_ failed while sending the startup message to the BACKUP or RESTORE process or while reading $RECEIVE. *error* is the return value of FILE_GETINFO_.

Effect. The program abends. Possibly a saveabend file is created. No archive is created, and no files are extracted.

Recovery. This is an internal error. Report it to your service provider.

```
FILE_OPEN_ failed: error, Filename filename
```

Cause. The call to FILE_OPEN_ failed while the process was opening its program file. *error* is the return value of FILE_OPEN_.

Effect. The program stops with completion code 2. The archive is created but is invalid.

Recovery. This is an internal error. Report it to your service provider.

```
FILE_OPEN_ of $RECEIVE failed: error
```

Cause. The $RECEIVE file could not be opened. *error* is the error code returned by FILE_OPEN_.

Effect. The program ABENDs. Possibly a saveabend file is created. No archive is created, and no files are extracted.

Recovery. This is an internal error. Report it to your service provider.

```
FILE_OPEN_ of BACKUP/RESTORE process failed: error
```

Cause. The needed BACKUP or RESTORE process could not be opened. *error* is the error code returned by FILE_OPEN_.

Effect. The program abends. Possibly a saveabend file is created. No archive is created, and no files are extracted.

Recovery. This is an internal error. Report it to your service provider.

```
FILE_OPEN_ of TACL process failed; error open-error
```

Cause. The -after option was specified, but the needed TACL process could not be opened because of a FILE_OPEN_ error *open-error*.

Effect. The program abends. Possibly a saveabend file is created. No archive is created, and no files are extracted.

Recovery. This is an internal error. Report it to your service provider.

```
FILE_PURGE_ of 'archive' failed: Error error
```

Cause. The PAK option -purge was specified, but PAK could not purge the archive file. *error* is the file-system error.

Effect. The program stops with completion code 2.

Recovery. Recovery depends on the error code. For an explanation and possible recovery, use the ERROR command or see the *Guardian Procedure Errors and Messages Manual*.

```
FILENAME_DECOMPOSE_ failed: error
```

Cause. The call to FILENAME_DECOMPOSE_ failed while building the startup message for the BACKUP or RESTORE process. *error* is the return value of FILENAME_DECOMPOSE_.

Effect. The program abends. Possibly a saveabend file is created. No archive is created, and no files are extracted.

Recovery. This is an internal error. Report it to your service provider.

```
FILENAME_TO_OLDFILENAME_ failed: error
```

Cause. Cause. The call to FILENAME_TO_OLDFILENAME_ failed while building the startup message for the BACKUP or RESTORE process. *error* is the return value of FILENAME_TO_OLDFILENAME_.

Effect. Effect. The program abends. Possibly a saveabend file is created. No archive is created, and no files are extracted.

Recovery. This is an internal error. Report it to your service provider.

```
INITDES failed
```

Cause. The call to INITDES failed. This call initializes the encryption data structures.

Effect. The program stops with completion code 2. The archive is not created. No files are extracted.

Recovery. This is an internal error. Report it to your service provider.

```
Invalid archive
```

Cause. UNPAK tries to read the archive file past its EOF. The archive might be corrupted.

Effect. The program stops with completion code 2. Some files might have been extracted.

Recovery. Re-create the archive.

```
Invalid comment string: string
```

Cause. The string specified in the -comment option is invalid. For example, it should be enclosed in double quotes.

Effect. The program stops with completion code 2. No archive is created.

Recovery. Correct the syntax and try again.

```
Invalid extent size
```

Cause. The extent size given in the -ext option or the maxextents value given in the -max option is larger than 2147483647.

Effect. The program stops with completion code 2. No archive is created.

Recovery. Use a smaller extent size and try again.

```
Invalid filename: filename
```

Cause. The file name specified in the -backup or -restore option is invalid.

Effect. The program stops with completion code 2. No archive is created, and no files are extracted.

Recovery. Correct the syntax and try again.

```
Invalid option: option
```

Cause. An unknown or invalid option was specified

Effect. The program stops with completion code 2. No archive is created, and no files are extracted.

Recovery. Correct the syntax and try again.

```
Invalid self-extracting file (id='id' pos=pos)
```

Cause. The self-extracting file has an invalid header. The *id* string should be CM or AF (corresponding to options -c and -after, respectively). *pos* indicates the file position where the invalid ID was found

Effect. The program stops with completion code 2. Files are not extracted.

Recovery. Re-create the archive.

```
Invalid split size
```

Cause. The file size given in the -split option is larger than 2147483647.

Effect. The program stops with completion code 2. No archive is created.

Recovery. Use a smaller file size and try again.

```
Memory allocation failure: 1
```

Cause. The disk is full, so PAK cannot allocate memory for file buffers.

Effect. The program stops with completion code 2. The created archive files are purged. Some files might have been extracted.

Recovery. Delete items so that sufficient disk space is available, and submit the command.

```
No more files can be created
```

Cause. The -split option was used but more than the maximum number of files were needed. The maximum number is determined by the length of the base name of the archive. That is, if the base name length is 7 characters, only 10 files can be created. A similar error is generated during UNPAKing an archive if the generated files have been renamed.

Effect. The program stops with completion code 2. The created archive files are purged.

Recovery. Use a shorter name for the archive or use a larger split size.

```
PROCESS_CREATE_ failed: pc-err,detail-err (program progname)
```

Cause. The needed BACKUP or RESTORE process could not start because of a PROCESS_CREATE_ error *pc-err*. *detail-err* is the detailed error code. *progname* is the program filename.

Effect. The program stops with completion code 2. No archive is created, and no files are extracted.

Recovery. If appropriate, specify a correct -backup or -restore option and resubmit the command. Code 1,48 indicates a security violation. You might need to log on as a different user ID. For an explanation of and possible recovery from other process creation errors, use the ERROR command or see the *Guardian Procedure Errors and Messages Manual*.

```
PROCESS_GETINFO_ failed: Error error
```

Cause. The call to PROCESS_GETINFO_ failed while the process is retrieving its program file name. *error* is the return value of PROCESS_GETINFO_.

Effect. The program stops with completion code 2. The created archive file is purged.

Recovery. This is an internal error. Report it to your service provider.

```
READUPDATEX failed: error
```

Cause. The call to READUPDATEX failed while reading $RECEIVE. *error* is the return value of FILE_GETINFO_.

Effect. The program abends. Possibly a saveabend file is created.

Recovery. This is an internal error. Report it to your service provider.

```
RESTORE issued a WRITE
```

Cause. The RESTORE process issued an unexpected WRITE operation. RESTORE should only read data from the UNPAK process.

Effect. The program abends. Possibly a saveabend file is created. The archive extraction is interrupted.

Recovery. This is an internal error. Report it to your service provider.

```
TACL creation failed: error pc-err,detail-err
```

Cause. The -after option was specified, but the needed TACL process could not start because of a PROCESS_CREATE_ error *pc-err*. *detail-err* is the detailed error code.

Effect. The program abends. Possibly a saveabend file is created. No archive is created, and no files are extracted.

Recovery. Code 1,48 indicates a security violation. You might need to log on as a different user ID. For an explanation of and possible recovery from other process creation errors, see the *Guardian Procedure Errors and Messages Manual*.

```
This is an encrypted file - please supply a password
```

Cause. The archive file is encrypted, but the -password option was not given in the UNPAK command.

Effect. The program stops with completion code 2. No files are extracted.

Recovery. Supply the correct password and try again.

```
This is a multi-file archive; filesize=filesize,
totalbytes=totalsize
```

Cause. UNPAK read the header of an archive file and determined that this is a multi-file archive. The maximum size of each file is *filesize* bytes. The total size of all files is *totalsize* bytes.

Effect. The program continues.

Recovery. Informational message; no corrective action is necessary.

```
This is not an encrypted archive
```

Cause. The -password option was given in the UNPAK command, but the archive file is not an encrypted archive. It does not start with the letters PAKX.

Effect. The program stops with completion code 2. No files are extracted.

Recovery. Use a correct file name, remove the password option, or re-create the archive.

```
Total bytes: num-bytes
```

Cause. PAK created an archive. *num-bytes* is the number of bytes of data received from the BACKUP process.

Effect. None.

Recovery. Informational message; no corrective action is necessary.

```
Unable to create temporary file 'filename' (error error)
```

Cause. The -after or @*infile* option was specified, but the needed temporary file could not be created. *error* is the value of the *errno* variable after calling function fopen().

Effect. The program abends. Possibly a saveabend file is created. No archive is created, and no files are extracted.

Recovery. If the default volume is not accessible, issue a VOLUME command to an accessible volume and resubmit the command after changing the appropriate volume references. For an explanation of the error code and possible recovery, use the ERROR command or see the *Guardian Procedure Errors and Messages Manual*.

```
Unable to create temporary file name (probably default volume
does not exist)
```

Cause. The -after option was specified but the needed temporary file name could not be created with the tmpnam() function.

Effect. The program abends. Possibly a saveabend file is created. No archive is created, no files are extracted.

Recovery. If the default volume is not accessible, issue a VOLUME command to an accessible volume and resubmit the command after changing the appropriate volume references.

```
Unable to open infile 'filename' (error error)
```

Cause. The @*filename* syntax was used, but the specified file could not be opened. *error* is the value of the *errno* variable after calling function fopen().

Effect. The program stops with completion code 2. No archive is created, and no files are extracted.

Recovery. If the error code is 4002, the specified file does not exist. Correct the name and resubmit the command. For explanation and possible recovery for other errors, use the ERROR command or see the *Guardian Procedure Errors and Messages Manual*.

```
Unknown CONTROL num
```

Cause. The BACKUP or RESTORE process sent an unexpected CONTROL command. *num* is the command code.

Effect. The program continues normally.

Recovery. None if there are no other problems. Report this to your service provider.

```
Unknown SETMODE num
```

Cause. The BACKUP or RESTORE process sent an unexpected SETMODE command. *num* is the command code. Using the BACKUP option DENSITY in PAK results in the message "Unknown SETMODE 66".

Effect. The program continues normally.

Recovery. None if there are no other problems. Report this to your service provider if this is not caused by using the DENSITY parameter.

```
Unsupported encryption method
```

Cause. The -password option was given in the UNPAK command but the encryption method code used in the archive file is not supported. The only supported method is 1.

Effect. The program stops with completion code 2. No files are extracted.

Recovery. Use a correct file name, remove the password option, or re-create the archive.

```
Unsupported version
```

Cause. The -password option was given in the UNPAK command, but the encryption version used in the archive file is not supported. The only supported version is 1.

Effect. The program stops with completion code 2. No files are extracted.

Recovery. Use a correct file name, remove -password, or re-create the archive.

```
Warning: file might be corrupt
```

Cause. UNPAK tries to read the archive but notices one of these conditions:

- The length of the text stored with the -c, -after, or -comment option is longer than 255 characters.
- The size of the archive stored in the file header is not equal to the actual EOF of the file. The PAK operation was interrupted when the archive was created or the file was otherwise truncated.

Effect. The program continues but might fail later.

Recovery. None if the operation succeeds. Otherwise re-create the archive.

```
Warning: file might be corrupt (expected EOF = num)
```

Cause. UNPAK tries to read the archive, but the size of the archive stored in the file header is not equal to the actual EOF of the file. The PAK operation was interrupted when the archive was created or the file was otherwise truncated.

Effect. The program continues but might fail later.

Recovery. None if the operation succeeds. Otherwise re-create the archive.

```
WRITEX failed: error
```

Cause. The call to WRITEX failed while sending the startup message to the BACKUP or RESTORE process. *error* is the file-system error retrieved with FILE_GETINFO_.

Effect. The program abends. Possibly a saveabend file is created. No archive is created, and no files are extracted.

Recovery. This is an internal error. Report it to your service provider.

```
Wrong password
```

Cause. The archive file is encrypted, but the password in -password is wrong.

Effect. The program stops with completion code 2. No files are extracted.

Recovery. Supply the correct password and try again.

```
Output file exceeded its limit, cannot write in Output file
further
```

Cause. The OUT option is specified in PAK/UNPAK command and the output file reaches the maximum file limit while writing log records.

Effect. The program stops with completion code 2. The created archive file is purged or unpacking operation is stopped.

Recovery. Resubmit the command by splitting the file set into subsets such that all files in the subset can be logged in one output file or run PAK/UNPAK command without OUT option.

D Syntax Summaries

BACKCOPY Syntax

```
[[[\node.]$volume.]subvolume.]BACKCOPY

[ / run-option [ , run-option ] ... /                    ]

   source-tape, dest-tape, *.*.*

   [ , ARCHIVEFORMAT                                     ]
   [ , CATALOGFILES every-n-files                        ]
   [ , { DENSITY density | TAPEMODE tapemode }           ]
   [ , LISTALL                                           ]
   [ , { NOREWINDIN | NOUNLOADIN }                       ]
   [ , { NOREWINDOUT | NOUNLOADOUT }                     ]
   [ , PAGELENGTH number                                 ]
   [ , VERIFYREEL                                        ]
```

BACKUP File-Mode Syntax

```
[[[\node.]$volume.]subvolume.]BACKUP [ / run-options /]

{  tape-device-name                                      }
{  ( tape-device-name1, tape-device-name2, ...) }

  , backup-files

  [ , ALTFILE ( key-file-num,
              [$volume.[subvolume.]file-id ) ]
  [ , ARCHIVEFORMAT ]
  [ , AUDITED ]
  [ , BLOCKSIZE data-record-size ]
  [ , CATALOGFILES every-n-files ]
  [ , { DENSITY density | TAPEMODE tapemode } ]
  [ , DP1FORMAT ]
  [ , DP2FORMAT ]
  [ , DSLACK percentage ]
  [ , EXT { extent-size                                  }
          { ( pri-extent-size , sec-extent-size ) }
  [ , IGNORE ]
  [ , INDEXES [ IMPLICIT | EXPLICIT ] ]
  [ , ISLACK percentage ]
  [ , LISTALL ]
  [ , MSGONLOCK ]
  [ , MULTIDRIVE ]
  [ , NEEDBOTH ]
  [ , NOMYID ]
  [ , NOPROMPT ]
  [ , NOPURGEUNTIL ]
  [ , { NOREWIND | NOUNLOAD }                          ]
  [ , NOSAFEGUARD ]
  [ , NOSQLDATA ]
```

```
  [ , NOT not-fileset-list ]
    [ , OPEN ]
  [ , PAGELENGTH number ]
  [ , PART ( sec-part-num , [ \node.][$volume]
       [ , pri-extent-size, [ sec-extent-size ] ] )
  [ , PARTIAL partial-dump-date ]
  [ , PARTONLY [ ON | OFF ] ]
  [ , REMOTEIOSIZE ]
  [ , SCRATCHVOL $volume ]
  [ , SHAREOPEN ]
  [ , SQLCATALOGS [ ON | OFF ] ]
  [ , START [ $volume.] [ subvolume.] file-id ]
  [ , TAPEMODE [ STARTSTOP | STREAM ] ]
  [ , VERIFYREEL ]
  [ , VERIFYTAPE ]
  [ , VOL [ $new-vol.] new-subvol ]
```

backup-files is:

```
{ fileset }
{ fileset-list }
{ qualified-fileset-list }
```

fileset is:

```
[[[ \node.]$volume.]subvolume.]file-id
```

fileset-list is:

```
( fileset [ , fileset ] ... )
```

Qualified File-Set Lists

qualified-fileset-list is:

```
{list-element}{( list-element [ , list-element ] ... )}
```

list-element is:

```
fileset-list [ qualifier [ qualifier ] ... ]
```

qualifier

```
   [ EXCLUDE fileset-list          ]
   [ FROM CATALOG[S] catalog-list  ]
   [ START filename                ]
   [ WHERE expression              ]
```

catalog-list is:

```
{ catalog-name                              }
{ ( catalog-name [ , catalog-name ] ... ) }
```

expression is one or more of the following:

```
expression
NOT expression
(expression AND expression)
(expression OR expression)
```

Possible expressions are:

```
   OWNER = user-id
   timestamp-field  conditional  time-value
   FILECODE conditional number
   EOF conditional number
   file-attribute
```

```
OWNER = user-id
```

user-id is:

```
   { group-name.user-name       }
   { group-name.*               }
   { group-number, user-number  }
   { group-number,*             }
```

```
timestamp-field  conditional  time-value
```

timestamp-field is any one of:

```
   CREATIONTIME
   EXPIRATIONTIME
   LASTOPENTIME
   MODTIME
```

conditional is:

```
   <
   >
   BEFORE
   AFTER
```

time-value is:

```
   day [ time ] | [ day ] time
```

day is specified as:

```
   dd mmm yyyy | mmm dd yyyy
```

where *dd* (day) is an integer from 1 through 31; *mmm* (month) is one of:

```
   JAN, FEB, MAR, APR, MAY, JUN,
   JUL, AUG, SEP, OCT, NOV, DEC,
```

and *yyyy* (year) is a 4-digit integer from 1900 through 2999.

The default for *day* is today's date.

time is specified as:

```
hh:mm[:ss ]
```

where *hh* (hour) is an integer from 0 through 23; *mm* (minute) and *ss* (second) are 2-digit integers from 00 through 59. The default *time* is 0:00:00; that is, midnight.

FILECODE *conditional number*

conditional is:

```
<
>
<=
=
>=
<> (not equal to)
```

EOF *conditional number*

conditional is:

```
<
>
<=
=
>=
<> (not equal to)
```

file-attribute is one of:

```
AUDITED
BROKEN
CORRUPT
CRASHOPEN
ENSCRIBE
ENTRYSEQUENCED
INDEX
KEYSEQUENCED
LICENSED
OPEN
[ PRIMARY | SECONDARY ] PARTITION
PROGID
RELATIVE
ROLLFORWARDNEEDED
SAFEGUARD
SQL
SQLPROGRAM
TABLE
TRUSTED
TRUSTME
TRUSTSHARED
UNSTRUCTURED
```

```
[ SHORTHAND | PROTECTION ] VIEW
```

Note. TRUSTED, TRUSTME, and TRUSTSHARED attributes are supported only on systems running H-series RVUs or J-series RVUs.

BACKUP Volume-Mode Syntax

```
[[[\node.]$volume.]subvolume.]BACKUP [ / run-options / ]

{  tape-device-name                                  }
{  ( tape-device-name1, tape-device-name2, ... )  }

  , VOLUMEMODE, { $volume | $ldev } [ -P | -M ]

  [ , BLOCKSIZE data-record-size ]
  [ , { DENSITY density | TAPEMODE tapemode } ]
  [ , LISTALL ]
  [ , MULTIDRIVE ]
  [ , NOPROMPT ]
  [ , { NOREWIND | NOUNLOAD }                ]
  [ , TAPEMODE tapemode ]
  [ , VERIFYTAPE ]
  [ , WHOLEDISC ]
```

DCOM Syntax

```
[run ] [\node.]DCOM /run-options / [$volume | HELP ][, options ]

  [ , IGNOREBADSECTORS   ]
  [ , MAXMOVES n         ]
  [ , VERIFY             ]
  [ , WORKFILE volume    ]
```

DSAP Syntax

```
[ \node.]DSAP [ / run-options / [, run-option ]... / ]

  [ $volume-specification | HELP ] [ , options ] ...
```

$*volume-specification*

$*volume specification* takes the form:

$*volume*

for reports on a single volume

($*volume1*, $*volume2*,...)

for reports on a list of volumes

```
$* | *
```

for reports on all volumes

options

can be a combination of:

```
EXTENTCHECK
WORKFILE { $volume | filename }
report-options
detail selection options
```

report-options

```
ANALYSIS
BYSUBVOL
BYUSER
DETAIL
FILESIZE
FILESPACE
FREESPACE
SHORT
SPACE
SUMMARY
```

detail-selection-options

```
AGE { [ OVER ] number | UNDER number }
AUDITED
BROKEN
CRASHOPENED
DEALLOC number
ENSCRIBE
EXPIRED
LICENSED
OPENED { [ OVER ] number | UNDER number }
PARTITIONED
PROGID
ROLLFORWARD[NEEDED]
SEPARATE
SHOWNAMEMAP
SIZE { [ OVER ] number | UNDER number }
SQL
TEMPORARY
UNEXPIRED
UNUSED number
USER [ group-name.user-name           ]
     [ group-number , user-number     ]
     [ -1                             ]
     [ group-name.*                   ]
     [ group-number , *               ]
```

PAK/UNPAK Syntax

```
PAK [run-option] archive, volumemode, fileset-list, backup-
option
```

or

```
PAK [run-option] archive, volumemode, @infile, backup-option
```

run-option

```
-i
-s
-c "params"
-after "commmand"
-ext nn
-max nn
-purge
-backup backup-program
-password password
-split nnn
-nocompress
-comment "text"
```

```
UNPAK archive,fileset [,restore-options...]
```

or

```
RUN archive [ [,] fileset [,restore-options...] ]
```

RESTORE File-Mode Syntax

```
RESTORE [ / run-option [, run-option ]... / ]

{  tape-device-name                                  }
{  ( tape-device-name1, tape-device-name2, ...) }

  , restore-files
      [ , file-mode-restore-option ]...

 [ , ALTFILE ( key-file-number ,
             [$volume.]subvolume.file-id ) ]
 [ , AUDITED [ , TURNOFFAUDIT ] ]
 [ , AUTOCREATECATALOG [ ON | OFF ] ]
 [ , CATALOG[S] ( catalog-name FOR fileset-list
     [ , catalog-name FOR fileset-list ] ...) ]
 [ , COLLATION[S] ( mapping-rule [, mapping-rule ...)
 [ , DETAIL ]
 [ , DSLACK percentage ]
 [ , EXT { extent-size                              }
         { ( pri-extent-size , sec-extent-size ) } ]
 [ , IGNORE ]
 [ , INDEXES [ IMPLICIT | EXPLICIT ] ]
 [ , ISLACK percentage ]
 [ , KEEP ]
 [ , LISTALL ]
 [ , LISTONLY ]
 [ , MAP NAME[S] ( old-fileset-list TO new-fileset-list
     [ , old-fileset-list TO new-fileset-list ]...) ]
 [ , MULTIDRIVE ]
 [ , MYID ]
 [ , NOPROMPT ]
 [ , { NOREWIND | NOUNLOAD }                  ]
 [ , NOSAFEGUARD ]
 [ , NOSQLDATA ]
 [ , NOT not-fileset-list ]
 [ , OPEN ]
 [ , PAGELENGTH number ]
 [ , PART ( sec-partition-num , [ [ \node.$volume ]
       [ , pri-extent-size , [ sec-extent-size ] ] ) ]
 [ , PARTOF volume-spec ]
 [ , PARTONLY [ ON | OFF | PARTIAL ] ]
 [ , PHYSVOL ]
 [ , REBUILD ]
 [ , REGISTERONLY [ ON | OFF ] ]
 [ , REMOTEIOSIZE ]
 [ , SCRATCHVOL $volume ]
 [ , SQLCATALOGS [ ON | OFF ] ]
 [ , SQLCOMPILE [ ON | OFF ] ]
 [ , SQLTAPEPARTARRAY ]
 [ , START [ $volume.] [ subvolume.] file-id ]
 [ , TAPEDATE [ ON | OFF ]
 [ , TURNOFFAUDIT ]
 [ , VERIFY ]
```

```
 [ , VERIFYTAPE ]
 [ , VOL [\node.][$new-vol.]new-subvol |
         [\node.]$new-vol.[new-subvol] ]

restore-files is:

   { fileset }
   { fileset-list }
   { qualified-fileset-list }

   fileset is:

   [[[ \node.]$volume.]subvolume.]file-id

   fileset-list is:

   ( fileset [ , fileset ] ... )
```

For a syntax summary, see Qualified File-Set Lists on page D-2.

RESTORE Volume-Mode Syntax

```
RESTORE [ / run-option  [, run-option ]... / ]

{  tape-device-name                                  }
{  ( tape-device-name1, tape-device-name2, ... )  }

 , VOLUMEMODE, { $volume | * }
 [ , volume-mode-restore-option ]

 [ , LISTONLY ]
 [ , MULTIDRIVE ]
 [ , NOPROMPT ]
 [ , { NOREWIND | NOUNLOAD }                ]
 [ , RENAME $volume ]
 [ , TARGET $ldev [ -P | -M ] ]
 [ , VERIFYTAPE ]
```

E CLASS TAPE DEFINEs

This appendix lists the attributes that can be used in a DEFINE of CLASS TAPE for labeled-tape processing. For more information about creating DEFINEs, see the *TACL Reference Manual* and the *Guardian User's Guide.*

Note. You can also use CLASS TAPECATALOG DEFINEs to request scratch tapes for new tape files and cataloging of new tape files, and to access cataloged tape files. For more information, see the *DSM/Tape Catalog User's Guide*.

TAPE DEFINE Attributes

The TAPE DEFINE attributes are listed in alphabetical order and their syntax is provided. For the rules that apply to each of the TAPE DEFINE attributes, see Table E-1, TAPE DEFINE Consistency Rules, on page E-7.

`BLOCKLEN block-length`

specifies the data block size in bytes. If the RECFORM attribute is F, BLOCKLEN must be a multiple of the RECLEN attribute value.

BLOCKLEN is valid but ignored by Backup and Restore 2.0. The internally used value of BLOCKLEN will always be the default or user-specified value of the BLOCKSIZE job option.

`CLASS` *`attribute`*

works as a DEFINE subtype. You must specify TAPE as the attribute of CLASS to use the DEFINE attributes for labeled-tape processing. The CLASS TAPE working attribute set consists of all the attributes discussed in this appendix.

`DENSITY { 800 | 1600 | 6250 }`

specifies the tape density in bits per inch (bpi). The specified value appears in the mount messages sent to the operator. The default is the current setting of the tape drive. For guidelines on setting tape density, see the *SCF Reference Manual for the Storage Subsystem*.

DENSITY is not supported by Backup and Restore 2.0.

`DEVICE { $`*`device`* `| \`*`remote-node`*`.$`*`device`* `}`

specifies the name of the tape device where the tapes are mounted. If you specify a tape unit on a remote node, the remote node must be a node on your system network. If you name a specific device, the $ZSVR tape process does not support queuing when more than one process opens that tape drive. To avoid queuing problems, omit specific device names.

If you do not name a specific device, the $ZSVR tape process chooses an appropriate tape drive.

If the DEFINE does not include DEVICE or SYSTEM, tapes must be mounted on a local node. You cannot specify both the DEVICE and SYSTEM attributes.

`EBCDIC { IN | OUT | ON | OFF }`

specifies whether data on an IBM labeled tape is translated.

IN—data records read from the tape file are translated from EBCDIC to ASCII.

OUT—data records written to tape are translated from ASCII to EBCDIC.

ON—both IN and OUT; this is the default for IBM tapes.

OFF—data records are not translated.

EBCDIC is not supported by Backup and Restore 2.0.

`EXPIRATION` *`date`*

specifies the date that the files on the labeled tape can be overwritten. Specify the month, day, and year (for example, DEC 31 1992). If you specify EXPIRATION, you cannot specify the RETENTION attribute. If the DEFINE does not include EXPIRATION or RETENTION, the tape expiration date defaults to the following day (RETENTION = 1 is added to the DEFINE).

This definition applies only to BACKUP. Other applications, such as FUP, default to RETENTION = 0, and the tape is considered “scratch” as soon as it is written.

`FILEID` *`tape-file-name`*

specifies the name of the tape file. The *`tape-file-name`* can be as many as 17 characters long. If names are over 17 characters, only the right-most 17 characters are used.

`FILESECT` *`volume-order`*

specifies the position of this tape volume in a multivolume file, which is being created at the same time. FILESECT is meaningful only with files that require multiple volumes. The first section of a file that is written always has a FILESECT value of 1. If the file requires a second tape volume, the section of the file written to the second tape has a FILESECT value of 2. Sequentially increasing values for FILESECT are given to each file section assigned to separate tape volumes. Do not specify FILESECT for USE OUT. The tape process defaults FILESECT to a value of 1 and adjusts the value as necessary.

`FILESEQ` *`file-order`*

specifies the position of this tape file in a multifile volume. Specify an integer in the range of 1 through 9999 to indicate the relative position of the file. The number is always 1 for single-file organization, and it is always increasing sequentially for multiple files in a volume set.

GEN *gen-number*

indicates that this file is part of a generation group. Specify an integer in the range of 1 through 32767 to indicate the generation number. The default is 1.

LABELS { ANSI | IBM | OMITTED | BYPASS | BACKUP | IBMBACKUP }

specifies the processing mode to be used. OMITTED is the default.

ANSI—the tape labels conform to ANSI standard X3.27-1987. The system is to perform ANSI standard label processing on the file (LP mode).

IBM—the tape labels conform to IBM-MVS XA release 1.2 standard. The system is to perform IBM-MVS standard label processing on the file (LP mode).

OMITTED—there are no tape labels. The system does not do label processing except to check that the tape is not a standard labeled tape (NL mode). If you specify LABELS OMITTED, you must also include the DEVICE attribute.

BYPASS—the system does not perform label processing and ignores any tape label (BLP mode). If you specify LABELS BYPASS, you must also include the DEVICE attribute. Incorrect use of BYPASS can cause labeled tapes to be overwritten.

BACKUP—used for a BACKUP or RESTORE process.

IBMBACKUP—used for a BACKUP or RESTORE process.

Valid values are BACKUP or OMITTED for Backup and Restore 2.0.

LOGICAL { 1 | 2 }

identifies which copy of a volume set created from a parallel BACKUP or BACKCOPY run contains a tape file from which you want to read data. When such tape volume sets are created, one set is cataloged as logical copy 1 and the other set as logical copy 2. This attribute is typically used for input, not output. The default is 1. If CATALOG is OFF, LOGICAL is ignored.

This option is normally used for USE IN DEFINEs only. The BACKUP and BACKCOPY utilities manage the setting for USE OUT. BACKCOPY is a standard Backup Restore utility.

LOGICAL is ignored by Backup and Restore 2.0.

MOUNTMSG *text*

specifies an additional message to be displayed with the system mount message or drive-usage request sent to the operator when this DEFINE is opened. Specify a quoted character string of up to 80 characters. Include information such as length and urgency of the tape job. The MEDIACOM MOUNTS command also shows the MOUNTMSG for each mount request it displays.

OWNER *owner-id*

identifies an owner in the label. For *owner-id* specify any unique name or code. For ANSI labeled tapes or BACKUP labeled tapes, the owner ID can be 1 through 14 characters long. For IBM or IBMBACKUP labeled tapes, the owner ID can be from 1 through 10 characters long.

PHYSICAL { *number* }

identifies a backup volume set created by BACKCOPY that contains a file to which you want to read or write data. BACKCOPY is a standard Backup Restore utility.

number is the physical copy number of the volume set you want and is specified in the range 1 through 99. The default is 1.

If you try to open the same volume set for both USE IN and USE OUT on the same system, $ZSVR manages the physical copy number of the USE OUT DEFINE for you. If the USE OUT DEFINE designates a different system than the USE IN DEFINE, you must specify a PHYSICAL number greater than the USE IN tape file, or an error occurs.

The new duplicated copy has a higher physical copy number than the original, unless BACKCOPY uses a different FILEID or GEN for the USE OUT DEFINE. Two identical volume sets created by a parallel BACKUP or a parallel BACKCOPY have the same physical copy number but different logical copy numbers (1 and 2). If CATALOG is OFF, PHYSICAL is ignored.

PHYSICAL is ignored by Backup and Restore 2.0.

RECFORM { F | U }

specifies the record format.

F—indicates fixed-length records. If you specify RECFORM F, you must specify a BLOCKLEN value that is a multiple of the RECLEN value.

U—indicates undefined length (the default for IBM tapes).

Additional guidelines:

- BLOCKLEN, RECFORM, and RECLEN values are not checked for consistency in the input files.
- For an ANSI-standard labeled tape, these defaults are assumed: RECFORM is U; RECLEN is 0; and BLOCKLEN is as configured for the device. If you specify BLOCKLEN, RECFORM, and RECLEN for a labeled tape output file, label fields are generated with those values.
- RECFORM is valid but ignored by Backup and Restore 2.0. The record format used internally will always be FIXED.

RECLEN *record-length*

specifies the record length of the tape file. For ANSI-standard tapes with fixed-length records (RECFORM is F), the default RECLEN value is as configured for the device by SYSGEN. For tapes with undefined record lengths (RECFORM is U), the default RECLEN value is 0 (zero).

RECLEN is valid but ignored by Backup and Restore 2.0. The record length used internally will always be the user-specified or default value of the BLOCKSIZE job option.

REELS *volumes*

specifies the number of volumes (reels or cartridges) in a multivolume file. Specify an integer in the range of 1 through 255.

RETENTION *days*

specifies the retention period for this tape. For *days* specify an integer. This value is translated to an expiration date when the labels are written on the tape. An expiration date prevents overwriting tape contents. The default is 1 (one); that is, the tape expires the day after being written. If you specify RETENTION, you cannot specify the EXPIRATION attribute. If the DEFINE does not include RETENTION or EXPIRATION, the tape expires immediately.

SYSTEM { *local-node* | *remote-node* }

identifies the node to which the tape drive is attached. This is where the tapes must be mounted. If you specify a remote node, it must be a node on your system network. If the DEFINE does not include DEVICE or SYSTEM, all tapes must be mounted on a local node. You cannot specify both the SYSTEM and DEVICE attributes.

TAPEMODE { STARTSTOP | STREAM }

specifies the operating mode for a cartridge tape drive such as the 5120. STARTSTOP is the default. For other than cartridge drives, this attribute is ignored.

TAPEMODE is valid but ignored by Backup and Restore 2.0. The internally used value of TAPEMODE will always be STREAM.

USE { IN | OUT | EXTEND | OPENFLAG }

specifies how the tape file is to be used.

IN—the file is to be read from tape.

OUT—the file is to be written to tape.

EXTEND—data is to be appended to the tape file.

OPENFLAG—uses the type of access indicated by the access flag of the OPEN call (must be either read or write; read/write becomes write).

If you specify USE IN or USE EXTEND, you must include the VOLUME attribute and specify LABELS ANSI or LABELS IBM.

Valid values are IN, OUT, and OPENFLAG for Backup and Restore 2.0.

`VERSION` *`number`*

indicates a version within one generation. Specify an integer in the range of 0 through 99. The default is 0.

`VOLUME { ` *`volume-id`* ` | SCRATCH }`

specifies one or more tape volume IDs or indicates that any scratch tape is acceptable for label processing. The maximum limit on the number of tape volumes is 61.

Specify *`volume-id`* as a unique 1 through 6 byte identification code. For multiple volumes, enclose the list of volume IDs in parentheses and separate them with commas.

If you specify VOLUME, you must also specify LABELS ANSI, LABELS IBM, LABELS BACKUP, or LABELS IBMBACKUP. If you specify VOLUME SCRATCH, you cannot specify USE IN or USE EXTEND. If you specify USE IN, you must include *`volume-id`* with the VOLUME attribute; otherwise, VOLUME defaults to SCRATCH.

These attributes have corresponding fields in the tape system labels:

```
BLOCKLEN      OWNER
EXPIRATION    RECFORM
FILEID        RECLEN
FILESECT      RETENTION
FILESEQ       VERSION
GEN
```

For more information, see Appendix F, Tape Label Formats.

TAPE DEFINE Consistency Checks

To display the current TAPE DEFINE attribute values, use the SHOW DEFINE command. This command also checks these attributes for consistency and returns the number of the first consistency check that fails. For more information about using the SHOW DEFINE command, see the *TACL Reference Manual*.

Table E-1. TAPE DEFINE Consistency Rules

Check Number	Description
1	You can specify either RETENTION or EXPIRATION but not both.
2	If you specify USE IN or USE EXTEND, you must include VOLUME *volume-id*.
3	If you specify VOLUME *volume-id*, you must also specify LABELS ANSI, LABELS BACKUP, LABELS IBMBACKUP, or LABELS IBM.
4	If you specify LABELS ANSI, you cannot specify EBCDIC and vice versa.
5	If you specify RECFORM F, you must specify a BLOCKLEN that is a multiple of RECLEN.
6	If you specify DEVICE, you cannot specify SYSTEM and vice versa.
7	If you specify LABELS BYPASS or LABELS OMITTED, then DEVICE is required, and these attributes might not be specified: BLOCKLEN, EBCDIC, EXPIRATION, FILEID, FILESECT, FILESEQ, GEN, OWNER, RECFORM, RECLEN, REELS, RETENTION, SYSTEM, USE, VERSION, and VOLUME.
8	If you specify VOLUME SCRATCH, then USE IN or USE EXTEND is not allowed.
9	If you specify LABELS IBM or LABELS IBM BACKUP, you must also specify FILEID and RECFORM.
10	If you specify RECLEN but not BLOCKLEN, you must specify a RECLEN value greater than zero.
11	If you specify LABELS IBMBACKUP, the DEVICE or SYSTEM attribute must specify a system running version C20 or later of the NonStop operating system.

F Tape Label Formats

This appendix lists the standards for the two tape label formats (ANSI version-3 and IBM-MVS) that HP supports. In these tables, all fields of the standards are documented. However, HP does not support all fields within these standards. For example, IBM-specific fields such as VTOC Pointer are not supported.

Note. ""Spaces" and "Space" in the Content column of these tables indicate fields that HP does not support.

ANSI Standard Labels

ANSI VOL1 Label Format

Character Position in the Label Record	Field Name	Length (in Bytes)	Content
1 - 3	Label Identifier	3	"VOL"
4	Label Number	1	"1"
5 - 10	Volume Identifier	6	Name of this tape volume
11	Accessibility	1	Spaces
12 - 24	Reserved for Future Standardization	13	Spaces
25 - 37	Implementation Identifier	13	Spaces
38 - 51	Owner Identifier	14	Owner of this tape volume
52 - 79	Reserved for Future Standardization	28	Spaces
80	Label-Standard Version	1	"3"

ANSI HDR1/EOF/EOV1 Label Format

Character Position in the Label Record	Field Name	Length (in Bytes)	Content
1 - 3	Label Identifier	3	“HDR” or EOF” or “EOV”
4	Label Number	1	“1”
5 - 21	File Identifier	17	Tape-file name
22 - 27	Fileset Identifier	6	File-set name to which this volume belongs
28 - 31	File Section Number	4	Ordinal number of this volume within a multivolume tape file
32 - 35	File Sequence Number	4	Ordinal number of this tape file within a multifile file-set
36 - 39	Generation Number	4	A number indicating the absolute generation number of this file
40 - 41	Version Number	2	A number indicating the version of this generation
42 - 47	Creation Date	6	Year and day of the year this file was created
48 - 53	Expiration Date	6	Year and day of the year this file can be scratched or overwritten
54	Accessibility	1	Space
55 - 60	Block Count	6	Number of blocks written in this file section
61 - 73	Implementation Identifier	13	Spaces
74 - 80	Reserved for Future Standardization	7	Spaces

ANSI HDR2/EOF2/EOV2 Label Format

Character Position in the Label Record	Field Name	Length (in Bytes)	Content
1 - 3	Label Identifier	3	“HDR” or “EOF” or “EOV”
4	Label Number	1	“2”
5	Record Format	1	“F”: Fixed length “U”: Undefined length
6 - 10	Block Length	5	Number of bytes per block
11 - 15	Record Length	5	Number of bytes per record
16 - 50	Reserved for System Software Use	35	Spaces
51 - 52	Buffer Offset Length	2	“00”
53 - 80	Reserved for Future Standardization	28	Spaces

IBM Standard Labels

IBM VOL1 Label Format

Character Position in the Label Record	Field Name	Length (in Bytes)	Content
1 - 3	Label Identifier	3	“VOL”
4	Label Number	1	“1”
5 - 10	Volume Serial Number	6	Name of this tape volume
11	Reserved for Future Standardization	1	Space
12 - 21	VTOC Pointer	10	Spaces
22 - 31	Reserved for Future Standardization	10	Spaces
32 - 41	Reserved for Future Standardization	10	Spaces
42 - 51	Owner Name and Address Code	10	Owner of this tape volume
52 - 80	Reserved for Future Standardization	29	Spaces

IBM HDR1/EOF1/EOV1 Label Format

Character Position in the Label Record	Field Name	Length (in Bytes)	Content
1 - 3	Label Identifier	3	“HDR” or “EOF” or “EOV”
4	Label Number	1	“1”
5 - 21	Data Set Identifier	17	Tape-file name
22 - 27	Data Set Serial Number	6	File-set name to which this volume belongs
28 - 31	Volume Sequence Number	4	Ordinal number of this volume within a multivolume data set
32 - 35	Data Set Sequence Number	4	Ordinal number of this data set within a multi-dataset file-set
36 - 39	Generation Number	4	A number indicating the absolute generation number of this file
40 - 41	Version Number	2	A number indicating the version of this generation
42 - 47	Creation Date	6	Year and day of the year this file was created
48 - 53	Expiration Date	6	Year and day of the year this data set can be scratched or overwritten
54	Data Set Security	1	“0”
55 - 60	Block Count	6	Number of blocks written in this volume sequence
61 - 73	System Code	13	Spaces
74 - 80	Reserved for Future Standardization	7	Spaces

IBM HDR2/EOF2/EOV2 Label Format

Character Position in the Label Record	Field Name	Length (in Bytes)	Content
1 - 3	Label Identifier	3	“HDR” or “EOF” or “EOV”
4	Label Number	1	“2”
5	Record Format	1	“F”: Fixed length “U”: Undefined length
6 - 10	Block Length	5	Number of bytes per block
11 - 15	Record Length	5	Number of bytes per record
16	Tape Density (5180 Tape Drive only)	1	Value “0”: Full function mode Space: device other than 5180
17	Data Set Position	1	“0”: No volume switch has occurred. “1”: A volume switch previously occurred.
18 - 34	Job/Job Step Identification	17	Spaces
35 - 36	Tape Recording Technique	2	Spaces
37	Control Character	1	Space
38	Reserved for Future Standardization	1	Space
39	Block Attribute	1	“B”: Records are blocked Space: Records are not blocked
40 - 42	Reserved for Future Standardization	8	Spaces
43 - 46	Serial Number (5180 Tape Drive only)	4	“*NNNN*”: Unit's serial number in hexadecimal Spaces: device other than 5180
47	Device Address (5180 Tape Drive only)	1	“0”: 5180 device Space: device other than 5180
48	Checkpoint Data Set Identifier	1	Space
49 - 80	Reserved for Future Standardization	32	Spaces

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z

Special Characters

Content Feedback

First Name: _____ Last Name: _____
Phone: _____ e-mail address: _____
Company: _____

(All contact information fields are required.)

If you're reporting an error or omission, is your issue:

☐ **Minor**: I can continue to work, but eventual resolution is requested.

☐ **Major**: I can continue to work, but prompt resolution is requested.

☐ **Critical**: I cannot continue to work without immediate response.

Comments (give sufficient detail to help us locate the text):

_____
_____
_____
_____
_____

Thank you for taking the time to provide us with your comments.

You can submit this form online, e-mail it as an attachment to pubs.comments@hp.com, fax it to 408-285-5520, or mail it to:

Hewlett-Packard Company
NonStop Enterprise Division
19333 Vallco Parkway, MS 4421
Cupertino, CA 95014-2599
Attn.: Product Manager, Software Publications