Parameter	Description
ty_i	Record type of routine:

Record Type	Description
ABS	CPU multiple entry point overlay.
OVL	CPU overlay.
PP	PP absolute.
REL	Relocatable CPU routine.
	If a record type other than ABS, OVL, PP, or REL is specified, the run is aborted.

Parameter	Description
rec_i	Record name of routine. A routine is allowed on only one alternate system device.

Additional qualifications:

- Once a routine is placed on an alternate system device, SYSEDIT may be used to prohibit access to the routine; however, the space for that routine is not released until LIBDECK is modified and the system is reloaded.
- If extended memory is an alternate system device, all ABS, OVL, or REL routines residing there will be loaded from extended memory directly to the load address.
- If DDP is available, PP programs residing in extended memory will be loaded using DDP, and CPU programs will be loaded using the CPU access to extended memory.

*PROC Directive

The *PROC directive identifies a record as a procedure. It can be treated as any command, with parameters as required by the procedure itself. Further information on procedure creation and execution can be found in the NOS Version 2 Reference Set, Volumes 2 and 3.

The format of the directive is:

```
*PROC,rec1,rec2,...,recn
```

Parameter	Description
rec_i	Record name of routine to be defined as a procedure file.

*SC Directive

The *SC directive specifies the commands in a certain program that are to be processed in product set format rather than in NOS format (refer to the NOS Version 2 Reference Set, Volume 3).

The format of the directive is:

$*SC,ty_1/rec_1,ty_2/rec_2,\ldots,ty_n/rec_n$

Parameter	Description
ty_i/rec_i	Record type and record name of the routine to be processed in product set format.

*FL Directive

The * FL directive specifies the field length that routines to be loaded require to begin execution.

The format of the directive is:

$*FL,ty_1/rec_1-fl_1,ty_2/rec_2-fl_2,\ldots,ty_n/rec_n-fl_n$

Parameter	Description
ty_i/rec_i	Record type and record name of the routine.
fl_i	Field length divided by 100_8 required by the routine.

The actual field length obtained is subject to the rules governing RFL= and MFL= entry points, since the specified fl_i field is placed in the library directory. The system uses this information to determine field length in the following manner:

1. If bit 11 is not set, an RFL= entry point is indicated. The field length is set to the value in the entry.

2. If bit 11 is set (indicating a value of 400000_8), an MFL= entry point is indicated. The field length is determined in one of two ways:

   a. If bit 10 is also set, the field length is set to the maximum of the value of the last RFL command and the value in the entry after masking off these upper 2 bits.

   b. If bit 10 is not set, the field length is set to the maximum of the existing field length and the value in the entry after masking off these upper 2 bits.

*/ Directive

The */ directive specifies comment lines that are listed on the output file. Other than being listed on the output file, comment lines are ignored. They can occur any place in the directives file or on LIBDECK.

The format of the directive is:

```
*/ comment
```

Parameter	Description
comment	A comment line can contain any valid characters and be used for any purpose.

*DELETE Directive

The *DELETE directive deletes a record from the system. It cannot, however, delete a user library (ULIB type record).

The format of the directive is:

$*DELETE,ty_1/rec_1,ty_2/rec_2,...,ty_n/rec_n$

or

$*D,ty_1/rec_1,...,ty_n/rec_n$

Parameter	Description
ty_i/rec_i	Record type and record name to be deleted from the system. ty_i must not be ULIB.

*FILE Directive

The *FILE directive declares an additional file containing records to be added to the system or to logically replace records on the system.

The format of the directive is:

```
*FILE,filename
```

or

```
*FILE,filename,NR
```

Parameter	Description
filename	Name of local file containing addition or replacement records to be placed on the system. File filename is rewound before processing if NR is omitted.
NR	Optional parameter that inhibits rewinding of file filename before processing.

*IGNORE Directive

The *IGNORE directive specifies that records on a replacement file are to be ignored. If no *FILE directive precedes an *IGNORE directive, SYSEDIT ignores the records named on this directive on the replacement file specified by the B parameter. If one or more *FILE directives precede an *IGNORE directive, SYSEDIT ignores the records on the file specified in the most recent *FILE directive.

The format of the directive is:

$*IGNORE,ty_1/rec_1,ty_2/rec_2,\ldots,ty_n/rec_n$

Parameter	Description
ty_i/rec_i	Record type and record name to be ignored on the current replacement file.

*PPSYN Directive

The *PPSYN directive specifies one or more names to be synonymous with the name of an existing PP routine.

The format of the directive is:

$*PPSYN,name/name_1,name_2,\ldots,name_n$

Parameter	Description
name	Name of existing PP routine.
$name_i$	Additional (synonymous) name for name.

Record Types

The following record types may be specified in SYSEDIT directives.

Type	Description
ABS	Multiple entry point overlay.
CAP	Fast dynamic load capsule.
OPL	Modify old program library deck.
OPLC	Modify old program library common deck.
OPLD	Modify old program library directory.
OVL	Central processor overlay.
PP	Peripheral processor program.
PPU	Peripheral processor unit program.
PROC	Procedure record.
REL	Relocatable central processor program.
TEXT	Unrecognizable as a program.
ULIB	User library.

The system library contains a maximum of 62 ULIB type records. If more records are added, only the first 62 can be assessed.

System File Initialization 20

System File Initialization 20

This section describes the system mechanisms used to initialize the fast-attach files used by MODVAL, PROFILE, and RESEX.

ISF

The ISF command initializes the fast-attach system files VALIDUs, PROFILa, RSXDid, and RSXVid.[1] A fast-attach file is a special direct-access file under user index 377777_8 that is initialized with the E parameter on the ISF command and released with the R parameter on the ISF command. However, in order to release a fast-attach file, an idle family situation must be present. That is, the job containing the ISF,R=filename command must be the only job in the family (family count is zero, and the direct-access file count equals the number of fast-attach files). Therefore, the DSD command IDLEFAMILY must be used to clear the system of all jobs. All current jobs and subsystems (such as maintenance jobs, MAGNET, and IAF) must be allowed to complete or be terminated. Refer to the family status display (E,F) in the NOS 2 Operations Handbook to determine when the above conditions have been met.

The fast-attach file mechanism should be used by special system jobs for files that are to be retained as permanent files but have a high enough access rate to make permanent file ATTACHs excessively time-consuming. When a permanent file is activated as a fast-attach file, an entry in the system FNT is made that retains the basic data normally kept in the catalog entry and system sector of the file (interlocks and file name). This eliminates the catalog search and system sector read normally necessary to attach a permanent file.

If the file is a shared (global) fast-attach file for a multimainframe network, additional information is also maintained in the fast-attach table on the link device. The type of file determines whether it is entered as global fast attach or local fast attach. This criterion is kept internal to ISF. Basically, VALIDUs and PROFILa are entered as global fast-attach files if they reside on a shared device. The resource files are always entered as local fast attach. A limit of 77_8 exists on the number of files that can be entered as global fast attach in a multimainframe environment.

Because of the special nature of fast-attach files, a job containing an ISF command must be a system origin job. Processing the command causes a search of the system permanent file catalog (UI=377777_8) for files with the predefined names previously listed. They are defined in a table internal to ISF.

The format of the ISF command is:

```
SF,option,FM=familyname,SJ=filename,SP=filename.
```

1. The resource files are generated and maintained separately for each machine id in a multimainframe or single mainframe system by appending the machine id to the file name (for example, RSXVid becomes RSXVAB on the machine with an id of AB).

Parameter	Description
option	Specifies whether to initialize or release system files.

option	Description
E=filename	System file that is initialized. If E=0 or no filename is specified (neither E nor R appear), all files defined in the ISF table are initialized (refer to table 20-1).
R=filename	Currently active system file that is released from fast-attach status. If R=0, all of the files in the ISF table for the specified family that are currently active are released. When this parameter is specified, an idle family situation (family count is zero, and the direct-access file count equals the number of fast-attach files) must first be created with the DSD command IDLEFAMILY. When the family is idle, the IDLEFAMILY command must be entered again so that the system will accept the ISF command. To release fast-attach system files, you must use this format of the ISF command: `X.ISF(R=filename)`

Since initialization and release are mutually exclusive, E and R cannot appear on the same command.

Parameter	Description
FM=familyname	Family of devices. If FM is not specified, the calling job's current family is used. The calling job's family will be restored upon exit from ISF.
SJ=filename	Job file that ISF submits as a system origin job. The file must be an indirect-access permanent file stored under the system user index (377777_8). If SJ is specified without =filename, ISF assumes SJ=SYSJOB. If SJ=0 is specified, no job is submitted.
SP=filename	Procedure file that ISF calls with system origin. The file must be an indirect-access permanent file stored under the system user index. If SP is specified without =filename, ISF assumes SP=SYSPROC. If SP=0 is specified, no procedure is called.

ISF is automatically executed at each deadstart (refer to Deadstart Sequencing later in this section). This enables the use of SYSJOB and/or SYSPROC to SYSEDIT local modifications into the system.

ISF can also be entered as a command from the console with the DSD X command or from any system origin job.

The matrix in table 20-1 shows how the initialize (E) and release (R) parameters affect individual fast-attach files.

Table 20-1. Initialize and Release Parameters of Fast-Attach Files

Name of Fast-Attach System File	Initialize (E) Parameter	Release (R) Parameter
VALIDUs PROFILa	Make global fast-attach file.	Return from fast-attach status to normal direct access.
RSXDid RSXVid	Make local fast-attach file. If the file does not exist in the system catalog (UI=377777_8), ISF creates the file and makes it a fast-attach file. The file is created with the backup requirement set to none (BR=N). This prevents PFDUMP from dumping the file. If either RSXDid or RSXVid is specified, ISF initializes both files.	Return from fast-attach status to normal direct access. If either RSXDid or RSXVid is specified, ISF returns both files to normal direct access.

One use of the R parameter is to release fast-attach files activated on a device to be initialized or unloaded. Device initialization is not initiated as long as any direct-access files are active on the device (an activated fast-attach file is treated the same as an active direct-access file). Until these files are released, the system will reply to an attempted device initialization with the error message:

```
ACTIVE FILES ON DEVICE
```

Similarly, a device cannot be unloaded until all its fast-attach files are released.

Deadstart Sequencing

During a level 0 deadstart, a job is queued for input using the service class DSSC. This job executes the PP routine CMS, whose normal functions include mass storage initialization and recovery. After completing these functions, CMS checks its service class. If the service class is DSSC (indicating a deadstart sequencing call), CMS places one of the following ISF commands in its command buffer and causes the system to execute it.

Command	Description
ISF,FM=0,SJ.	Used for recovery deadstarts.
ISF,FM=0,SJ,SP.	Used for other deadstarts.

The CMS deadstart sequencing job begins execution only after job processing has been enabled by the DSD AUTO or MAINTENANCE command. The job scheduler prohibits scheduling of all other jobs until the CMS job completes, thereby ensuring that all system files in the default family are initialized and that a SYSPROC procedure (if any) is executed before normal job processing begins (refer to ISF earlier in this section).

NOTE

Since the deadstart job (including SYSPROC commands) must complete before normal job processing begins, avoid using SYSPROC for unnecessary or time-consuming tasks. SYSJOB may be more appropriate for some tasks. Also, SYSPROC cannot use tapes since MAGNET is not available. If SYSPROC uses removable disk packs, they must be mounted and ready.

Tracer/Probe Utilities 21

Tracer/Probe Utilities 21

The TRACER and PROBE utilities described in this section provide data for statistical analysis of the system. The data is used to determine where problems occur and where improvements in design might be made, and to perform system tuning. The TRACER utility monitors the system's activity and gathers data periodically. The PROBE utility traps and measures particular internal events in the system. Both utilities capture valuable data that may not be obtainable any other way.

Tracer Utility

The TRACER utility monitors these conditions:

- Channel activity by channel.
- Channel reserved.
- Channel requested.

- Requests pending.
- Number of active PPs.
- Number of active concurrent PPs.

- Buffered input/output list parameters.
- Buffered input/output channel busy.
- CPU use (idle, system, subsystem, system-related activity, or user activity).

- Subsystem CPU use.
- Storage moves pending.
- PP saturation.

- Extended memory transfer in progress.
- MTR cycle time.
- CPU0 or CPU1 is or is not in monitor mode.

- Same storage move request is pending.
- Control points in automatic recall (I) status.
- Control points in periodic or automatic recall (X) status.

- Control points in waiting (W) status.
- Amount of available memory.
- Amount of memory at control points by service class.

- Amount of memory in queue by service class.
- Amount of memory at control points by subsystem.
- Number of noninteractive jobs.

- Number of detached interactive jobs.
- Number of online jobs.
- Number of preinitial jobs by service class.

- Number of executing jobs by service class.
- Number of jobs rolled out by job scheduler for each service class.
- Number of jobs rolled in or out by system control point processing for each service class.
- Number of jobs in a timed/event rollout queue by service class.
- Number of jobs rolled out by interactive input/output processing for each service class.
- Number of disabled jobs rolled out by service class.
- Number of suspended jobs rolled out by service class.
- Number of jobs with rollout file errors by service class.
- Number of EST entries in use.
- Number of FNT entries in use.
- Number of EJT entries in use.
- Number of queued files assigned to jobs at control points.
- Number of input files by service class.
- Number of print and punch files by service class.
- Number of other queued files not assigned to jobs at control points.
- Number of QFT entries in use.
- Number of FOT entries in use.
- Number of control points in use.
- Number of pseudo-control points in use.
- Number of IAF active users.
- Number of IAF pots available.
- Number of IAF pots in use.
- Number of tape drives in use.
- Number of tracks available by mass storage device.
- Number of segment table reads.
- Number of missed clock updates.
- Number of extended memory moves.
- Number of central memory moves.
- Number of all rollouts.
- Number of rollouts to secondary rollout devices.
- Number of all sectors rolled.
- Number of sectors rolled out to secondary rollout devices.
- Number of rollouts/user limits.
- Number of time slices.
- Number of PP priority exchanges.
- Number of times communication buffer not available.
- Number of EJT scans.
- Number of schedulable jobs.
- Number of jobs preempted.

- Number of jobs scheduled.
- Number of jobs scheduled with no constraints.
- Number of insufficient CM scans.
- Number of insufficient EM scans.
- Number of no control point scans.
- Number of ISHARED table changes.
- Number of ISHARED device updates.
- Amount of ISHARED seek time.
- Amount of ISHARED updating time.

The TRACER utility includes the following programs:

Program	Description
ICPD	A CPU program that initiates system monitoring by CPD.
CPD	A PP program that monitors any of the system activities just listed. CPD is dedicated to a PP while it is monitoring system activity. Data is written to a direct access permanent file for future analysis.
ACPD	A postprocessor program that generates an output report from the direct access permanent file written by CPD.
ENDCPD	A CPU program that terminates system monitoring by CPD.

TRACER Commands

TRACER commands are described next.

ACPD Command

ACPD reads the sample data file produced by CPD and generates reports, in both user-readable and machine-readable formats, for further analysis. The sample data file must be attached before ACPD is called. If ICPD is called with the M=A or M=M parameters, the sample data file can be accessed while CPD is still active.

ACPD assumes a continuity of the sample data file. Therefore, the uncollected information during the time gap separating two consecutive files is assumed to be present, although the information is not reported. As a result, if the consecutive files on the sample data file are not in chronological order, ACPD terminates and issues an error message.

Format:

```
ACPD,p1,p2...pn.
```

pi	Description
FN=datafile	Name of sample file. Default is SAMPLE. This file is not rewound before or after processing.
L=outfile	Name of output file generated by ACPD. Default is OUTPUT.
S=sumfile	Name of machine-readable summary file generated by ACPD. If this parameter is omitted, no summary file is generated. If S is specified without sumfile, SUMMARY is assumed.
IC=nn	Select the report interval by specifying the number of CPD sample file records. This allows selection of report intervals of less than 1 minute. You can specify the report interval by using either the IC or IN parameter, but not both.
IN=nn	Time span of report interval, in minutes. Default value is 6 minutes. ACPD generates a report for each report interval. You can specify the report interval by using either the IN or IC parameter, but not both.
LO=Z	Report data items with zero values. Default is to suppress data items with zero values.
N=nn	Number of files on the sample data file to be analyzed and reported. Default is only one file processed. If N is not equivalenced, all files are processed until EOI is reached on the sample data file. The sample data file is not rewound before processing.
BT=hhmmss	Beginning time in the form hour minute second. If BT=hhmmss is specified, only data collected after this time on the date specified by BD=yymmdd is reported. If BT is omitted or no time is specified, BT=0 is assumed.

pi	Description
ET=hhmmss	Ending time in the form hour minute second. If ET=hhmmss is specified, only data collected before this time on the date specified by ED=yymmdd is reported. If ET is omitted or no time is specified, ET=0 is assumed.
BD=yymmdd	Beginning date in the form year month day. If BD=yymmdd is specified, only data collected on or after this date is reported. If BD is omitted or no date is specified, the beginning date is the date of the current file on the sample data file.
ED=yymmdd	Ending date in the form year month day. If ED=yymmdd is specified, only data collected on or before this date is reported. If ED is omitted or no date is specified, but ET=hhmmss is specified, the ending date is the same as the beginning date. If neither ED nor ET is specified, ACPD terminates when the number of files specified by the N parameter are processed, or end-of-information is encountered, whichever happens first.

ACPD begins by processing the command parameters. If the beginning time (BT) and beginning date (BD) are specified, ACPD first locates the correct file and then processing begins. Processing continues until the ending time (ET) and ending date (ED) are encountered, the number of files specified by the N parameter have been processed, or end-of-information is encountered, whichever happens first.

If the BT and BD parameters are not specified, ACPD starts at the current position of the sample data file. The sample data file is not rewound before processing starts.

ENDCPD Command

The ENDCPD command terminates all CPD data gathering.

Format:

```
ENDCPD.
```

ICPD Command

ICPD defines a mass storage file to which CPD will write statistical data and then initiates system monitoring by CPD.

NOTE

If you want the statistical data to include only data from the start of this TRACER run rather than an accumulation of data from the last deadstart, enter the command PROBE(OP=C,L=O) before entering the ICPD command so that the statistical data area of CMR will be cleared prior to the reporting done by CPD.

Format:

```
ICPD,p1,p2,...pn.
```

pi	Description
FL=fl	Fast loop sampling frequency, in milliseconds, during which items such as PPs active and move request pending are sampled. Default is 5 milliseconds.
ML=ml	Medium loop sampling frequency, in milliseconds, during which items such as control points in I, X, and W status and field length available are sampled. Default is 100 milliseconds.
SL=sl	Slow loop sampling frequency, in milliseconds, during which items such as IAF users and tape drives in use are sampled. Default is 1000 milliseconds.
FW=fw	Snapshot loop sampling frequency, in seconds. Default is 5 seconds.
FN=filename	Name of sample data file. Default is SAMPLE. ICPD will attempt to attach a direct access file by this name. If no file exists, it will be defined. If a file is found, ICPD will skip to EOI and write an EOF. CPD will then start writing data after the EOF.
M=mode	Permanent file mode for sample data file. Default is M=WRITE. mode can have one of these values:

mode	Description
WRITE or W	Sample data file attached in write mode.
APPEND or A	Sample data file attached in append mode.
MODIFY or M	Sample data file attached in modify mode.

NOTE

If the sample data file is attached in write mode, the file cannot be accessed until ENDCPD is run. If the sample data file is to be accessed while data is being collected, append or modify mode must be specified. In this situation, the file may be attached in read/allow modify (RM) mode. (Attaching the file in write mode rather than in modify or append mode expends less overhead when interlocking and writing the data file.)

If a loop time is set to 0 (zero), no samples for that loop will be taken. If the data block sample time is set to 0 (zero), the data file will be written only when the sampling interval terminates.

All numeric data should lie within the range 0 through 4095 (0 through 7777_8).

Output File Format

The first three pages of the output report produced by ACPD contain the header block information. Next the data items are reported for fast, medium, and slow loop samples. The report ends with the snapshot data items.

Figure 21-1 is an example showing the format of the output report. The example has been simplified and condensed to reduce the amount of output. Also, supporting text has been added to the example.

Data items monitored at successive time intervals are listed in the same row. For each data item, the average, standard deviation, and percentage are listed in successive rows. Up to 10 intervals can be listed per page in successive vertical columns. If the output file contains more than 10 columns per row, the output report lists the first 10 columns for all rows of data items and then resumes listing subsequent intervals following the snapshot data items.

The SUBTOTAL column contains the values of the data items for the time spanned by the preceding intervals on the current page; that is, the time spanned by the preceding intervals is considered one interval. The TOTAL column appears after the last interval reported and contains the statistical values of the data items for the entire run. The SUBTOTAL column is not listed if the subtotal data and total data are identical and would appear on the same page. In this case, only the TOTAL column is listed. The SUBTOTAL and TOTAL columns are not reported for the snapshot data items. The *MAX* and *MIN* columns appear at the end of the report and contain the maximum and minimum interval values of the data items for the entire run. The maximum and minimum interval values on each page are indicated by brackets and parentheses, respectively.

The average is not reported for data items that have a weighting factor of 1, and the percentage is not reported for data items that have a weighting factor of 100, since this information is redundant.

```
A C P D - VER  9.1         CDC NETWORK OPERATING SYSTEM.          NOS 2                    yy/mm/dd. hh.mm.ss.  PAGE     1

        START DATE               86/05/25.               } Start of sampling interval
        START TIME               23.50.13.

        DATA FILE NAME             SAMPLE                } ACPD command parameters (FN, IC, IN)
        REPORT INTERVAL (MINUTES)       2

        CPD VERSION                          9.1         }
        FAST LOOP INTERVAL (MSECS)            10         }
        MEDIUM LOOP INTERVAL (MSECS)         100         } ICPD command parameters (FL, ML, SL, FW)
        SLOW LOOP INTERVAL (MSECS)          1000         }
        SNAPSHOT LOOP INTERVAL (SECS)         60         }

        NUMBER OF CPUS                         1         }
        NUMBER OF PPUS                        20         }
        NUMBER OF CPPS                        10         }
        MEMORY SIZE / 100B                 7777B         } Hardware configuration at beginning of sampling interval
        USER EM / 1000B                     200B         }
        MAGNETIC TAPE UNITS                    4         }

        NUMBER OF EST ENTRIES                76B         }
        NUMBER OF FNT ENTRIES                23B         }
        NUMBER OF EJT ENTRIES              1000B         }
        NUMBER OF QFT ENTRIES              1000B         }
        NUMBER OF FOT ENTRIES                10B         }
        NUMBER OF CONTROL POINTS             33B         } Software configuration at beginning of sampling interval
        NUMBER OF PCP-S                      20B         }
        CMRSIZE / 100B                     1632B         }
        LIBDECK NUMBER                         0         }
        RECOVERY LEVEL                         0         }
        NUMBER OF IAF TERMINALS             400B         }
        MACHINE ID                            32         }

        CPU USAGE CATEGORIES (CTCUL)                5B   }
        CHANNEL TABLE LENGTH (CTALL)               55B   }
        LENGTH OF AN EST ENTRY (ESTE)               2B   }
        LENGTH OF AN FNT ENTRY (FNTE)               2B   }
        LENGTH OF AN EJT ENTRY (EJTE)               4B   }
        LENGTH OF A QFT ENTRY (QFTE)                4B   }
        NUMBER OF SERVICE CLASSES (MXSC)           17B   } System statistics; the symbols are assembly constants used in NOS
        NUMBER OF JOB STATUSES (MXJS)              16B   }
        NUMBER OF CONNECTION STATUSES (MXCS)        3B   }
        NUMBER OF FILE TYPES (MXFT)                21B   }
        MAXIMUM ORIGIN TYPES (MXOT)                 6B   }
        NUMBER OF MASS STORAGE DEVICES             11B   }
        LENGTH OF MSAL TABLE (MXRS)                14B   }

        CPU RECALL DELAY                           30B   }
        PP AUTO RECALL DELAY                     1750B   }
        CPU PRIORITY INCREMENTING FREQUENCY       200B   } System delay parameters at beginning of sampling interval
        MEMORY PAD SIZE                           400B   }
        INPUT FILE SCHEDULING INTERVAL              2B   }
        SCHEDULER CYCLE INTERVAL                    1B   }

        HIGH SPEED DISK BUFFERS                     0B   } System buffers at beginning of sampling interval
        EM/PP BUFFERS                               1B   }
```

Figure 21-1. Example of TRACER Output

(Continued)

(Continued)

```
A C P D - VER  8.0          CDC NETWORK OPERATING SYSTEM.         NOS 2                  yy/mm/dd. hh.mm.ss.  PAGE     2

EQUIPMENT STATUS TABLE
NO.  TYPE  STAT  EQ  UN  CHANNELS    EST ENTRY                                     TRACK      FILES

 0    RD    ON    1  11   0          00000000000022041271 00000000000000000000
 1    DS    ON    7   0  10          00000010000004237000 00000000000000000000
 2    NE    ON    0   0   0          20000000000016050000 00000000000000070000
 3    TE    ON    0   0   0          20000000000024050000 00000000000000070000
 4    TT    ON    0   0   0          20000000000024240200 00000000000000070000
 5    DE    ON            0          40100000000004051274 00000000000000000000    770   -------------
 6    DQ    ON           13          62100013137704211277 00000000000000000000   3222   -XT----------N
 7    DQ    ON           13          62100013137704211302 00000000000000000000   3222   -XT----------N
10    DJ    ON           26 32       42103226137704121305 00000000000000000000   3150   -------------
11    DJ    ON        1  32 26       47002632137704121310 00000000000000000000   3150   -------------
12    DJ    ON        2  26 32       47003226137704121313 00000000000000000000   3150   -------------   (1)
13    DI    ON        3  32 26       47002632137704111316 00000000000000000000   3140   -------------
14    DI    ON        4  26 32       47003226137704111321 00000000000000000000   3140   -------------
15    DI    ON        6  26 32       47003226137704111324 00000000000000000000   3140   -------------
16    DJ    ON        0  13          47000013137704121327 00000000000000000000   3150   -------------
17    DJ    ON        1  13          47000013137704121332 00000000000000000000   3150   -------------
20    DJ    ON        2  13          47000013137704121335 00000000000000000000   3150   -------------
21    CR    ON    4   0  12          00000012000003224000 00000000000000000000
22    CP    ON    5   0  12          00000012000003205000 00000000000000000014   Track capacity
23    LT    ON    7   0  12          02000012000014247100 00000000000000000000   for mass storage
24    LT    ON    6   0  12          00000012000014246000 00000000000000000000   devices
30    TT    OFF   6  10   6          20000006000064246310 00000000000000000000
31    TT    OFF   6   6   6          20000006000064246226 00000000000000000000
32    NP    ON    7   3   3          00000003000016207103 00000000000000000006
34    NP    OFF   7   1   1          00000001000056207101 00000000000000000000
35    NP    OFF   7   7   2          00000002000056207107 00000000000000000000
36    NP    OFF   7   4   7          00000007000056207104 00000000000000000000
43    CS    ON    0   6   0          00000000440003230106 00000000000000000000
44    CT    ON    0   0   0          00000000440003240000 00000000000000000000
45    CT    ON    0   2   0          00000000440203240202 00000000000000000000
50    NT    ON    0   0  31 33       02003331000015240000 00000000000000000000
51    NT    ON    0   1  31 33       02003331000016240001 00000000000000000000
52    NT    ON    0   2  31 33       02003331000016240002 00000000000000000000
53    NT    ON    0   3  31 33       02203331000016240003 00000000000000000030
54    NT    ON    0   4  31 33       02203331000016240004 00000000000000000000
55    NT    ON    0   5  31 33       02203331000016240005 00000000000000000000
60    NC    ON    0   5   6          20000006000016030045 00000000000000000000
```

- EST ordinal
- Equipment type
- ON/OFF status
- Equipment numbers for mass storage devices and unit numbers for nonremovable mass storage devices are not listed
- Octal representation of EST entry at beginning of sampling interval

(1) The letters to be used here are order dependent and indicate file types that can reside on a given device. The order is GXTIORDPLBSRRN, defined as follows.

Letter	File Type	Letter	File Type
G	Checkpoint	P	Primary
X	System	L	Local
T	Temporary	B	LGO
I	Input	S	Secondary rollout
O	Output	R	Reserved for CDC
R	Rollout	R	Reserved for CDC
D	Dayfile	N	Temporary on nonshared device

Figure 21-1. Example of TRACER Output

(Continued)

(Continued)

```
A C P D - VER  8.0          CDC NETWORK OPERATING SYSTEM.          NOS 2                   yy/mm/dd. hh.mm.ss.  PAGE      3

        SYSTEM CONTROL INFORMATION

        SERVICE   QUEUE              PRIORITIES                         SERVICE LIMITS
        CLASS                                               CB/L      CP        CP      CM      NJ     TD
                                                            CB/U      FL        AN              TP
                             IL      LP      UP     WF      IP        EC        EM      DS      FC  CS  FS
                                                                      SD

          SY        IN               7770    7776    1            30      6770        10      20    7777     341
                    EX       4000    2000    7600    1     7000   30      3777     77777            4004
                    OT               7000    7776    1                    3777      7777       0     0   0   0
                                                                            20

          BC        IN                 10    4000    1            30      3770        10     200      20     341
                    EX       2000    1000    4000    1     2000   30      3777     77777            4004
                    OT                  1    7000    1                    3777      7777       0     0   0   0
                                                                            20

          RB        IN                 10    4000    1            30      3770        10     200       6     341
                    EX       2000    1000    4000    1     2000   30      3777     77777            4004
                    OT                  1    7000    1                    3777      7777       0     0   0   0
                                                                            20

          TS        IN               7000    7770    1            30      6770        10      10    7777     702      This page shows the
                    EX       3770    3760    7000    1     4000   30      3777     77777            4004               status of the job
                    OT                  1    7000    1                    3777      7777       0     0   0   0      control area at the
                                                                            20                                      beginning of the
                                                                                                                    sampling interval.
          DI        IN               3000    5000    1            30      4660        20     240    7777     702      The values are set
                    EX       3300    2710    4700    4     3700   30      3777     77777            3600               IPRDECK entries
                    OT                200    7000    1                    3777      7777       0     0   0   0      QUEUE and SERVICE or
                                                                            20                                      by the DSD commands
                                                                                                                    QUEUE and SERVICE.
          NS        IN               7770    7776    1            74      7770        20     200    7777     341
                    EX       7772    7770    7776    1     7000   74      3777     77777            7770
                    OT                  1    7000    1                    3777      7777       0     0   0   0
                                                                            20

          SS        IN               7770    7776    1            70      7770        10      20    7777     341
                    EX       7772    7770    7776    1     7000   70      3777     77777            7770
                    OT               7400    7776    1                    3777      7777       0     0   0   0
                                                                            20

          MA        IN                  1      10    1             2         1        10      20    7777     341
                    EX          4       1     100    4       10    2      3777     77777              10
                    OT               7000    7776    1                    3777      7777       0     0·  0   0
                                                                            20

          CT        IN               7770    7776   .1            30      6770        10     200    7777     341
                    EX       5010    4720    7000    1     7000   30      3777     77777            4004
                    OT               7000    7776    1                    3777      7777       0     0   0   0
                                                                            20

          I0        IN               2400    4010   20            20       760        10    1130      10     341
                    EX        170     110     770   40      160   27      3777     77777            4004
                    OT                200    7000    1                    3777      7777       0     0   0   0
                                                                            20

          I1        IN               3000    5000    2            30      4670        10     240      20     341
                    EX       3300    2710    4700   10     3700   30      3777     77777            4004
                    OT                200    7000    1                    3777      7777       0     0   0   0
                                                                            20

          I2        IN                 10    4000    1            30      3770        10     200       0     341
                    EX       2000    1000    4000    1     2000   30      3777     77777            4004
                    OT                  1    7000    1                    3777      7777       0     0   0   0
                                                                            20

          I3        IN               7000    7770    1            30      6770        10      20    7777     702
                    EX       3770    3760    7000    1     4000   30      3777     77777            4004
                    OT                  1    7000    1                    3777      7777       0     0   0   0
                                                                            20
```

Figure 21-1. Example of TRACER Output

(Continued)

(Continued)

A C P D - VER 8.0 CDC NETWORK OPERATING SYSTEM. NOS 2 yy/mm/dd. hh.mm.ss. PAGE 4

2 MINS INTERVAL		(2) 23.52.13 INTERVAL	23.54.13 INTERVAL	23.56.13 INTERVAL	23.58.12 INTERVAL	00.00.12 INTERVAL	00.02.13 INTERVAL	00.04.12 INTERVAL	00.06.13 INTERVAL	00.08.13 INTERVAL	00:18 HR SUBTOTAL (5) (6)
(3) FAST LOOP SAMPLES		10480	10519	10525	10517	10528	10524	10516	10507	10517	94633
PPUS ACTIVE (4)	AV	(4.236)	6.272	4.987	7.186	[7.708]	4.552	4.504	4.250	6.154	5.539
	SD	(0.014)	0.139	0.118	0.151	[0.363]	0.238	0.054	0.021	0.143	1.261
	PC	(21.180)	31.359	24.933	35.931	[38.538]	22.760	22.519	21.250	30.770	27.693
MOVE REQUEST PENDING	SD	(0.000)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	[0.001]	0.001
	PC	(0.000)	0.010	0.067	0.000	[0.124]	0.048	0.000	0.000	0.067	0.035
NO PPU AVAILABLE	SD	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	PC	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
EM TRANSFER IN PROG	SD	0.000	0.000	0.000	0.000	0.001	0.000	[0.001]	(0.000)	0.000	0.001
	PC	(0.048)	0.152	0.105	0.095	0.209	0.162	[0.209]	0.086	0.057	0.125
MTR CYCLE TIME	AV	0.371	0.393	0.372	[0.436]	0.412	0.372	0.370	(0.364)	0.433	0.392
	SD	0.010	0.007	0.008	(0.000)	0.013	0.007	0.001	0.004	[0.023]	0.029
MONITOR MODE - CPU 0	SD	[0.013] (7)	0.002	0.012	0.009	0.003	(0.002)	0.010	0.002	0.005	0.020
	PC	(7.887)	12.358	8.603	[13.605]	10.961	8.200	9.870	8.376	9.186	9.895
MONITOR MODE - CPU 1	SD	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	PC	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SCHEDULER ACTIVE	SD	(0.001) (8)	0.001	0.006	0.041	0.022	0.019	0.001	0.001	[0.069]	0.057
	PC	5.114	(5.067)	7.335	15.915	[19.136]	7.051	5.696	5.805	12.110	9.248
CHANNEL RESERVED											
CH 12	SD	0.000	0.000	(0.000)	0.000	0.036	0.000	0.001	0.000	[0.068]	0.034
	PC	0.105	0.076	0.076	0.095	3.776	0.095	0.095	(0.038)	[6.898]	1.250
CH 13	SD	0.010	0.262	0.012	(0.003)	0.033	0.059	0.016	0.005	[0.275]	0.168
	PC	7.781	[39.808]	11.135	16.810	12.914	19.718	25.217	(7.434)	33.625	19.383

(2) The interval ending time is shown at the top of each report column. The report interval is specified by the IC or IN parameter of the ACPD command.

(3) The sampling frequency for fast loop items is specified by the FL parameter of the ICPD command. Refer to table 21-1 for all fast loop items that TRACER reports. Fast loop items are described following the table.

(4) The statistics reported for each data item are the average (AV), standard deviation (SD), and percentage (PC) for the report interval.

(5) The SUBTOTAL column contains a summation of all data gathered from all intervals of each page. This column is not listed if the subtotal data and the total data are identical and would appear on the same page.

(6) This example has 13 columns of information for each row of data items. However, only 10 columns can be shown per page, so the remaining columns are listed at the end of the report.

(7) The values enclosed in brackets are the maximum interval values for each row of data items on each page. Refer to note (14).

(8) The values enclosed in parentheses are the minimum interval values for each row of data items on each page. Refer to note (15).

Figure 21-1. Example of TRACER Output

(Continued)

(Continued)

************************											
(9) MEDIUM LOOP SAMPLES		1170	.1175	1175	1173	1175	1175	1175	1173	1175	10566
CPS IN W STATUS	AV	(0.225)	0.339	0.889	1.868	[2.494]	1.699	2.085	1.282	1.278	1.351
	SD	0.043	0.144	[0.320]	0.113	0.293	(0.002)	0.122	0.134	0.137	0.747
	PC	(0.805)	1.209	3.175	6.671	[8.907]	6.067	7.446	4.579	4.566	4.825
CPS IN X STATUS	AV	(5.787)	6.452	5.894	[6.802]	6.078	5.906	6.080	5.827	6.079	6.101
	SD	0.020	0.014	(0.003)	0.044	0.101	0.009	0.162	[0.190]	0.159	0.328
	PC	(20.669)	23.043	21.052	[24.293]	21.705	21.091	21.714	20.811	21.712	21.788
CPS IN I STATUS	AV	(0.067)	0.130	0.083	[0.185]	0.177	0.091	0.095	0.075	0.141	0.116
	SD	(0.001)	0.016	0.015	0.004	0.005	0.011	0.008	0.002	[0.075]	0.049
	PC	(0.238)	0.465	0.295	[0.661]	0.632	0.325	0.340	0.268	0.505	0.414
SAME MOVE REQUEST	SD	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	PC	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
FL AVAILABLE	AV	[43470]	42360	33638	26750	24710	30170	(24120)	33750	31400	32500
	SD	(1.227)	7.470	22.167	82.904	[126.210]	5.241	60.961	9.916	117.505	337.642
	PC	[61.730]	59.740	48.185	39.794	36.213	42.016	(34.957)	48.459	44.193	46.143
USER EM AVAILABLE	AV	0B	0B	0B	0B	0B	0B	0B	0B	0B	0B
	SD	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	PC	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
NON INTERACTIVE JOBS	AV	(1.022)	2.096	2.668	4.243	[6.591]	2.603	2.351	2.191	3.598	3.040
	SD	0.136	(0.023)	0.456	[0.619]	0.139	0.462	0.258	0.033	0.178	1.556
	PC	(0.177)	0.364	0.463	0.737	[1.144]	0.452	0.408	0.380	0.625	0.528
DETACHED JOBS	AV	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	SD	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	PC	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ON-LINE JOBS	AV	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	SD	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	PC	0.174	0.174	0.174	0.174	0.174	0.174	0.174	0.174	0.174	0.174
PRE-INITIAL JOB STEP											
SY	AV	(1.000)	1.000	1.001	1.029	1.000	1.000	1.000	1.001	[1.344]	1.042
	SD	(0.000)	0.000	0.001	0.029	0.000	0.000	0.000	0.001	[0.344]	0.157
	PC	(0.174)	0.174	0.174	0.179	0.174	0.174	0.174	0.174	[0.233]	0.181
BC	AV	(0.025)	1.000	0.684	1.166	[1.665]	0.094	0.236	0.092	0.462	0.603
	SD	0.025	(0.000)	0.125	0.166	0.335	0.094	0.236	0.029	[0.462]	0.582
	PC	(0.004)	0.174	0.119	0.202	[0.289]	0.016	0.041	0.016	0.080	0.105
RB	AV	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	SD	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	PC	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
TS	AV	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	SD	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	PC	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

(9) The sampling frequency for medium loop items is specified by the ML parameter of the ICPD command. Refer to table 21-1 for all medium loop items that TRACER reports. Medium loop items are described following the table.

Figure 21-1. Example of TRACER Output

(Continued)

(Continued)

```
  ***********************
⑩ SLOW LOOP SAMPLES            119       119       119       119       119       119       119       119       119      1071
  IAF USERS            AV    1.000     1.000     1.000     1.000     1.000     1.000     1.000     1.000     1.000     1.000
                       SD    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                       PC    0.781     0.781     0.781     0.781     0.781     0.781     0.781     0.781     0.781     0.781
  IAF POTS ALLOCATED   AV  455.000   455.000   455.000   455.000   455.000   455.000   455.000   455.000   455.000   455.000
                       SD    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
  IAF POTS IN USE      AV (193.000)  193.000   193.017   193.008   193.017  [197.297]  193.000   193.000   193.000   193.482
                       SD  (0.000)     0.000     0.000     0.008     0.017    [4.297]    0.000     0.000     0.000     1.967
                       PC (42.418)    42.418    42.421    42.419    42.421   [43.362]   42.418    42.418    42.418    42.524
  QUEUE FILES ASSIGNED AV  (0.000)     0.008     0.000     0.008     0.050     0.000     0.000     0.000   [0.092]     0.018
                       SD  (0.000)     0.008     0.000     0.008     0.050     0.000     0.000     0.000   [0.092]     0.046
                       PC  (0.000)     0.002     0.000     0.002     0.012     0.000     0.000     0.000   [0.023]     0.004
  INPUT FILES
          SY           AV    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                       SD    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                       PC    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
          BC           AV    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                       SD    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                       PC    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
          RB           AV    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                       SD    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                       PC    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
          TS           AV    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                       SD    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                       PC    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
  PRINT FILES
          SY           AV  (0.000)     0.000     0.000     0.000   [0.008]     0.000     0.000     0.000     0.000     0.001
                       SD  (0.000)     0.000     0.000     0.000   [0.008]     0.000     0.000     0.000     0.000     0.004
                       PC  (0.000)     0.000     0.000     0.000   [0.002]     0.000     0.000     0.000     0.000     0.000
          BC           AV    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                       SD    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                       PC    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
          RB           AV    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                       SD    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                       PC    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
          TS           AV    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                       SD    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                       PC    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                       PC    0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
```

⑩ The sampling frequency for slow loop items is specified by the SL parameter of the ICPD command. Refer to table 21-1 for all slow loop items that TRACER reports. Slow loop items are described following the table.

Figure 21-1. Example of TRACER Output

(Continued)

(Continued)

```
   ************************
(11) SNAP SHOT OF INSL
         (IN0L) =      0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0200   0200   0200   0200   0200   0200   0200   0200   0200
         (IN1L) =      0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
         (IN2L) =      0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
         (IN3L) =      0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
         (IN4L) =      0000   0000   0000   0000   0000   0000   0000   0000   0000   Installation area
                       0000   0000   0000   0000   0000   0000   0000   0000   0000   from CMR
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
         (IN5L) =      0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
         (IN6L) =      0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
         (IN7L) =      0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
                       0000   0000   0000   0000   0000   0000   0000   0000   0000
```

(11) The sampling frequency for snapshot loop items is specified by the FW parameter of the ICPD command.

Figure 21-1. Example of TRACER Output

(Continued)

(Continued)

MTR MXN TIME	743	743	743	743	743	743	743	743	743	
WORST CASE MTR CYCLE TIME	6	6	6	6	6	6	6	6	6	
MISSED CLOCK UPDATES	0	0	0	0	0	0	0	0	0	
EM MOVES	0	0	0	0	0	0	0	0	0	
CM MOVES	243	243	276	276	315	344	344	344	377	
DIRECT MOVES	54	56	70	74	77	85	90	96	97	
ROLLOUTS TO MS	0	0	0	0	0	0	0	0	0	
SECONDARY ROLLOUTS	0	0	0	0	0	0	0	0	0	
SECTORS ROLLED TO MS	6046	6100	7168	7240	7783	8862	10029	10119	10146	
SECONDARY SEC ROLLED	0	0	0	0	0	0	0	0	0	
ROLLOUTS/USER LIMITS	0	0	0	0	1	1	3	3	3	Statistical data area
TIME SLICES	20	22	40	45	46	49	51	51	52	from CMR
PP PRIORITY EXCHANGES	0	0	0	0	0	0	0	0	0	
NO COMM BUFFER AVAIL	0	0	0	0	0	0	0	0	0	
NO PCP AVIALABLE	0	0	0	0	0	0	0	0	0	
EJT SCANS	112	114	190	334	714	781	786	797	846	
SCHEDULABLE JOBS	1024	1171	1357	1498	1633	1795	1946	2101	2238	
JOBS PREEMPTED	0	0	0	0	0	0	0	0	0	
JOBS SCHEDULED	72	74	98	104	106	115	120	127	130	
SCHEDULED NO CONSTRAINTS	0	0	0	0	0	0	0	0	0	
INSUFFICIENT CM SCANS	0	0	0	0	0	0	0	0	0	
INSUFFICIENT EM SCANS	0	0	0	0	0	0	0	0	0	
NO CONTROL POINT SCANS	0	0	0	0	0	0	0	0	0	

Figure 21-1. Example of TRACER Output

(Continued)

(Continued)

```
SNAP SHOT OF CPTW - CPO
               0000  0000  0000  0000  0000  0000  0000  0000  0000 |
               0104  0104  0104  0104  0104  0104  0104  0104  0104 |
               7475  7475  7475  7475  7475  7475  7475  7475  7475 |
               0005  0005  0005  0005  0005  0005  0005  0005  0005 |
               7750  7750  7750  7750  7750  7750  7750  7750  7750 |
SNAP SHOT OF CPTW - CPN                                              |  CP accumulator area
               0000  0000  0000  0000  0000  0000  0000  0000  0000 |  from CMR
               0000  0000  0000  0000  0000  0000  0000  0000  0000 |
               1141  1342  1406  1616  1672  1752  2072  2120  2163 |
               5056  1115  6704  4124  5014  4315  7132  4576  5100 |
               2400  4000  1000  1400  2000  2400  7000  0400  6000 |
SNAP SHOT OF RTCL
               0040  0040  0040  0040  0040  0040  0040  0040  0040 |
               1551  1741  2131  2321  2511  2701  3071  3261  3451 |  Real time clock area
               0000  0000  0000  0000  0000  0000  0000  0000  0000 |  from CMR
               0325  0362  0417  0455  0512  0547  0605  0642  0677 |
               3020  5320  7620  2120  4420  6721  1220  3520  6020 |
SNAP SHOT OF PDTL
               0000  0000  0000  0000  0000  0000  0000  0000  0000 |
               0000  0000  0000  0000  0000  0000  0000  0000  0000 |  Packed date/time area
               1505  1505  1505  1505  1505  1505  1505  1505  1505 |  from CMR
               3127  3127  3127  3127  3200  3200  3200  3200  3200 |
               6415  6615  7015  7214  0014  0215  0414  0615  1015 |
```

Figure 21-1. Example of TRACER Output

(Continued)

(Continued)

```
     A C P D - VER  8.0         CDC NETWORK OPERATING SYSTEM.          NOS 2                      yy/mm/dd. hh.mm.ss.  PAGE    30

     2  MINS  INTERVAL          00:18 HR   83/05/25 TO 83/05/26
                             (13)TOTAL  (14)*MAX*  (15)*MIN*

(12) FAST LOOP SAMPLES               94633
     PPUS ACTIVE           AV        5.539      7.708      4.236
                           SD        1.261      0.363      0.014
                           PC       27.693     38.538     21.180
     MOVE REQUEST PENDING  SD        0.001      0.001      0.000
                           PC        0.035      0.124      0.000
     NO PPU AVAILABLE      SD        0.000      0.000      0.000
                           PC        0.000      0.000      0.000
     EM TRANSFER IN PROG   SD        0.001      0.001      0.000
                           PC        0.125      0.209      0.048
     MTR CYCLE TIME        AV        0.392      0.436      0.364
                           SD        0.029      0.023      0.000
     MONITOR MODE - CPU 0  SD        0.020      0.013      0.002
                           PC        9.895     13.605      7.887
     MONITOR MODE - CPU 1  SD        0.000      0.000      0.000
                           PC        0.000      0.000      0.000
     SCHEDULER ACTIVE      SD        0.057      0.069      0.001
                           PC        9.248     19.136      5.067
     CHANNEL RESERVED
               CH 12       SD        0.034      0.068      0.000
                           PC        1.250      6.898      0.038
               CH 13       SD        0.168      0.275      0.003
                           PC       19.383     39.808      7.434
```

(12) These fast loop items are the remaining columns from page 4 of the example. Refer to note (6).

(13) The TOTAL column contains the total samples, average, standard deviation, and percentage for the entire ACPD run.

(14) The *MAX* column contains the maximum interval value for each row of preceding intervals for the entire ACPD run. Refer to note (7).

(15) The *MIN* column contains the minimum interval value for each row of preceding intervals for the entire ACPD run. Refer to note (8).

Figure 21-1. Example of TRACER Output

Summary File Format

The machine-readable summary file has two types of records, the header block record and the data block record. An *EOR* separates consecutive records.

The header block record contains the header data in an unpacked format.

Each data block record is divided into two equal length parts. The first part contains the average values of the data items. The second part contains the corresponding standard deviations of the data items.

Total and subtotal data and interval percentage data does not appear on the summary file. The loop sample times and snapshot data items have 0 standard deviations to simplify the summary file format.

Data Items Reported by TRACER

The data items reported by TRACER are described in the following paragraphs and are summarized in table 21-1. TRACER increments each item's counter when the item is in a given state, and periodically writes the contents of the counters to the data file for future processing by ACPD. When and how the counter for a given item is incremented is also discussed.

Suggestions are given to help you determine system performance. Performance may be affected by several areas within the system; therefore, a full analysis should be done prior to making any changes.

Table 21-1. Data Items

Fast Loop Items	Weighting Factor
Fast loop samples	None
PPs active	Number of PPs available
Move request pending	1[1]
No PP available	1
No CPP available	1
EM transfer in progress	1
MTR cycle time	100[2]
Monitor mode — CPU0	1
Monitor mode — CPU1	1
Scheduler active	1
Channel reserved	1
Channel active	1
Channel requested	1
Requests pending	100
Buffered I/O lists	Number of I/O buffers
Buffered I/O channel busy	1
CPU usage — CPUO	1
CPU usage — CPU1	1
Subsystem CPU usage	1

1. The average value will not be reported if the weighting factor is 1.

2. The percentage value will not be reported if the weighting factor is 100.

(Continued)

Table 21-1. Data Items *(Continued)*

Medium Loop Items	Weighting Factor
Medium loop samples	None
CPS in W status	Number of control points
CPS in X status	Number of control points
CPS in I status	Number of control points
Same move request	1
FL available	Available field length (machine size — CMR size)
User EM available	Available user extended memory field length
Noninteractive jobs	Number of EJT entries
Detached jobs	Number of EJT entries
Online jobs	Number of EJT entries
Preinitial job step	Number of EJT entries
Executing	Number of EJT entries
Scheduler rollout	Number of EJT entries
SCP rollin	Number of EJT entries
SCP rollout	Number of EJT entries
Timed/event rollout	Number of EJT entries
Interactive rollout	Number of EJT entries
Disabled rollout	Number of EJT entries
Suspended rollout	Number of EJT entries
Rollout file error	Number of EJT entries
EJT entries in use	Number of EJT entries
FL at control points	Available field length
FL at pseudo-control points	Available field length
FL in rollout queue	Available field length
EM memory at control point	User extended memory size/1000_8
EM memory at pseudo-control point	User extended memory size/1000_8
EM in rollout queue	User extended memory size/1000_8
Subsystem FL	Available field length

(Continued)

Table 21-1. Data Items *(Continued)*

Slow Loop Items	**Weighting Factor**
Slow loop samples	None
IAF users	Number of terminals defined
IAF pots allocated	100
IAF pots in use	IAF pots allocated
Queue files assigned	Number of QFT entries
Input files	Number of QFT entries
Print files	Number of QFT entries
Punch files	Number of QFT entries
Other queue files	Number of QFT entries
QFT entries in use	Number of QFT entries
Tape drives in use	Number of available tape drives
Tracks available	Maximum number of tracks for monitored device
CPPs active	Number of CPPs available
ISHARED table changes	None
ISHARED device updates	None
ISHARED seek time	None
ISHARED updating time	None

Fast Loop Items

The following paragraphs describe fast loop items. Fast loop items are continuously changing, so they should be sampled frequently.

Fast Loop Samples

TRACER increments the fast loop samples counter each time the fast loop items are sampled.

PPs Active

TRACER increments the PP active counter for each active PP it finds when scanning the PP communication area. Disabled PPs (those turned off at deadstart time) are not considered active. An active PP is one that has a nonzero input register. MTR (PP0) and DSD (PP1) are always counted as active PPs.

If this counter's percentage exceeds 80 percent and the no PP available counter is greater than 20 percent, you may need more PPs. Check the channel active counters for disks. If the disk channels range from 35 to 40 percent, you may not need additional PPs; instead, look at what is causing your disk channels to be so busy. If you can reduce their activity to below 30 percent, the percentages should also drop for the PPs active counter and the no PP available counter.

Examine which PPs are dedicated. If the percentage of floating PPs is low (about 20 percent of the total PPs configured), you may need more PPs.

Move Request Pending

TRACER increments the move request pending counter whenever it determines, from word CMCL of CMR, that a storage move request is outstanding.

If this counter's percentage ranges from 30 to 40 percent, the system is degrading. The following items could cause excessive storage movements.

- Are subsystems at the highest or lowest control points? If not, NOS may be moving them up or down in memory frequently in order to satisfy other user job memory requests. If a subsystem like NAM is being moved, its users may experience slow response time since a subsystem cannot execute while it is moving in memory.

- Try to stabilize the field lengths of your subsystems. For example, if TAF is not built correctly for your site's use, it may roll out and roll in quite often. Depending on your CM constraints, you may want TAF to roll out less frequently. You should evaluate each subsystem and place it at the correct control point.

- Are some user jobs changing their field length excessively? For example, jobs that open and close many files repeatedly cause CMM to adjust the user's field length, which causes other job movement within memory. This may save on field length but takes resources from the sytem, which increases system overhead. This type of job should be changed to be more efficient.

No PP Available

TRACER increments the no PP available counter each time it determines, from word PPAL of CMR, that there are no PPs available.

The percentage for this counter should be examined with the percentage for the PP active counter. If the PP active percentage is low (less than 50 percent) but you have counts of no PPs available, it means that you have peaks in your work load and this may be acceptable. However, if the PP active percentage is high (from 80 to 90 percent) and you have counts of no PPs available, you may need more PPs.

No CPP Available

TRACER increments the no CPP available counter each time it determines, from word PPAL of CMR, that there are no CPPs available.

EM Transfer in Progress

TRACER increments the extended memory transfer in progress counter whenever the S/C register (maintenance register for models 865 and 875) indicates an extended memory transfer is active. This data is available for CYBER 170 Computer Systems except models 176, 815, 825, 835, 845, and 855.

MTR Cycle Time

TRACER increments the MTR cycle time counter by the cycle time for the last MTR cycle. This item indicates how fast MTR is completing one complete scan of all PP output registers and processing those functions present.

The MTR cycle time should not exceed 0.5 seconds. If it does, MTR could miss updating the real time clock. Refer to MTR MXN TIME reported in the statistical data area from CMR (figure 21-1) for the worst case.

Monitor Mode — CPU0/CPU1

TRACER increments the monitor mode counter for CPU0/CPU1 whenever the S/C register (maintenance register for models 865 and 875) indicates that the CPU0/CPU1 is in monitor mode. This data is available for CYBER 170 Computer Systems except models 815, 825, 835, 845, and 855.

During monitor mode, the CPU is executing the code of CPUMTR, which performs NOS overhead. During program mode, CPUMTR is executing to perform a function that a user job requested and only CPUMTR has the capability or security to do.

When the monitor mode counter reaches 20 percent for each CPU on a dual CPU configuration or 30 percent for a single CPU machine, you should examine the data provided by PROBE. PROBE reports each monitor request for both monitor mode and program mode. Examine the requests to determine what functions were called in excess. As a basis of comparison, use a PROBE report for which system performance was good and the monitor mode statistics were normal.

Scheduler Active

TRACER increments the scheduler active counter whenever it determines that the job scheduler (1SJ) is active. The job scheduler is considered active when the scheduler active flag in word JSCL is set.

The job scheduler is called to process input queues and rollout queues. The scheduler searches all of the corresponding queues, starting with the last EJT or QFT entry scheduled to a control point. It schedules the highest priority jobs into execution. Queue length and decision processing determines the amount of time the scheduler spends. Limit the size of the queues to the amount shown in the EJT entries in use and the QFT entries in use statistics.

If the scheduler active counter reaches 70 to 90 percent, you may notice problems. The reason is that 1SJ is called to process EJT entries or QFT entries, not both. For example, if 1SJ spends all of its time processing EJT entries, no input queue jobs are processed. In this case, interactive users entering the system for the first time will never get started. Also, excessive rollin/rollout because of central memory constraints or CM/CP time slice expirations could cause the scheduler to be busy. Examine the SERVICE parameters; you might have to do some tuning.

Channel Reserved

TRACER maintains a channel reserved counter for each available channel or concurrent channel and increments a counter whenever its channel is logically reserved by the operating system. TRACER samples the channel status table for this data.

A PP program may have a channel reserved without actually using it to transfer data. This can cause a problem only if the channel has a high reserve percentage (from 80 to 100 percent) and the corresponding activity for the channel is low (less than 20 percent). The exceptions are those channels that are dedicated to PPs such as DSD, PIP, and 1HP.

Channel Active

TRACER maintains a channel active counter for each available channel and increments a counter whenever it detects that the channel is not inactive, as determined by an IJM PP instruction.

System performance is affected most by disk channel activity. A percentage of from 10 to 30 percent is considered normal. When this percentage reaches 35 to 40 percent, your system may experience performance degradation. High channel activity could be caused by the following factors. If you can scale down these problems, you may decrease channel activity and improve system performance; otherwise, you may need more disk controllers.

- Do all the disk channels show similar activity levels? If not, examine the attributes you have described for each disk unit. For example, what units contain permanent files, rollout files, a copy of the system, and temporary files? You may have to redistribute the work load by changing attributes. Determine which files are busy and the devices on which they reside using the PFCAT utility. You may find that altering device masks and/or moving files will lower the channel activity. You are looking for an even percentage of work performed by each of the disk channels.

  The system selects file residency based on several factors. If the file is a permanent file, the system uses the device masks you have set for your family. If it is a temporary file or rollout file, the system looks at the attributes set for each device and selects the best candidate based on the following.

  - First, the system looks at the number of tracks available on each potential device. If any of the devices have fewer tracks available than the low space threshold set by the THRESHOLD command, those devices will no longer be candidates for selection unless all of the other potential devices are also below their threshold levels.

  - Second, the system picks (from the remaining candidates) the device with the least activity.

- Are there an excessive number of rollouts? User jobs may be reaching the CM or CP time slice limits. Check your SERVICE parameters and examine the number of time slices reported in the statistical data area from CMR. The system may be rolling jobs in and out of the control points if your SERVICE limits are too small. By adjusting the limits, you should improve system performance.

- Are large user jobs or subsystems rolling in or out? This can be determined by watching the DSD rollout status display (R). One large job can cause problems if it uses too many resources. Redesigning or rescheduling the job may help.

- Examine the PROBE utility output. PROBE reports the PP programs that were loaded (program name, residency, and number of loads). Any program that is loading more than once per second, should be made central memory resident. If you are short of central memory, place these programs in extended memory.

Channel Requested

TRACER maintains a channel requested counter for each available channel or concurrent channel and increments a counter whenever there is an outstanding request for that channel. TRACER uses the channel status table to determine the channel requested status.

This counter tells you how often PP programs have requested a channel but were unable to reserve it because another PP program had it reserved. A normal count is from 20 to 50 percent. The exceptions are those channels that are dedicated to PPs such as DSD, PIP, and 1HP.

This can be used for debugging purposes when developong your own PP programs that require channels. Be sure that you follow the rules governing channel dialog.

Requests Pending

TRACER maintains a request pending counter for each mass storage device available and increments a counter by the number of outstanding requests on that device when the sampling occurs. The MST provides this information.

Ensure that the number of requests pending is evenly distributed between all units. If not, you may have to redistribute files (permanent, system, temporary, and rollout files) for the devices to obtain a more even distribution. Adding more disk units to a configuration will also aid in reducing the number of requests outstanding.

Buffered I/O Lists

TRACER maintains a counter for each type of buffered I/O buffer list (Empty, Data Written, Read, Write). The number of buffers assigned to each list is incremented based on the buffered I/O data tables.

An average of 25 percent of the buffers on the Empty and Data Written lists is normal. The Write list should contain from 2 to 3 percent of the buffers, but a higher percentage is acceptable. The Read list is normal at from 65 to 75 percent. If there seems to be a shortage of Empty and Data Written buffers, you can add more extended memory for I/O buffer space.

Buffered I/O Channel Busy

TRACER maintains a PP driver busy counter for each buffered I/O channel and increments a counter whenever a data transfer associated with that channel is in progress. For 885-4x disks, the data transfer occurs through the low-speed port to ESM rather than the channel; only control and status information transfers over the channel. TRACER uses the buffered I/O channel control table to determine the PP driver busy status.

The normal percentage is from 15 to 20 percent for a model 760 with two channels. This percentage may be higher for machines with slower CPUs.

CPU Usage

TRACER maintains a set of CPU usage counters for each CPU. There is a counter for each type of CPU use.

Counter	CPU Use
IDLE	CPU is not currently being used.
SYSTEM	CPU is being used by CPUMTR.
SUB-SYS	CPU is being used by a subsystem.
SYS ORG	CPU is being used by a system origin job (subsystems are not considered system origin).
USER	CPU is being used by a user program.

Each time the CPU status is sampled (in words CPAL and CPAL+1 in CMR), TRACER increments one of the CPU usage counters. It determines which counter to update by investigating the control point area to which the CPU is assigned.

Tune your system to get the most CPU usage for your users. Following are suggestions on how to do this.

- If you run a mixture of jobs, such as interactive and batch jobs, set the limit of executing batch jobs to a number that minimally impacts interactive users with respect to central memory and control points. You may have to assign batch jobs a higher queue priority so they essentially will be locked into the control points. Since interactive jobs tend to use the CPU for short durations, the batch jobs will be there using up whatever CPU time is left from the interactive jobs.

- On machines with big memories, it is usually advantageous to assign most of the memory to the users instead of using UEM as a rollout device. This helps in two ways: 1) rolling jobs to disk usually takes less CPU overhead than rolling jobs to UEM; 2) with more memory for the users, NOS performs less storage moves.

  However, if you are rolling jobs to 895 disks and your system is CPU-saturated, you may want to rollout jobs to UEM and use the ENABLE,DDP ROLLOUT PATH entry in the IPRDECK. This will make rollout processing slower but should reduce system CPU usage.

- If you run site-developed subsystems, examine the possibility of performing more of the work as disk activity rather than manipulating the data in central memory.

Subsystem CPU Usage

Subsystem CPU usage is a further breakdown of subsystems of the SUB-SYS CPU usage data. TRACER maintains a CPU usage counter for every subsystem. Whenever the SUB-SYS CPU usage counter is incremented, TRACER also increments the appropriate subsystem CPU usage counter.

Medium Loop Items

The following paragraphs describe medium loop items.

Medium Loop Samples

TRACER increments the medium loop samples counter each time the medium loop items are sampled.

CPS in W Status

TRACER increments the control points in waiting (W) status counter whenever a control point is found in W status (determined by STSW word of control point area). TRACER scans all control points during each medium loop cycle.

CPS in X Status

TRACER increments the control points in periodic or automatic recall (X) status counter whenever it finds a control point in X status (determined by STSW word of control point area). TRACER scans all control points during each medium loop cycle.

CPS in I Status

TRACER increments the control points in automatic recall (I) status counter whenever a control point is found in I status (determined by STSW word of control point area). TRACER scans all control points during each medium loop cycle.

Same Move Request

TRACER increments the same move request counter every time the move request pending for the current medium loop cycle is the same as the previous medium loop cycle. Word CMCL of CMR is used to determine this status.

Use this counter to determine if NOS is having trouble moving jobs in central memory. A value of from 5 to 10 percent is not unreasonable. A higher percentage indicates that you probably are short of memory. Examine the FL available counter.

FL Available

TRACER increments the FL available counter by the amount of available FL during the current medium loop cycle. Word ACML of CMR determines the available FL.

Compare this counter with the average FL in the rollout queue (FL in the rollout queue divided by the average number of jobs in the queue). On a busy system, the average amount of FL available should be less than that required for the average job in the rollout queue. You may need to consider any jobs that are in the rollout queue but are not schedulable, such as NAM supervisory applications (CS, NS, and NVF). On a system that is not heavily used, the FL available counter is of little value since there should always be memory available.

User EM Available

TRACER increments the user extended memory available counter by the amount of available user extended memory FL during the current medium loop cycle. Word AECL of CMR determines the available user extended memory FL.

Compare this number to the amount of extended memory at control points and rollout queues to determine if you have enough user extended memory.

Noninteractive Jobs

TRACER maintains a count of noninteractive jobs by monitoring the number of EJT entries with a connection status of NICS.

Detached Jobs

TRACER maintains a count of detached interactive jobs by monitoring the number of EJT entries with a connection status of DTCS.

Online Jobs

TRACER maintains a count of online jobs by monitoring the number of EJT entries with a connection status of OLCS.

Preinitial Job Step

TRACER maintains a count of preinitial jobs for each service class by monitoring the number of EJT entries with a job status of PRJS. A job is in this state after it has been assigned to an EJT entry but before it has been assigned to a control point for the first time.

Executing

TRACER maintains a count of executing jobs for each service class by monitoring the number of EJT entries with a job status of EXJS.

Scheduler Rollout

TRACER maintains a count of jobs rolled out by the job scheduler for each service class by monitoring the number of EJT entries with a job status of ROJS.

SCP Rollin

TRACER maintains a count of jobs rolled in by system control point processing for each service class by monitoring the number of EJT entries with a job status of SIJS.

SCP Rollout

TRACER maintains a count of jobs rolled out by system control point processing for each service class by monitoring the number of EJT entries with a job status of SOJS.

Timed/Event Rollout

TRACER maintains a count of jobs in a timed/event rollout queue for each service class by monitoring the number of EJT entries with a job status of TOJS.

Interactive Rollout

TRACER maintains a count of jobs rolled out by interactive input/output processing for each service class by monitoring the number of EJT entries with a job status of IOJS.

Disabled Rollout

TRACER maintains a count of disabled jobs rolled out for each service class by monitoring the number of EJT entries with a job status of DOJS.

Suspended Rollout

TRACER maintains a count of suspended jobs rolled out for each service class by monitoring the number of EJT entries with a job status of SUJS.

Rollout File Error

TRACER maintains a count of jobs with rollout file errors for each service class by monitoring the number of EJT entries with a job status of ERJS.

EJT Entries in Use

TRACER maintains a count of the number of EJT entries in use for all jobs during the specified time interval. Each EJT entry is four central memory words long. If you are short of memory, you can gain some memory by decreasing the number of EJT entries.

FL at Control Points

TRACER maintains an FL at control point counter for each service class. When an active control point is found, the service class is determined and the appropriate counter is incremented by the amount of FL assigned to that control point (determined by FLSW of control point area). TRACER scans all control points during the medium loop cycle.

Use this count to determine which service class of jobs is using the most memory. For example, if you have poor interactive job response and notice that the batch service class is using the most memory, it may help to restrict batch jobs to a smaller central memory time slice using the CM parameter of the SERVICE entry in the IPRDECK.

FL at Pseudo-control Points

TRACER maintains an FL at pseudo-control point counter for each service class. When an active pseudo-control point is found, the service class is determined and the appropriate counter is incremented by the amount of FL assigned to that pseudo-control point (determined by FLSW of pseudo-control point area). TRACER scans all pseudo-control points during the medium loop cycle.

FL in Rollout Queue

TRACER maintains an FL in rollout queue counter for each service class. When a rollout file is found in the queue, and it is not assigned to a control point, the amount of FL it will require (determined from the EJT) when it is rolled into a control point is added to the appropriate counter. TRACER scans all EJT entries during the medium loop cycle.

Use this count in conjunction with the FL available counter to determine if you have a central memory shortage.

EM Memory at Control Points

TRACER maintains an extended memory at control point counter for each service class. When an active control point is found, the service class is determined and the appropriate counter is incremented by the amount of extended memory assigned to that control point (determined by ECSW of control point area). TRACER scans all control points during the medium loop cycle.

Use this count in conjunction with the user EM available counter and the EM in rollout queue counter to determine if you have an extended memory shortage.

EM Memory at Pseudo-control Points

Use this count in conjunction with the user EM available counter and the EM in rollout queue counter to determine if you have an extended memory shortage.

EM in Rollout Queue

TRACER maintains an extended memory in rollout queue counter for each service class. When a rollout file is found in the queue, and it is not assigned to a control point, the amount of extended memory FL it will require when rolled in (determined from the EJT) is added to the appropriate counter. TRACER scans all EJT entries during the medium loop cycle.

Use this count in conjunction with the user EM available counter and the EM memory at CP counter to determine if you have an extended memory shortage.

Subsystem FL

TRACER maintains a subsystem FL counter for every possible subsystem. When a subsystem is found at a control point, the amount of FL assigned to that subsystem is added to the appropriate counter. TRACER scans all control points during the medium loop cycle.

Slow Loop Items

The following paragraphs describe slow loop items.

Slow Loop Samples

TRACER increments the slow loop samples counter each time the slow loop items are sampled.

IAF Users

TRACER increments the IAF users counter by the number of users connected to IAF during the slow loop cycle. IAF must be active for this data to be collected. Word VANL of IAF FL determines the number of users.

IAF Pots Allocated

TRACER increments the pots allocated counter by the number of pots that are currently available for use, whether they are being used or not. Word VPAL of IAF FL determines the number of pots allocated.

IAF Pots in Use

TRACER increments the pots in use counter by the number of pots currently assigned to a connection. Word VPUL of IAF FL determines the number of pots in use.

Queue Files Assigned

TRACER maintains a count of the number of queued files assigned to control points by monitoring QFT entries.

Input Files

TRACER maintains a count of the number of input files for each service class by monitoring QFT entries.

Print Files

TRACER maintains a count of the number of print files for each service class by monitoring QFT entries.

Punch Files

TRACER maintains a count of the number of punch files for each service class by monitoring QFT entries.

Other Queue Files

TRACER maintains a count of the number of other queued files not assigned to control points for each service class by monitoring QFT entries. This includes any queued file that is not an input, print, or punch file.

QFT Entries in Use

TRACER maintains a count of the number of QFT entries in use for all jobs during the specified time interval. Each QFT entry is four central memory words long. If you are short of memory, you can gain some memory by decreasing the number of QFT entries.

Tape Drives in Use

TRACER maintains two tape drives in use counters; one for 7-track and one for 9-track drives. A tape drive is considered to be in use if an EST entry indicates it is logically turned on, and it is assigned to a job. TRACER increments the appropriate counter for each drive found in use.

Tracks Available

TRACER maintains a tracks available counter for each mass storage equipment and adds the number of available tracks (tracks not currently assigned to a file) for a device to the appropriate counter for each mass storage device found in the EST. TRACER obtains this information from TDGL of the MST.

CPPs Active

TRACER increments the CPP active counter for each active CPP it finds when scanning the PP communication area. Disabled CPPs (those turned off at deadstart time) are not considered active. An active CPP is one that has a nonzero input register.

ISHARED Table Changes

TRACER maintains a count of the number of monitor function calls that force a table update in an ISHARED environment.

ISHARED Device Updates

TRACER maintains a count of the number of times the device-resident tables are updated in an ISHARED environment.

ISHARED Seek Time

TRACER monitors the ISHARED seek time by recording the total time spent reading the first sector during a table update sequence.

ISHARED Updating Time

TRACER monitors the ISHARED updating time by recording the total time spent updating tables and the maximum time and minimum time to update the tables for both single and simultaneous operations.

Statistical Summary

The following paragraphs describe various statistics collected by TRACER.

MTR Maximum Time

TRACER reports the maximum time, in microseconds, that MTR waits for a CPU exchange to occur after being initiated by a monitor exchange jump (MXN).

Direct Moves

Tracer reports the number of direct storage moves of central memory.

Worst Case MTR Cycle Time

TRACER reports the maximum time, in milliseconds, that it takes MTR to make a complete cycle; processing all PP output register requests and performing its other system functions.

Missed Clock Updates

TRACER reports the number of times MTR was not able to update the clock.

Extended Memory Moves

TRACER reports the number of storage moves of extended memory performed by CPUMTR.

Central Memory Moves

TRACER reports the number of storage moves of central memory performed by CPUMTR.

Total Rollouts

TRACER reports the total number of jobs rolled out by the job rollout routine (1RO).

Secondary Rollouts

TRACER reports the number of jobs rolled out to secondary rollout devices by the job rollout routine (1RO).

Total Sectors Rolled

TRACER reports the total number of sectors of mass storage used by the job rollout routine (1RO) for rollouts.

Secondary Sec Rolled

TRACER reports the number of sectors of mass storage used by the job rollout routine (1RO) for rollouts to secondary rollout devices.

Rollouts/User Limits

TRACER reports the number of jobs rolled out due to a time limit or SRU limit detected by the job scheduler (1SJ).

Time Slices

TRACER reports the number of jobs whose scheduling is set to the lower bound for its service class due to an expired CPU or CM time slice.

PP Priority Exchanges

TRACER reports the number of priority exchange requests issued by PPs and CPPs. A priority exchange is issued after three unsuccessful attempts to perform a monitor exchange jump (MXN).

No Comm Buffer Avail

TRACER reports the number of times CPUMTR could not use a communications buffer because all buffers were full.

No PCP Available

TRACER reports the number of times a pseudo-control point was not available.

EJT Scans

TRACER reports the number of times the job scheduler (1SJ) scans the executing job table (EJT) to schedule a job to a control point.

Schedulable Jobs

TRACER reports the total number of schedulable jobs. This is determined by adding the number of schedulable jobs in EJT on each EJT scan.

Jobs Preempted

TRACER reports the number of jobs rolled out so the job scheduler (1SJ) could schedule a higher priority job.

Jobs Scheduled

TRACER reports the number of jobs scheduled to a control point.

Scheduled No Constraints

TRACER reports the number of jobs scheduled with no service class, central memory, or extended memory constraints imposed.

Insufficient CM Scans

TRACER reports the number of EJT scans for which a schedulable job was found, but could not be scheduled due to insufficient central memory.

Insufficient EM Scans

TRACER reports the number of EJT scans for which a schedulable job was found, but could not be scheduled due to insufficient extended memory.

No Control Point Scans

TRACER reports the number of EJT scans for which a schedulable job was found, but could not be scheduled because all control points were busy.

PROBE Utility

The PROBE utility measures the following:

- Number of times a PP or CPP routine was loaded.
- Number of CIO RA+1 requests by function number.
- Number of PP requests to CPUMTR by function number.
- Number of MTR requests to CPUMTR by function number.
- Statistical data accumulated in CMR includes items such as number of sectors rolled, number of rollouts, and number of pseudo-rollouts.
- Input/output statistics for 885-42 and 895 Disk Storage Subsystems showing the number of sectors transferred and control buffer tables transferred.

PROBE data gathering is selectable at deadstart time by an IPRDECK entry. SYSEDIT resets the PROBE data tables to zeros.

The PROBE utility generates a report from the data collected by the system. PROBE analyzes data either from system tables or from a binary file containing data from a previous PROBE run. An IPRDECK entry ENABLE,PROBE must be specified at deadstart time to allow the system to collect the data. (Refer to section 3, Deadstart Decks, for more information.)

Format:

```
PROBE,p1,p2,...,pn.
```

p_i	Description
B=readfile	Binary file to be read. Default is STATS.
L=outfile	Report file. Default is OUTPUT.
L=0	No report is to be generated.
LO=opt	Sort option for PP program load information. Default is F.

opt	Description
A	Sort data in alphabetic order.
F	Sort data by frequency of loads.
R	Sort data by location and frequency of loads.

pi	Description
OP=opt	Processing option. Default is P.

opt	Description
C	Perform R option functions and clear system tables after they are read.
P	Generate report from binary file specified by the B parameter. This binary file has been created by a previous PROBE run with OP=C or OP=R specified.
R	Read system tables, and write binary file and report file as specified.

pi	Description
P=writefile	Binary file to be written. Default is STATS.
P=0	No binary file is to be written.
R	Rewind binary files before and after operation. Default is no rewind.

NOTE

The file names must be unique. If you specify OP=C or OP=R, then you may not specify P=0.

Figure 21-2 is an example of the PROBE output file format.

When looking at the PP program load information reported on the PROBE output, check to see if it meets these suggested performance requirements:

- PP programs called at least once every second should be central memory resident.
- PP programs called at least once every 2 seconds are good candidates for alternate residency using extended memory.
- PP programs called every 3 seconds or less should be disk resident.

You can change a PP program's residency using the SYSEDIT command (refer to section 19, SYSEDIT).

PROBE VERSION 1.0. yy/mm/dd. hh.mm.ss. PAGE 1

PROBE VERSION 1.0	yy/mm/dd.	hh.mm.ss.
START OF SAMPLE INTERVAL	yy/mm/dd.	hh.mm.ss.
END OF SAMPLE INTERVAL	yy/mm/dd.	hh.mm.ss.

MONITOR REQUESTS

	PROGRAM MODE CPU0	MONITOR MODE CPU0	MONITOR MODE CPU1
CHGM(21)	0	0	0
HNGM(22)	0	0	0
(23)	0	0	0
AFAM(24)	0	7177	0
DLKM(25)	0	0	0
DTKM(26)	131214	189955	0
RTCM(27)	129969	129969	0
STBM(30)	0	121555	0
VMSM(31)	9108	9108	0
ACTM(32)	1133	20059	0
BFMM(33)	0	0	0
CKSM(34)	435	435	0
CSTM(35)	34405	34670	0
ECSM(36)	0	0	0
PIOM(37)	0	0	0
RDCM(40)	3170	3170	0
ABTM(41)	0	76	0
BIOM(42)	0	0	0
BMIM(43)	0	0	0
CCAM(44)	0	77321	0
CEFM(45)	0	832	0
CPRM(46)	0	0	0
DCPM(47)	0	25696	0
DEQM(50)	0	41425	0
DFMM(51)	0	36827	0
DPPM(52)	0	1108876	0
EATM(53)	0	48694	0
JACM(54)	0	59846	0
LDAM(55)	0	844946	0
LMSM(56)	0	0	0
MTRM(57)	0	2495	0
PLFM(60)	0	76738	0
RCLM(61)	0	447037	0
RCPM(62)	0	1110	0
RECM(63)	0	46638	0
REQM(64)	0	41426	0
RJSM(65)	0	675	0
RLMM(66)	0	30666	0
ROCM(67)	0	15092	0
RPNM(70)	0	0	0
RPPM(71)	0	106311	0
RSJM(72)	0	15481	0
SCDM(73)	0	62387	0
SFBM(74)	0	70802	0
SJCM(75)	0	22237	0
SPLM(76)	0	1358228	0
TDAM(77)	0	17566	0
TGPM(100)	0	11532	0
TIOM(101)	0	21228	0
TSEM(102)	0	37099	0
UADM(103)	0	583735	0
UTEM(104)	0	1473766	0
VFLM(105)	0	0	0
VFPM(106)	0	14518	0
VSAM(107)	0	13180	0
(110)	0	0	0
SUBTOTAL		7230584	0
TOTAL	309434	7230584	

Figure 21-2. Example of PROBE Output

(Continued)

(Continued)

```
PROBE VERSION 1.0.                                              yy/mm/dd. hh.mm.ss.  PAGE    2

     MTR REQUESTS

                      CPU0      CPU1
       ARTF( 1)      27487         0
       IARF( 2)          0         0
       CSLF( 3)      23220         0
       RCLF( 4)    6134559         0
       MFLF( 5)      66297         0
       SCSF( 6)          0         0
       SMSF( 7)          0         0
       CMSF(10)          0         0
       PRQF(11)      41638         0
       ACSF(12)      93425         0
       PCXF(13)          0         0
       ARMF(14)          0         0
       ----(15)          0         0
       MFEF(16)          0         0
     SUBTOTAL      6386626         0
     TOTAL         6386626


     PROGRAM MODE

                      CPU0
       MSTF( 1)      23220
       PDMF( 2)          0
       PMRF( 3)          0
       MECF( 4)          0
          TOTAL      23220


     MTR PERFORMANCE PARAMETERS
       WORST CASE *MXN* TIME =                             751
       WORST CASE CYCLE TIME =                               6
       MISSED CLOCK UPDATES =                                2

     CPUMTR PERFORMANCE PARAMETERS
       EM STORAGE MOVES =                                    0
       CM STORAGE MOVES =                                24173
       DIRECT MOVES =                                    20540
       COMMUNICATIONS BUFFER NOT AVAILABLE =                 0
       PP PRIORITY EXCHANGE REQUESTS =                       0
       PSEUDO-CONTROL POINT NOT AVAILABLE                  351

     1RO PERFORMANCE PARAMETERS
       NUMBER OF ROLLOUTS =                              18393
       PSEUDO-ROLLOUTS =                                  4000
       PSEUDO-CONTROL POINT ROLLOUTS =                    1000
       ROLLOUTS TO SECONDARY DEVICES =                       0
       NUMBER OF SECTORS ROLLED =                      2918049
       SECTORS ROLLED TO SECONDARY DEVICES =                 0

     1SJ PERFORMANCE PARAMETERS
       EJT SCANS =                                       76193
       SCHEDULABLE JOBS=                                 21917
       JOBS PREEMPTED =                                     95
       JOBS SCHEDULED =                                  17866
       JOBS SCHEDULED NO CONSTRAINTS =                       0
       EJT SCANS WITH INSUFFICIENT CM =                    130
       EJT SCANS WITH INSUFFICIENT EM =                      0
       EJT SCANS WITH NO CONTROL POINT =                     2
       ROLLOUTS FOR RESOURCE LIMITS =                        1
       TIME SLICES =                                       460
```

Figure 21-2. Example of PROBE Output

(Continued)

(Continued)

```
PROBE VERSION 1.0.                                              yy/mm/dd. hh.mm.ss.  PAGE    3

     CIO REQUESTS

     FUNCTION              CALLS   FUNCTION             CALLS

     RPHR(000)              3015   READ(010)            154978
     WRITE(014)            83438   READSKP(020)           4907
     WRITER(024)           53751   WRITEF(034)            5237
     BKSP(040)               465   REWIND(050)           55857
     ****(54)                128   UNLOAD(060)             459
     RETURN(070)           30345   OPEN(100)              2200
     OPEN(104)              2643   EVICT(114)             1653
     OPEN(120)              2602   CLOSE(130)             1356
     OPEN(140)               753   OPEN(144)              4885
     CLOSE(150)              582   OPEN(160)               543
     CLOSE(170)             5339   CLOSE(174)              465
     READCW(200)           15476   WRITECW(204)           7361
     READLS(210)            1235   REWRITE(214)          13111
     REWRITER(224)         11907   RPHRLS(230)           25137
     SKIPF(240)             9784   READNS(250)             400
     READEI(600)            5893   SKIPB(640)             4386

     TOTAL CIO FUNCTIONS PROCESSED     510291
```

Figure 21-2. Example of PROBE Output

(Continued)

(Continued)

```
PROBE VERSION 1.0.                                              yy/mm/dd. hh.mm.ss.  PAGE    4

     PPU PROGRAM LOADS

     NAME   LOC   LOADS          NAME   LOC   LOADS
     CPM    CMR  605181          PIP    DSK       0
     1MS    CMR  460275          PNC    DSK       0
     4MB    CMR  243588          SBP    DSK       0
     4MD    CMR  169670          SLL    DSK       0
     4ME    CMR  155651          SMP    DSK       0
     QAC    CMR   83410          TMG    DSK       0
     3Q3    CMR   83372          VEJ    DSK       0
     1SJ    CMR   73838          VER    DSK       0
     3AE    CMR   70341          WRM    DSK       0
     LFM    CMR   69611          XIS    DSK       0
     1AJ    CMR   47141          X26    DSK       0
     3LB    CMR   46964          0CI    DSK       0
     TCS    CMR   46406          0IP    DSK       0
     3QU    DSK   46381          0MC    DSK       0
     LDR    CMR   43575          0MD    DSK       0
     1MA    CMR   39939          0SD    CMR       0
     0BF    CMR   37181          0SI    DSK       0
     3QS    CMR   36988          0TD    DSK       0
     4MC    CMR   31223          1FA    DSK       0
     3NW    CMR   29968          1HP    DSK       0
     1IO    CMR   27405          1IS    DSK       0
     NDR    CMR   26489          1KB    DSK       0
     RHH    DSK   25371          1LC    DSK       0
     1MT    CMR   24435          1LT    CMR       0
     3ME    CMR   24435          1MF    DSK       0
     0DF    CMR   24317          1MR    DSK       0
     3MF    CMR   23257          1PC    DSK       0
     3CD    CMR   21678          1PD    DSK       0
     3AF    CMR   19990          1PL    DSK       0
     3MJ    CMR   19709          1PR    DSK       0
     QFM    DSK   19637          1RM    DSK       0
     RPV    CMR   18598          1RP    DSK       0
     1RO    CMR   18282          1RU    DSK       0
     1RI    CMR   18262          1TM    DSK       0
     1MI    CMR   14294          1TN    DSK       0
     3LD    CMR   12765          1TP    DSK       0
     SFM    CMR   12584          1TS    DSK       0
     3SX    CMR   10536          1VP    DSK       0
     STD    DSK    9223          2DC    DSK       0
     4SD    DSK    9222          2DD    DSK       0
     3AD    DSK    8614          2IE    DSK       0
     2MA    DSK    8501          2IM    DSK       0
     TLX    CMR    8444          2IN    DSK       0
     1DD    CMR    7995          2IP    DSK       0
     4MF    CMR    7600          2IQ    DSK       0
     3RP    CMR    7357          2KA    DSK       0
     3RH    CMR    7342          2KB    DSK       0
     PFM    CMR    6961          2KC    DSK       0
     3PA    CMR    5122          2LD    DSK       0
     3LC    CMR    5085          2MB    DSK       0
     3AI    CMR    4722          2ME    DSK       0
```

Figure 21-2. Example of PROBE Output

(Continued)

(Continued)

PROBE VERSION 1.0. yy/mm/dd. hh.mm.ss. PAGE 5

3LE	CMR	4024	2MF	DSK	0
3PC	CMR	3960	2MN	DSK	0
3QD	DSK	3959	2MR	DSK	0
LDQ	CMR	3790	2MZ	DSK	0
3PD	CMR	3619	2NT	DSK	0
3QA	DSK	2800	2NU	DSK	0
3CR	CMR	2749	2PT	DSK	0
3CS	CMR	2748	2PV	DSK	0
1TO	CMR	2296	2PX	DSK	0
2TO	CMR	2203	2RP	DSK	0
3PE	CMR	2080	2RU	DSK	0
0AV	CMR	1887	2SB	DSK	0
1TA	DSK	1867	2SC	DSK	0
3CB	CMR	1822	2SE	DSK	0
1CK	DSK	1699	2SF	DSK	0
3PR	CMR	1682	2SG	DSK	0
3MQ	CMR	1670	2SI	DSK	0
3CC	DSK	1643	2TM	DSK	0
3SV	DSK	1598	2TN	DSK	0
3PN	CMR	1489	3AG	DSK	0
0RP	CMR	1325	3AH	CMR	0
LDD	CMR	1301	3BB	DSK	0
3TC	DSK	1200	3BC	DSK	0
0FA	CMR	1162	3BD	DSK	0
DSP	DSK	1161	3BE	DSK	0
3DB	DSK	1120	3BF	DSK	0
0DQ	DSK	1086	3CA	DSK	0
3RG	DSK	1029	3CV	DSK	0
2NW	DSK	995	3FB	DSK	0
3MI	CMR	981	3IC	DSK	0
0AU	DSK	947	3ID	DSK	0
3AK	DSK	941	3IE	DSK	0
3AA	DSK	934	3IM	DSK	0
3PQ	CMR	923	3IN	DSK	0
2NR	DSK	922	3IO	DSK	0
3AJ	DSK	918	3IP	DSK	0
3QW	DSK	904	3IQ	DSK	0
3QX	DSK	903	3KA	DSK	0
3QO	DSK	903	3KB	DSK	0
3PH	CMR	791	3KC	DSK	0
CVL	DSK	747	3LA	DSK	0
3QC	DSK	685	3MA	DSK	0
SFP	CMR	605	3MB	DSK	0
3AB	DSK	596	3MD	DSK	0
1DS	CMR	521	3MK	CMR	0
3DA	DSK	491	3MM	DSK	0
1ML	DSK	490	3MR	CMR	0
3TE	DSK	480	3MU	DSK	0
3PI	DSK	479	3PM	DSK	0
1DL	CMR	454	3PP	DSK	0
CMS	DSK	429	3QR	DSK	0
4DB	DSK	426	3Q1	DSK	0
4DD	DSK	426	3RF	DSK	0
3SA	DSK	425	3RU	DSK	0
3MO	CMR	383	3SU	DSK	0
3NM	DSK	382	3SZ	DSK	0
3ML	DSK	354	3TA	DSK	0
3MH	DSK	350	4DF	DSK	0

Figure 21-2. Example of PROBE Output

(Continued)

(Continued)

```
PROBE VERSION 1.0.                                            yy/mm/dd. hh.mm.ss.  PAGE    6

        3PF    CMR    348            4DI    DSK     0
        2MD    DSK    338            4DJ    DSK     0
        4MA    CMR    313            4DK    DSK     0
        0VJ    DSK    310            4DZ    DSK     0
        0VU    DSK    310            4IM    DSK     0
        3PK    DSK    245            4IO    DSK     0
        3MC    CMR    237            4IP    DSK     0
        3AL    DSK    235            4IQ    DSK     0
        2MC    DSK    216            4KB    DSK     0
        9AA    DSK    215            4KC    DSK     0
        SSH    DSK    210            4KD    DSK     0
        3MG    DSK    193            4MG    DSK     0
        3TB    DSK    188            4NB    DSK     0
        PFU    DSK    177            4NC    DSK     0
        9EA    DSK    168            4NM    DSK     0
        0RF    DSK    163            4RA    DSK     0
        2SD    DSK    141            4RB    DSK     0
        3MT    DSK    138            4RC    DSK     0
        2NS    DSK    136            4RD    DSK     0
        3MP    CMR    133            4RE    DSK     0
        3PB    DSK    129            4RF    DSK     0
        3PJ    DSK    126            4RG    DSK     0
        3AM    DSK     91            5BA    DSK     0
        9BA    DSK     83            5BB    DSK     0
        3MN    DSK     82            5BC    DSK     0
        0RT    DSK     74            5BD    DSK     0
        3MS    DSK     67            5CU    DSK     0
        2NQ    DSK     63            5CV    DSK     0
        T76    DSK     58            5D1    DSK     0
        6DI    CMR     58            5D2    DSK     0
        0PT    CMR     51            5D3    DSK     0
        QAP    DSK     48            5FF    DSK     0
        NLD    DSK     46            5FH    DSK     0
        3DC    DSK     36            5IA    DSK     0
        3FA    DSK     35            5IC    DSK     0
        3Q2    DSK     32            5ID    DSK     0
        DIS    DSK     29            5IG    DSK     0
        3BA    DSK     24            5IH    DSK     0
        3BG    DSK     24            5LC    DSK     0
        3IA    DSK     24            5LL    DSK     0
        3IB    DSK     24            5MA    DSK     0
        5IE    DSK     24            5MB    DSK     0
        3QB    DSK     22            5MC    DSK     0
        1CD    DSK     21            5MD    DSK     0
        3PL    DSK     19            5ME    DSK     0
        3AC    DSK     15            5MG    DSK     0
        2DA    DSK     13            5MR    DSK     0
        2SA    DSK     13            5MS    DSK     0
        7CI    CMR     13            5MT    DSK     0
        7DI    CMR     13            5MU    DSK     0
        7EI    CMR     13            5MW    DSK     0
        7EL    CMR     13            5SE    DSK     0
        7EM    CMR     13            5SU    DSK     0
        7EN    CMR     13            5SV    DSK     0
        7EO    CMR     13            5S1    DSK     0
        7EP    CMR     13            5S2    DSK     0
        7SI    CMR     13            5XA    DSK     0
        4MH    DSK     12            5XB    DSK     0
```

Figure 21-2. Example of PROBE Output

(Continued)

(Continued)

PROBE VERSION 1.0. yy/mm/dd. hh.mm.ss. PAGE 7

IMS	DSK	10	5XC	DSK	0
3PG	DSK	10	5XD	DSK	0
3QY	DSK	9	5XE	DSK	0
3QZ	DSK	9	5XF	DSK	0
2SH	DSK	8	5XG	DSK	0
3CE	DSK	8	541	DSK	0
0PI	DSK	7	55X	DSK	0
1MB	DSK	7	56X	DSK	0
1MP	CMR	7	57X	DSK	0
2DB	DSK	6	58F	DSK	0
3DD	DSK	6	58H	DSK	0
4DA	DSK	6	58X	DSK	0
4DC	DSK	6	7ER	CMR	0
4DG	DSK	6	7FI	CMR	0
026	DSK	5	7GI	CMR	0
9GA	DSK	5	7HI	CMR	0
3RI	DSK	4	711	CMR	0
2NX	DSK	3	7JI	CMR	0
2NY	DSK	3	7KI	CMR	0
3AN	DSK	3	7LI	CMR	0
3PO	DSK	3	7SE	CMR	0
3QT	DSK	3	7WI	CMR	0
3QV	DSK	3	8XA	DSK	0
7EQ	CMR	3	8XB	DSK	0
7ES	CMR	3	8XC	DSK	0
0ST	DSK	2	9CA	CMR	0
3SY	DSK	2	9CB	CMR	0
0TI	DSK	1	9CC	CMR	0
0TJ	DSK	1	9CD	CMR	0
3CK	DSK	1	9CE	CMR	0
3RJ	DSK	1	9CF	CMR	0
3SW	DSK	1	9CG	CMR	0
3TD	DSK	1	9CH	CMR	0
ADC	DSK	0	9CI	CMR	0
BAT	DSK	0	9CJ	CMR	0
CPD	DSK	0	9CK	CMR	0
CUX	DSK	0	9CL	CMR	0
DDF	DSK	0	9CM	CMR	0
DOG	DSK	0	9CN	CMR	0
DS1	DSK	0	9CO	CMR	0
ELM	DSK	0	9CP	CMR	0
EYE	DSK	0	9CQ	CMR	0
FDP	DSK	0	9CR	CMR	0
HFM	DSK	0	9CS	CMR	0
LIF	DSK	0	9JN	DSK	0
MDD	DSK	0	9VA	DSK	0
MLD	DSK	0	9WA	DSK	0
MP3	DSK	0			
PIM	DSK	0			

NUMBER OF ASR PROGRAMS = 0
NUMBER OF CMR PROGRAMS = 123
NUMBER OF DSK PROGRAMS = 313
NUMBER OF PLD ENTRIES = 436

Figure 21-2. Example of PROBE Output

(Continued)

(Continued)

```
PROBE VERSION 1.0.                                                          yy/mm/dd. hh.mm.ss.  PAGE    8

     I/O STATISTICS FOR   DB006:

          SECTORS            CBT-S
        TRANSFERRED       TRANSFERRED      READS      WRITES
                  1            1-40B          0           1
                2-3          41B-140B         0          23
                4-7         141B-340B         0           0
            10B-17B         341B-740B         0           0
            20B-37B         741B-1740B        0           0
            40B-77B        1741B-3740B        0           0
          100B-177B        3741B-7740B        0           0
          200B-MORE        7741B-MORE         0           0

         TOTALS                               0          24


     I/O STATISTICS FOR   DB007:

          SECTORS            CBT-S
        TRANSFERRED       TRANSFERRED      READS      WRITES
                  1            1-40B         19           1
                2-3          41B-140B         4           1
                4-7         141B-340B         0           0
            10B-17B         341B-740B         0           2
            20B-37B         741B-1740B        0           0
            40B-77B        1741B-3740B        0           0
          100B-177B        3741B-7740B        0           0
          200B-MORE        7741B-MORE         0           0

         TOTALS                              23           4


     I/O STATISTICS FOR   DB010:

          SECTORS            CBT-S
        TRANSFERRED       TRANSFERRED      READS      WRITES
                  1            1-40B          0           1
                2-3          41B-140B         0           0
                4-7         141B-340B         0           0
            10B-17B         341B-740B         0           3
            20B-37B         741B-1740B        0           0
            40B-77B        1741B-3740B        0           0
          100B-177B        3741B-7740B        0           0
          200B-MORE        7741B-MORE         0           0

         TOTALS                               0           4
```

Figure 21-2. Example of PROBE Output

Appendixes

Character Sets A

A character set is composed of graphic and control characters. A code set is a set of codes used to represent each character within a character set.

A graphic character may be displayed at a terminal or printed by a line printer. Examples are the characters A through Z and the digits 0 through 9. A control character initiates, modifies, or stops a control operation. An example is the backspace character that moves the terminal carriage or cursor back one space. Although a control character is not a graphic character, a terminal may produce a graphic representation when it receives a control character.

All references within this manual to the ASCII character set or the ASCII code set refer to the character set and code set defined in the American National Standard Code for Information Interchange (ASCII, ANSI Standard X3.4-1977). References in this manual to the ASCII character set do not necessarily refer to the ASCII code set.

NOS supports the following character sets.

- CDC graphic 64- (or 63-) character set.
- ASCII 128-character set.
- ASCII graphic 64- (or 63-) character set.
- ASCII graphic 95-character set.

Each installation selects either the 64-character set or the 63-character set. The differences between the two are described next, under Character Set Anomalies. Any reference in this appendix to the 64-character set implies either the 63- or 64-character set, unless otherwise stated.

NOS supports the following code sets.

- 6-bit display code.
- 6/12-bit display code.
- 7-bit ASCII code.

Display code is a set of 6-bit codes from 00_8 to 77_8.

The 6/12-bit display code is a combination of 6-bit codes and 12-bit codes. The 6-bit codes are 00_8 through 77_8, excluding 74_8 and 76_8. (Refer to Character Set Anomalies, next, for the interpretation of the 00_8 and 63_8 codes.) The 12-bit codes begin with either 74_8 or 76_8 and are followed by a 6-bit code. Thus, 74_8 and 76_8 are considered escape codes and are never used as 6-bit codes within the 6/12-bit display code set. The 12-bit codes are 7401_8, 7402_8, 7404_8, 7407_8, and 7601_8 through 7677_8. All other 12-bit codes ($74xx_8$ and 7600_8) are undefined.

The 7-bit ASCII code (as defined by ANSI Standard X3.4-1977) is right-justified in a 12-bit byte. Assuming that the bits are numbered from the right starting with 0, bits 0 through 6 contain the ASCII code, bits 7 through 10 contain zeros, and bit 11 distinguishes the 0000_8 code from the end-of-line byte. The 7-bit codes are 0001_8 through 0177_8 and 4000_8.

Character Set Anomolies

NOS interprets two codes differently when the installation selects the 63-character set rather than the 64-character set. In tables A-1, A-2, and A3, the codes for the colon and percent graphic characters in the 64-character set are unshaded; the codes for the colon and percent graphic characters in the 63-character set are shaded.

If an installation uses the 63-character set, the colon graphic character is always represented by a 63_8 code, and the 00_8 code is undefined. However, if the installation uses the 64-character set, output of 6/12-bit display codes 7404_8 or 6-bit display code 00_8 produces a colon. In ASCII mode for interactive jobs, a colon can be input only as a 7404_8 6/12-bit display code.

When using either the 63- or 64-character set, the use of undefined 6/12-bit display codes in output files produces unpredictable results and should be avoided.

On input, NOS recognizes alternate 029 punch codes of 11-0 for the right bracket (]) and 12-0 for the left bracket ([). The alternate codes support the COBOL sign overpunch convention and are not recommended for other uses. Refer to the COBOL 5 Reference Manual.

Also, two 00_8 codes may be confused with an end-of-line byte and should be avoided (refer to the NOS 2 Reference Set, Volume 3 for further explanation).

Translation of 7-bit ASCII to 6-bit display code causes character set folding from the 128-character ASCII set to the 63- or 64-character ASCII subset. The following special character substitutions occur:

7-Bit ASCII		6-Bit Display Code		7-Bit ASCII	
Code	Character	Code	Character	Code	Character
0140	`	74	@	0100	@
0173	{	61	[	0133	[
0174	|	75	\	0134	\
0175	}	62	]	0135	]
0176	~	76	^	0136	^

Character Set Tables

This appendix includes character set tables for interactive jobs, batch jobs, and jobs involving magnetic tapes. Table A-1 is for interactive jobs, and table A-2 is for batch jobs. Table A-3 is a conversion table used to cross-reference 7-bit ASCII codes and 6/12-bit display codes and to convert ASCII codes from octal to hexadecimal.

Tables A-4, A-5, and A-6 list the magnetic tape codes and their display code equivalents.

The character set tables are designed so that you can find the character represented by a code (such as in a dump) or find the code that represents a character. To find the character represented by a code, look up the code in the column listing the appropriate code set and then find the character on that line in the column listing the appropriate character set. To find the code that represents a character, you first look up the character and then find the code on the same line in the appropriate column.

Interactive Jobs

Table A-1 shows the character sets and code sets available to you at an ASCII code terminal. When in NORMAL mode (specified by the NORMAL command), NOS displays the ASCII graphic 64-character set and interprets all input and output as display code. When in ASCII mode (specified by the ASCII command), NOS displays the ASCII 128-character set and interprets all input and output as 6/12-bit display code.

To determine the octal or hexadecimal ASCII code for a character, refer to table A-3. (Certain terminal definition commands require specification of an ASCII code.)

On output, the US code is reserved for network use and defined as an end-of-line. Use of this character, except in transparent mode, causes incorrect formatting and possible loss of output characters.

Batch Jobs

Table A-2 lists the CDC graphic 64-character set, the ASCII graphic 64-character set, and the ASCII graphic 95-character sets. It also lists the code sets and card punch codes (O26 and O29) that represent the characters.

The 64-character sets use display code as their code set; the 95-character set uses 7-bit ASCII code. The 95-character set is composed of all the characters in the ASCII 128-character set that can be printed at a line printer (refer to Jobs Using Line Printers, next). Only 7-bit ASCII code files can be printed using the ASCII graphic 95-character set. To print a 6/12-bit display code file (usually created by an interactive job in ASCII mode), you must convert the file to 7-bit ASCII code. To do this, you enter the FCOPY command (refer to the NOS 2 Reference Set, Volume 3). The 95-character set is represented by 7-bit ASCII codes 0040_8 through 0176_8.

Jobs Using Line Printers

The batch character set printed depends on the print train used on the line printer to which the file is sent (refer to the ROUTE command in the NOS 2 Reference Set, Volume 3). The following are the print trains corresponding to each of the batch character sets.

Character Set	Print Train	PSU Printer Print Band
CDC graphic 64-character set	596-1	-
ASCII graphic 64-character set	596-5	530-1
ASCII graphic 95-character set	596-6	530-2

The characters of the default 596-1 print train are listed in the table A-2 column labeled CDC Graphic (64 Character); the 596-5 print train characters are listed in the table A-2 column labeled ASCII Graphic (64 Character); and the 596-6 print train characters are listed in the table A-2 column labeled ASCII Graphic (95 Character).

If a transmission error occurs when printing a line, the system prints the line again. The CDC graphic print train prints a concatenation symbol (⌐) in the first printable column of the repeated listing of the line. The ASCII print trains print an underline (_) instead of the concatenation symbol.

If an unprintable character exists in a line (that is, a 7-bit ASCII code outside the range 0040_8 through 0176_8), the number sign (#) appears in the first printable column of a print line, and a space replaces the unprintable character.

To route and correctly print a 6/12-bit display code file on a line printer with the ASCII graphic 95-character set, you must convert the 6/12-bit display code file to a 7-bit ASCII code file with the FCOPY command (refer to the NOS 2 Reference Set, Volume 3). The resulting 7-bit ASCII file can be routed to a line printer (refer to the ROUTE command in the NOS 2 Reference Set, Volume 3) but cannot be output at an interactive terminal.

Table A-1. Character Sets for Interactive Jobs

ASCII Graphic (64-Character)	ASCII Character (128 Character)	6-Bit Display Code	6/12-Bit Display Code	7-Bit ASCII Code
: colon		00[1]		
Display code 00 is undefined at sites using the 63-character set.				
A	A	01	01	0101
B	B	02	02	0102
C	C	03	03	0103
D	D	04	04	0104
E	E	05	05	0105
F	F	06	06	0106
G	G	07	07	0107
H	H	10	10	0110
I	I	11	11	0111
J	J	12	12	0112
K	K	13	13	0113
L	L	14	14	0114
M	M	15	15	0115
N	N	16	16	0116
O	O	17	17	0117
P	P	20	20	0120
Q	Q	21	21	0121
R	R	22	22	0122
S	S	23	23	0123
T	T	24	24	0124
U	U	25	25	0125
V	V	26	26	0126
W	W	27	27	0127
X	X	30	30	0130
Y	Y	31	31	0131
Z	Z	32	32	0132
0	0	33	33	0060
1	1	34	34	0061
2	2	35	35	0062
3	3	36	36	0063
4	4	37	37	0064

1. The interpretation of this character or code may depend on its context. Refer to Character Set Anomalies elsewhere in this appendix.

(Continued)

Table A-1. Character Sets for Interactive Jobs *(Continued)*

ASCII Graphic (64-Character)	ASCII Character (128 Character)	6-Bit Display Code	6/12-Bit Display Code	7-Bit ASCII Code
5	5	40	40	0065
6	6	41	41	0066
7	7	42	42	0067
8	8	43	43	0070
9	9	44	44	0071
+ plus	+ plus	45	45	0053
- dash	- dash	46	46	0055
* asterisk	* asterisk	47	47	0052
/ slant	/ slant	50	50	0057
(opening parenthesis	(opening parenthesis	51	51	0050
) closing parenthesis	) closing parenthesis	52	52	0051
$ dollar sign	$ dollar sign	53	53	0044
= equal	= equal	54	54	0075
space	space	55	55	0040
, comma	, comma	56	56	0054
. period	. period	57	57	0056
# number sign	# number sign	60	60	0043
[opening bracket	[opening bracket	61	61	0133
] closing bracket	] closing bracket	62	62	0135
% percent sign	% percent sign	63	63[1]	0045
: colon	: colon	63	63	0072
" quote	" quote	64	64	0042
_ underline	_ underline	65	65	0137
! exclamation point	! exclamation point	66	66	0041
& ampersand	& ampersand	67	67	0046
' apostrophe	' apostrophe	70	70	0047
? question mark	? question mark	71	71	0077
< less than	< less than	72	72	0074
> greater than	> greater than	73	73	0076
@ commercial at	@ commercial at	74[1]	7401	0100
\ reverse slant	\ reverse slant	75	75	0134
^ circumflex	^ circumflex	76	7402	0136
; semicolon	; semicolon	77	77	0073

1. The interpretation of this character or code may depend on its context. Refer to Character Set Anomalies elsewhere in this appendix.

(Continued)

Table A-1. Character Sets for Interactive Jobs *(Continued)*

ASCII Graphic (64-Character)	ASCII Character (128 Character)	6-Bit Display Code	6/12-Bit Display Code	7-Bit ASCII Code
	: colon		7404	0072
	% percent		7404	0045
	` grave accent	74[1]	7407	0140
	a		7601	0141
	b		7602	0142
	c		7603	0143
	d		7604	0144
	e		7605	0145
	f		7606	0146
	g		7607	0147
	h		7610	0150
	i		7611	0151
	j		7612	0152
	k		7613	0153
	l		7614	0154
	m		7615	0155
	n		7616	0156
	o		7617	0157
	p		7620	0160
	q		7621	0161
	r		7622	0162
	s		7623	0163
	t		7624	0164
	u		7625	0165
	v		7626	0166
	w		7627	0167
	x		7630	0170
	y		7631	0171
	z		7632	0172
	{ opening brace	61[1]	7633	0173
	| vertical line	75[1]	7634	0174
	} closing brace	62[1]	7635	0175
	~ tilde	76[1]	7636	0176
	DEL		7637	0177

1. The interpretation of this character or code may depend on its context. Refer to Character Set Anomalies elsewhere in this appendix.

(Continued)

Table A-1. Character Sets for Interactive Jobs *(Continued)*

ASCII Graphic (64-Character)	ASCII Character (128 Character)	6-Bit Display Code	6/12-Bit Display Code	7-Bit ASCII Code
	NUL		7640	4000
	SOH		7641	0001
	STX		7642	0002
	ETX		7643	0003
	EOT		7644	0004
	ENQ		7645	0005
	ACK		7646	0006
	BEL		7647	0007
	BS		7650	0010
	HT		7651	0011
	LF		7652	0012
	VT		7653	0013
	FF		7654	0014
	CR		7655	0015
	SO		7656	0016
	SI		7657	0017
	DLE		7660	0020
	DC1		7661	0021
	DC2		7662	0022
	DC3		7663	0023
	DC4		7664	0024
	NAK		7665	0025
	SYN		7666	0026
	ETB		7667	0027
	CAN		7670	0030
	EM		7671	0031
	SUB		7672	0032
	ESC		7673	0033
	FS		7674	0034
	GS		7675	0035
	RS		7676	0036
	US[1]		7677	0037

1. Reserved for network use. Refer to Character Set Tables in this appendix.

Table A-2. Character Sets for Batch Jobs

CDC Graphic (64 Character)	ASCII Graphic (64 Character)	ASCII Graphic (95 Character)	6-Bit Display Code	6/12-Bit Display Code	7-Bit ASCII Code	Punch Code 026	Punch Code 029
: colon[1]	: colon[1]		00[1]			8-2	8-2
Display code 00 is undefined at sites using the 63-character set.							
A	A	A	01	01	0101	12-1	12-1
B	B	B	02	02	0102	12-2	12-2
C	C	C	03	03	0103	12-3	12-3
D	D	D	04	04	0104	12-4	12-4
E	E	E	05	05	0105	12-5	12-5
F	F	F	06	06	0106	12-6	12-6
G	G	G	07	07	0107	12-7	12-7
H	H	H	10	10	0110	12-8	12-8
I	I	I	11	11	0111	12-9	12-9
J	J	J	12	12	0112	11-1	11-1
K	K	K	13	13	0113	11-2	11-2
L	L	L	14	14	0114	11-3	11-3
M	M	M	15	15	0115	11-4	11-4
N	N	N	16	16	0116	11-5	11-5
O	O	O	17	17	0117	11-6	11-6
P	P	P	20	20	0120	11-7	11-7
Q	Q	Q	21	21	0121	11-8	11-8
R	R	R	22	22	0122	11-9	11-9
S	S	S	23	23	0123	0-2	0-2
T	T	T	24	24	0124	0-3	0-3
U	U	U	25	25	0125	0-4	0-4
V	V	V	26	26	0126	0-5	0-5
W	W	W	27	27	0127	0-6	0-6
X	X	X	30	30	0130	0-7	0-7
Y	Y	Y	31	31	0131	0-8	0-8
Z	Z	Z	32	32	0132	0-9	0-9
0	0	0	33	33	0060	0	0
1	1	1	34	34	0061	1	1
2	2	2	35	35	0062	2	2
3	3	3	36	36	0063	3	3
4	4	4	37	37	0064	4	4

1. The interpretation of this character or code may depend on its context. Refer to Character Set Anomalies elsewhere in this appendix.

(Continued)

Table A-2. Character Sets for Batch Jobs *(Continued)*

CDC Graphic (64 Character)	ASCII Graphic (64 Character)	ASCII Graphic (95 Character)	6-Bit Display Code	6/12-Bit Display Code	7-Bit ASCII Code	Punch Code 026	Punch Code 029
5	5	5	40	40	0065	5	5
6	6	6	41	41	0066	6	6
7	7	7	42	42	0067	7	7
8	8	8	43	43	0070	8	8
9	9	9	44	44	0071	9	9
+ plus	+ plus	+ plus	45	45	0053	12	12-8-6
- dash	- dash	- dash	46	46	0055	11	11
* asterisk	* asterisk	* asterisk	47	47	0052	11-8-4	11-8-4
/ slant	/ slant	/ slant	50	50	0057	0-1	0-1
(opening parenthesis	(opening parenthesis	(opening parenthesis	51	51	0050	0-8-4	12-8-5
) closing parenthesis	) closing parenthesis	) closing parenthesis	52	52	0051	12-8-4	11-8-5
$ dollar sign	$ dollar sign	$ dollar sign	53	53	0044	11-8-3	11-8-3
= equal	= equal	= equal	54	54	0075	8-3	8-6
space	space	space	55	55	0040	no punch	no punch
, comma	, comma	, comma	56	56	0054	0-8-3	0-8-3
. period	. period	. period	57	57	0056	12-8-3	12-8-3
≡ equivalence	# number sign	# number sign	60	60	0043	0-8-6	8-3
[opening bracket	[opening bracket	[opening bracket	61	61	0133	8-7	12-8-2[1]
] closing bracket	] closing bracket	] closing bracket	62	62	0135	0-8-2	11-8-2[1]
% percent sign[1]	% percent sign[1]	% percent sign[1]	63[1]	63[1]	0045	8-6	0-8-4
: colon	: colon	: colon	63	63	0072	8-2	8-2
≠ not equal	" quote	" quote	64	64	0042	8-4	8-7
↱ concatenation	_ underline	_ underline	65	65	0137	0-8-5	0-8-5
∨ logical OR	! exclamation point	! exclamation point	66	66	0041	11-0	12-8-7
∧ logical AND	& ampersand	& ampersand	67	67	0046	0-8-7	12

1. The interpretation of this character or code may depend on its context. Refer to Character Set Anomalies elsewhere in this appendix.

(Continued)

Table A-2. Character Sets for Batch Jobs *(Continued)*

CDC Graphic (64 Character)	ASCII Graphic (64 Character)	ASCII Graphic (95 Character)	6-Bit Display Code	6/12-Bit Display Code	7-Bit ASCII Code	Punch Code 026	Punch Code 029
↑ superscript	' apostrophe	' apostrophe	70	70	0047	11-8-5	8-5
↓ subscript	? question mark	? question mark	71	71	0077	11-8-6	0-8-7
< less than	< less than	< less than	72	72	0074	12-0	12-8-4
> greater than	> greater than	> greater than	73	73	0076	11-8-7	0-8-6
≤ less or equal	@ commercial at		74			8-5	8-4
≥ greater or equal	\ reverse slant	\ reverse slant	75	75	0134	12-8-5	0-8-2
¬ logical NOT	^ circumflex		76			12-8-6	11-8-7
; semicolon	; semicolon	; semicolon	77	77	0073	12-8-7	11-8-6
		@ commercial at	74[1]	7401	0100		
		^ circumflex	76[1]	7402	0136		
		: colon[1]		7404[1]	0072		
		% percent sign		7404	0045		
		` grave accent	74[1]	7407	0140		
		a		7601	0141		
		b		7602	0142		
		c		7603	0143		
		d		7604	0144		
		e		7605	0145		
		f		7606	0146		
		g		7607	0147		

1. The interpretation of this character or code may depend on its context. Refer to Character Set Anomalies elsewhere in this appendix.

(Continued)

Table A-2. Character Sets for Batch Jobs *(Continued)*

CDC Graphic (64 Character)	ASCII Graphic (64 Character)	ASCII Graphic (95 Character)	6-Bit Display Code	6/12-Bit Display Code	7-Bit ASCII Code	Punch Code 026	Punch Code 029
		h		7610	0150		
		i		7611	0151		
		j		7612	0152		
		k		7613	0153		
		l		7614	0154		
		m		7615	0155		
		n		7616	0156		
		o		7617	0157		
		p		7620	0160		
		q		7621	0161		
		r		7622	0162		
		s		7623	0163		
		t		7624	0164		
		u		7625	0165		
		v		7626	0166		
		w		7627	0167		
		x		7630	0170		
		y		7631	0171		
		z		7632	0172		
		{ opening brace	61[1]	7633	0173		
		| vertical line	75[1]	7634	0174		
		} closing brace	62[1]	7635	0175		
		~ tilde	76[1]	7636	0176		

1. The interpretation of this character or code may depend on its context. Refer to Character Set Anomalies elsewhere in this appendix.

Table A-3. ASCII to 6/12-Bit Display Code Conversion

ASCII Character (128 Character)	7-Bit ASCII Code Octal	7-Bit ASCII Code Hexadecimal	6/12-Bit Display Code
NUL	4000	00	7640
SOH	0001	01	7641
STX	0002	02	7642
ETX	0003	03	7643
EOT	0004	04	7644
ENQ	0005	05	7645
ACK	0006	06	7646
BEL	0007	07	7647
BS	0010	08	7650
HT	0011	09	7651
LF	0012	0A	7652
VT	0013	0B	7653
FF	0014	0C	7654
CR	0015	0D	7655
SO	0016	0E	7656
SI	0017	0F	7657
DLE	0020	10	7660
DC1	0021	11	7661
DC2	0022	12	7662
DC3	0023	13	7663
DC4	0024	14	7664
NAK	0025	15	7665
SYN	0026	16	7666
ETB	0027	17	7667
CAN	0030	18	7670
EM	0031	19	7671
SUB	0032	1A	7672
ESC	0033	1B	7673
FS	0034	1C	7674
GS	0035	1D	7675
RS	0036	1E	7676
US[1]	0037	1F	7677[1]
space	0040	20	55
! exclamation point	0041	21	66
" quote	0042	22	64
# number sign	0043	23	60
$ dollar sign	0044	24	53
% percent sign[2]	0045	25	63[2]
% percent sign	0045	25	7404
& ampersand	0046	26	67
' apostrophe	0047	27	70

1. Reserved for network use. Refer to Character Set Tables in this appendix.

2. The interpretation of this character or code may depend on its context. Refer to Character Set Anomalies in this appendix.

(Continued)

Table A-3. ASCII to 6/12-Bit Display Code Conversion *(Continued)*

ASCII Character (128 Character)	7-Bit ASCII Code Octal	7-Bit ASCII Code Hexadecimal	6/12-Bit Display Code
(opening parenthesis	0050	28	51
) closing parenthesis	0051	29	52
* asterisk	0052	2A	47
+ plus	0053	2B	45
, comma	0054	2C	56
- dash	0055	2D	46
. period	0056	2E	57
/ slant	0057	2F	50
0	0060	30	33
1	0061	31	34
2	0062	32	35
3	0063	33	36
4	0064	34	37
5	0065	35	40
6	0066	36	41
7	0067	37	42
8	0070	38	43
9	0071	39	44
: colon[1]	0072	3A	7404[1]
: colon	0072	3A	63
; semicolon	0073	3B	77
< less than	0074	3C	72
= equal	0075	3D	54
> greater than	0076	3E	73
? question mark	0077	3F	71
@ commercial at	0100	40	7401
A	0101	41	01
B	0102	42	02
C	0103	43	03
D	0104	44	04
E	0105	45	05
F	0106	46	06
G	0107	47	07
H	0110	48	10
I	0111	49	11
J	0112	4A	12
K	0113	4B	13
L	0114	4C	14
M	0115	4D	15
N	0116	4E	16
O	0117	4F	17

1. The interpretation of this character or code may depend on its context. Refer to Character Set Anomalies in this appendix.

(Continued)

Table A-3. ASCII to 6/12-Bit Display Code Conversion *(Continued)*

ASCII Character (128 Character)	7-Bit ASCII Code Octal	7-Bit ASCII Code Hexadecimal	6/12-Bit Display Code
P	0120	50	20
Q	0121	51	21
R	0122	52	22
S	0123	53	23
T	0124	54	24
U	0125	55	25
V	0126	56	26
W	0127	57	27
X	0130	58	30
Y	0131	59	31
Z	0132	5A	32
[opening bracket	0133	5B	61
\ reverse slant	0134	5C	75
] closing bracket	0135	5D	62
^ circumflex	0136	5E	7402
_ underline	0137	5F	65
` grave accent	0140	60	7407
a	0141	61	7601
b	0142	62	7602
c	0143	63	7603
d	0144	64	7604
e	0145	65	7605
f	0146	66	7606
g	0147	67	7607
h	0150	68	7610
i	0151	69	7611
j	0152	6A	7612
k	0153	6B	7613
l	0154	6C	7614
m	0155	6D	7615
n	0156	6E	7616
o	0157	6F	7617
p	0160	70	7620
q	0161	71	7621
r	0162	72	7622
s	0163	73	7623
t	0164	74	7624
u	0165	75	7625
v	0166	76	7626
w	0167	77	7627

(Continued)

Table A-3. ASCII to 6/12-Bit Display Code Conversion *(Continued)*

ASCII Character (128 Character)	7-Bit ASCII Code Octal	7-Bit ASCII Code Hexadecimal	6/12-Bit Display Code
x	0170	78	7630
y	0171	79	7631
z	0172	7A	7632
{ opening brace	0173	7B	7633
| vertical line	0174	7C	7634
} closing brace	0175	7D	7635
~ tilde	0176	7E	7636
DEL	0177	7F	7637

Jobs Using Magnetic Tape

Coded data to be copied from disk to magnetic tape is assumed to be represented in display code. NOS converts the data to external BCD code when writing a coded 7-track tape and to ASCII or EBCDIC code (as specified on the tape assignment command) when writing a coded 9-track tape.

Because only 63 characters can be represented in 7-track even parity, one of the 64 display codes is lost in conversion to and from external BCD code. Figure A-1 shows the differences in conversion depending on the character set (63 or 64) that the system uses. The ASCII character for the specified character code is shown in parentheses. The output arrow shows how the 6-bit display code changes when it is written on tape in external BCD. The input arrow shows how the external BCD code changes when the tape is read and converted to 6-bit display code.

63-Character Set

6-Bit Display Code		External BCD		6-Bit Display Code
00		16 (%)		00
33 (0)	Output →	12 (0)	Input →	33 (0)
63 (:)		12 (0)		33 (0)

64-Character Set

6-Bit Display Code		External BCD		6-Bit Display Code
00 (:)		12 (0)		33 (0)
33 (0)	Output →	12 (0)	Input →	33 (0)
63 (%)		16 (%)		63 (%)

Figure A-1. Conversion Differences

If a lowercase ASCII or EBCDIC code is read from a 9-track coded tape, it is converted to its uppercase 6-bit display code equivalent. To read or write lowercase ASCII or EBCDIC characters, you must assign the tape in binary mode and use FCOPY to read or write the tape.

Tables A-4 and A-5 show the character set conversion for 9-track tapes. Table A-4 lists the conversions to and from the 7-bit ASCII character code and 6-bit display code. Table A-5 lists the conversions between the EBCDIC character code and the 6-bit display code. Table A-6 shows the character set conversions between external BCD and 6-bit display code for 7-track tapes.

Table A-4. Nine-Track ASCII Coded Tape Conversion

7-Bit ASCII Code (Hex)	7-Bit ASCII Char[1]	7-Bit ASCII Code (Hex)	7-Bit ASCII Char[2]	6-Bit Display Code Char	6-Bit Display Code Code (Octal)
20	space	00	NUL	space	55
21	!	7D	}	!	66
22	"	02	STX	"	64
23	#	03	ETX	#	60
24	$	04	EOT	$	53
25	%	05	ENQ	%	63
25	%	05	ENQ	space[3]	55
26	&	06	ACK	&	67
27	'	07	BEL	'	70
28	(	08	BS	(	51
29	)	09	HT	)	52
2A	*	0A	LF	*	47
2B	+	0B	VT	+	45
2C	,	0C	FF	,	56
2D	-	0D	CR	-	46
2E	.	0E	SO	.	57
2F	/	0F	SI	/	50
30	0	10	DLE	0	33
31	1	11	DC1	1	34
32	2	12	DC2	2	35
33	3	13	DC3	3	36
34	4	14	DC4	4	37
35	5	15	NAK	5	40
36	6	16	SYN	6	41
37	7	17	ETB	7	42
38	8	18	CAN	8	43
39	9	19	EM	9	44
3A	:	1A	SUB	:	00
6-Bit display code 00 is undefined at sites using the 63-character set.					
3A	:	1A	SUB	:	63
3B	;	1B	ESC	;	77
3C	<	7B	{	<	72
3D	=	1D	GS	=	54

1. When these characters are copied from/or to a tape, the characters remain the same but the codes change from one code set to the other.

2. These characters do not exist in 6-bit display code. Therefore, when the characters are copied from a tape, each 7-bit ASCII character is changed to an alternate 6-bit display code character. The corresponding codes are also changed. Example: When the system copies a lowercase a, 61_{16}, from tape, it writes an uppercase A, 01_8.

3. A 6-bit display code space always translates to a 7-bit ASCII space.

(Continued)

Table A-4. Nine-Track ASCII Coded Tape Conversion *(Continued)*

7-Bit ASCII Code (Hex)	7-Bit ASCII Char[1]	7-Bit ASCII Code (Hex)	7-Bit ASCII Char[2]	6-Bit Display Code Char	6-Bit Display Code Code (Octal)
3E	>	1E	RS	>	73
3F	?	1F	US	?	71
40	@	60	`	@	74
41	A	61	a	A	01
42	B	62	b	B	02
43	C	63	c	C	03
44	D	64	d	D	04
45	E	65	e	E	05
46	F	66	f	F	06
47	G	67	g	G	07
48	H	68	h	H	10
49	I	69	i	I	11
4A	J	6A	j	J	12
4B	K	6B	k	K	13
4C	L	6C	l	L	14
4D	M	6D	m	M	15
4E	N	6E	n	N	16
4F	O	6F	o	O	17
50	P	70	p	P	20
51	Q	71	q	Q	21
52	R	72	r	R	22
53	S	73	s	S	23
54	T	74	t	T	24
55	U	75	u	U	25
56	V	76	v	V	26
57	W	77	w	W	27
58	X	78	x	X	30
59	Y	79	y	Y	31
5A	Z	7A	z	Z	32
5B	[	1C	FS	[	61
5C	\	7C	|	\	75
5D	]	01	SOH	]	62
5E	^	7E	~	^	76
5F	_	7F	DEL	_	65

1. When these characters are copied from/or to a tape, the characters remain the same but the codes change from one code set to the other.

2. These characters do not exist in 6-bit display code. Therefore, when the characters are copied from a tape, each 7-bit ASCII character is changed to an alternate 6-bit display code character. The corresponding codes are also changed. Example: When the system copies a lowercase a, 61_{16}, from tape, it writes an uppercase A, 01_8.

Table A-5. Nine-Track EBCDIC Coded Tape Conversion

EBCDIC Code (Hex)	EBCDIC Char[1]	EBCDIC Code (Hex)	EBCDIC Char[2]	6-Bit Display Code Char	6-Bit Display Code Code (Octal)
40	space	00	NUL	space	55
4A	¢	1C	IFS	[	61
4B	.	0E	SO	.	57
4C	<	C0	{	<	72
4D	(	16	BS	(	51
4E	+	0B	VT	+	45
4F	|	D0	}	!	66
50	&	2E	ACK	&	67
5A	!	01	SOH	]	62
5B	$	37	EOT	$	53
5C	*	25	LF	*	47
5D	)	05	HT	)	52
5E	;	27	ESC	;	77
5F	-	A1	~	^	76
60	-	0D	CR	-	46
61	/	0F	SI	/	50
6B	,	0C	FF	,	56
6C	%	2D	ENQ	%	63
6C	%	2D	ENQ	space[3]	55
6D	_	07	DEL	_	65
6E	>	1E	IRS	>	73
6F	?	1F	IUS	?	71
7A	:	3F	SUB	:	00
6-Bit display code 00 is undefined at sites using the 63-character set.					
7A	:	3F	SUB	:	63

1. When these characters are copied from/or to a tape, the characters remain the same (except EBCDIC codes 4A, 4F, 5A, and 5F) but the codes change from one code set to the other.

2. These characters do not exist in 6-bit display code. Therefore, when the characters are copied from a tape, each EBCDIC character is changed to an alternate 6-bit display code character. The corresponding codes are also changed. Example: When the system copies a lowercase a, 81_{16}, from tape, it writes an uppercase A, 01_8.

3. All EBCDIC codes not listed translate to 6-bit display code 55_8 (space). A 6-bit display code space always translates to an EBCDIC space.

(Continued)

Table A-5. Nine-Track EBCDIC Coded Tape Conversion *(Continued)*

EBCDIC Code (Hex)	EBCDIC Char[1]	EBCDIC Code (Hex)	EBCDIC Char[2]	6-Bit Display Code Char	6-Bit Display Code Code (Octal)
7B	#	03	ETX	#	60
7C	@	79	\	@	74
7D	'	2F	BEL	'	70
7E	=	1D	IGS	=	54
7F	"	02	STX	"	64
C1	A	81	a	A	01
C2	B	82	b	B	02
C3	C	83	c	C	03
C4	D	84	d	D	04
C5	E	85	e	E	05
C6	F	86	f	F	06
C7	G	87	g	G	07
C8	H	88	h	H	10
C9	I	89	i	I	11
D1	J	91	j	J	12
D2	K	92	k	K	13
D3	L	93	l	L	14
D4	M	94	m	M	15
D5	N	95	n	N	16
D6	O	96	o	O	17
D7	P	97	p	P	20
D8	Q	98	q	Q	21
D9	R	99	r	R	22
E0	\	6A	|	\	75
E2	S	A2	s	S	23
E3	T	A3	t	T	24
E4	U	A4	u	U	25
E5	V	A5	v	V	26
E6	W	A6	w	W	27
E7	X	A7	x	X	30

1. When these characters are copied from/or to a tape, the characters remain the same (except EBCDIC codes 4A, 4F, 5A, and 5F) but the codes change from one code set to the other.

2. These characters do not exist in 6-bit display code. Therefore, when the characters are copied from a tape, each EBCDIC character is changed to an alternate 6-bit display code character. The corresponding codes are also changed. Example: When the system copies a lowercase a, 81_{16}, from tape, it writes an uppercase A, 01_8.

(Continued)

Table A-5. Nine-Track EBCDIC Coded Tape Conversion *(Continued)*

EBCDIC Code (Hex)	EBCDIC Char[1]	EBCDIC Code (Hex)	EBCDIC Char[2]	6-Bit Display Code Char	6-Bit Display Code Code (Octal)
E8	Y	A8	y	Y	31
E9	Z	A9	z	Z	32
F0	0	10	DLE	0	33
F1	1	11	DC1	1	34
F2	2	12	DC2	2	35
F3	3	13	TM	3	36
F4	4	3C	DC4	4	37
F5	5	3D	NAK	5	40
F6	6	32	SYN	6	41
F7	7	26	ETB	7	42
F8	8	18	CAN	8	43
F9	9	19	EM	9	44

1. When these characters are copied from/or to a tape, the characters remain the same (except EBCDIC codes 4A, 4F, 5A, and 5F) but the codes change from one code set to the other.

2. These characters do not exist in 6-bit display code. Therefore, when the characters are copied from a tape, each EBCDIC character is changed to an alternate 6-bit display code character. The corresponding codes are also changed. Example: When the system copies a lowercase a, 81_{16}, from tape, it writes an uppercase A, 01_8.

Table A-6. Seven-Track Coded Tape Conversions

External BCD	ASCII Character	6-Bit Display Code (Octal)
01	1	34
02	2	35
03	3	36
04	4	37
05	5	40
06	6	41
07	7	42
10	8	43
11	9	44
12[1]	0	33
13	=	54
14	"	64
15	@	74
16[1]	%	63
17	[	61
20	space	55
21	/	50
22	S	23
23	T	24
24	U	25
25	V	26
26	W	27
27	X	30
30	Y	31
31	Z	32
32	]	62
33	,	56
34	(	51
35	_	65
36	#	60
37	&	67

1. As explained previously in this appendix, conversion of these codes depends on whether the tape is being read or written.

(Continued)

Table A-6. Seven-Track Coded Tape Conversions *(Continued)*

External BCD	ASCII Character	6-Bit Display Code (Octal)
40	-	46
41	J	12
42	K	13
43	L	14
44	M	15
45	N	16
46	O	17
47	P	20
50	Q	21
51	R	22
52	!	66
53	$	53
54	*	47
55	'	70
56	?	71
57	>	73
60	+	45
61	A	01
62	B	02
63	C	03
64	D	04
65	E	05
66	F	06
67	G	07
70	H	10
71	I	11
72	<	72
73	.	57
74	)	52
75	\	75
76	^	76
77	;	77

Glossary B

A

Access Category

See File Access Category and System Access Categories.

Access Code

A hardware/software security code assigned to each NAD on the network. A NAD may communicate only with other NADs having matching codes.

Access Level

A property of each file, job, and equipment on a secured system that is used to indicate the sensitivity of information in the file or job, or the sensitivity of information that can be processed by the equipment. On a secured system, there are up to eight access levels corresponding to increasing levels of sensitivity; you are authorized to access some or all of those levels. Refer also to Equipment Access Levels, File Access Level, Job Access Level, and System Access Levels.

Access Level Limits

See Job Access Level Limits.

Account Dayfile

A dayfile that provides a history of system usage over the life of the account. It provides information necessary for accurate billing and system usage analysis.

ACN

See Application Connection Number.

AFD Utility

A dayfile dumping utility that dumps all or selected parts of the account dayfile to produce a listing.

AIP Trace Utility

See Application Interface Program Trace Utility.

Allocation Summary Table (AST)

A table that contains information used by the allocation algorithm to select the cartridges on which a file will reside in an MSE environment.

Allocation Unit (AU)

In an MSE environment, an allocation unit is the smallest allocatable portion of a cartridge. Each AU consists of 14 data strips; each cartridge has 1931 AUs.

Alternate Storage

The storage of permanent file data on external media other than mass storage such as tape alternate storage or MSE. When a file resides on alternate storage, the file's permanent file catalog (PFC) entry and permit data still reside on disk, but the disk space occupied by the file data can be released.

Alternate Storage Address (ASA)

The pointers in a file's PFC entry that point to the location of the file data on tape alternate storage or MSE.

Application

See Application Program.

Application Connection Number (ACN)

A number assigned by the NAM program to identify a particular logical connection within an application.

Application Interface Program (AIP) Trace Utility

A utility that produces a trace file of the messages transferred between IAF and NAM. The information contained in this trace can be useful in tracking network problems and in debugging application programs.

Application Program

A program resident in a host computer that provides an information storage, retrieval, and/or processing service to a remote user via the data communication network and the Network Access Method. Application programs use the system control point feature of NOS to communicate with the Network Access Method.

In the context of network software, an application program is not an interactive job, but rather a terminal servicing facility that provides terminal users with a specific processing capability such as remote job entry from batch terminals, transaction processing, entry and execution of interactive jobs, and so forth. For example, the standard CDC Interactive Facility IAF makes terminal input and output appear the same to an executing program as file input and output; IAF is a network application program, but the executing program using IAF is an interactive job.

Archive Files

A dump of permanent files accumulated on disk that are dumped as a whole or in part to a backup tape (or other type of backup medium) to protect the files from loss in case of a device malfunction or to free a device for temporary use during preventive maintenance.

ASCII

American National Standard Code for Information Interchange. The standard character set and code used for information interchange between systems. It is a 7-bit code representing a prescribed set of 128 characters.

AST

See Allocation Summary Table.

Attach

The process of making a direct access permanent file accessible to a job by specifying the proper permanent file identification and passwords.

AU

See Allocation Unit.

Auto Recall

The act of a program releasing control of the CPU until a requested function is complete. Refer to Recall.

Auxiliary Device

A disk device that is not part of a permanent file family. Auxiliary devices can contain direct or indirect access permanent files.

B

Backup Tape

See Archive Files.

Batch Job

The instructions and data that are submitted as a complete unit without further user intervention. The job can be punched on cards or created and submitted from a terminal.

Beginning-of-Information (BOI)

The start of the first programmer record in a file is known as the beginning-of-information. System information, such as tape labels on sequential files or indexes, does not affect the beginning-of-information.

Binary File

A noneditable file that contains a precompiled program.

BOI

See Beginning-of-Information.

Breakpoint

A designated location in a program where, if reached during program execution, a break or suspension in execution occurs.

Buffer

An intermediate storage area used to compensate for a difference in rates of data flow, or times of event occurrences, when transmitting data between central memory and an external device during input/output operations.

Byte

A group of bits. Unless prefixed (for example, a 6-bit byte), the term means 8-bit groups. When used for encoding character data, a byte represents a single character.

C

Cache

A high-speed memory that resides in the central processor.

Cartridge

A component of the MSE. The cartridge consists of a plastic housing that encloses a strip of magnetic tape on which data is stored under program control.

Cartridge Alternate Storage

Cartridge (MSE) used as an alternate storage medium for permanent files. See Alternate Storage.

Cassette

The magnetic tape device in an NPU used for bootstrap loading of offline diagnostics and (in remote NPUs) the bootstrap load/dump operation.

Catalog Image Record (CIR)

A record written at the beginning of the archive file on which the permanent files are dumped for each incremental dump. When a file is loaded, this CIR information is placed in the permanent file catalog of the device being loaded.

Catalog Track

A track on a user's master device containing the catalog entries that define and specify the location of each permanent file created by that user. Users are assigned by groups to catalog tracks according to user index and number of catalog tracks on the master device.

Catenet

A collection of network solutions connected using gateways. Abbreviated from concatenated network.

CCITT

See Consultative Committee of International Telephone and Telegraph (CCITT).

CCP

See Communications Control Program.

CDCNET

See CONTROL DATA® Distributed Communications Network.

CDCNET Operator (COP)

The administrative operator who resides at either the system console or a terminal and controls CDCNET network elements (such as lines, terminals, trunks, and DIs associated with CDCNET).

Central Memory Resident (CMR)

The low address area of central memory reserved for tables, pointers, and subroutines necessary for operation of the operating system. It is never accessible to a user's central processor program. The monitor allocates the remainder of central memory to jobs as they are selected on a priority basis for execution.

Channel Number

The number of the data channel on which a peripheral device controller can be accessed.

Character

Unless otherwise specified, references to characters in this manual are to 7-bit ASCII code.

Charge Number

An alphanumeric identifier the installation uses to allocate charges to individual users for system usage.

Checkpoint

The process of writing a copy of your job's central memory, the system information used for job control, and the names and contents of all assigned files that are identified in a CHECKPT request to a magnetic tape or disk.

CIO

See Combined Input/Output

CIR

See Catalog Image Record.

CLA

See Communications Line Adapter.

CMR

See Central Memory Resident.

CMRDECK

The central memory resident deck that resides on the deadstart file. It defines central memory, table sizes, and configuration information not oriented to equipment.

Coldstart

Procedure used to deadstart if the tape or disk controller has not yet been loaded with controlware, or the controlware is not running.

Combined Input/Output (CIO)

A system routine that performs NOS I/O.

Common Testing and Initialization (CTI)

Common deadstart process that resides on the deadstart file and the maintenance system library.

Communication Line

A complete communication circuit between a terminal and its network processing unit.

Communications Control Program (CCP)

A portion of the network software that resides in a 255x series network processing unit. This software can include routines such as the terminal interface program.

Communications Line Adapter (CLA)

Hardware that provides the interface between NPUs and modems.

Communications Supervisor (CS)

A portion of the network software written as an application program; the CS coordinates the network-oriented activities of the host computer and of the lines and terminals logically linked to it.

COMPASS

COMPrehensive ASsembly System. The standard assembly language used with CYBER 180, CYBER 170, CYBER 70, and 6000 Computer Systems. Also, the command used to assemble a program written in the COMPASS assembly language.

Connection Number

A number assigned to an IAF terminal by the system when the terminal is logged in and an entry is made for the job in the executing job table.

Connection Status

A job attribute kept in the job's executing job table (EJT) entry. The system uses it to determine the job's relationship with IAF.

Consultative Committee of International Telephone and Telegraph (CCITT)

(Actually, the Comite Consultif International Telephonique et Telegraphique). An organization chartered by the United Nations to develop and publish international standards for the communications industry.

CONTROL DATA® Distributed Communications Network (CDCNET)

A catenet system using Control Data Network Architecture, CDC hardware, and CDC software.

Control Point

The portion of central memory that is assigned to a job. When a job is allocated a portion of central memory, it becomes eligible for assignment to the central processor for execution.

Control Point Number

The number of the control point to which a job is assigned while the job resides in central memory. The actual number of control points is an installation parameter. Before the job can execute, each central processor program must be assigned to a control point.

Controller

Hardware device that connects channels to peripheral devices. For example, a tape controller might connect up to eight tape units to one channel.

Controlling NOP

An NPU operator (NOP) who is allowed to change the status of network elements (lines, logical links, terminals, and trunks) connected to an NPU.

Controlware

A special type of software that resides in a peripheral controller. The controlware defines the functional characteristics of the controller.

COP

See CDCNET Operator.

CS

See Communications Supervisor.

CTI

See Common Testing and Initialization.

D

DAT

See Device Access Table.

Data Channel

One of the 9 to 24 channels (12-bit) by which information passes between the peripheral processors and peripheral devices. Refer to Channel Number.

Data Recording Drive (DRD)

A component of the 7991 Storage Module. The DRD reads data from and writes data to the cartridges.

Dayfile

A chronological file created during job execution that forms a permanent accounting and job history record. Dayfile messages are generated by operator action or when some commands are processed. A copy of the dayfile is printed with the output for each job. The user must explicitly request it in an interactive job.

DDP

The distributive data path.

Deadstart

The process of initializing the system by loading the operating system library programs and any of the product set from magnetic tape or disk. Deadstart recovery is reinitialization after system failure.

Deadstart Sequencing

The execution of a selected set of commands before normal system job scheduling is enabled.

Debug Log File Processor (DLFP)

A processor that analyzes trace files produced by the application interface program trace utility.

Destage Dump Tapes

Tapes that contain files that were destaged from disk using the PFDUMP utility with the DT parameter specified.

Destaging

The process of creating an alternate storage copy of a file on tape alternate storage or MSE. Files are destaged to tape alternate storage when the site executes PFDUMP with the DT parameter specified. Files are destaged to MSE when the site executes SSMOVE with the appropriate parameters specified.

Detached Job

An interactive service class job removed from control of the Interactive Subsystem. It may or may not continue to execute, depending on the presence of commands in the command buffer or an active job step. Control is regained by recovering the EJT entry for the job.

Device Access Table (DAT)

A table that contains the logical description (family name/pack name and device number) of each disk device (shared or nonshared) that is accessible by any machine in a linked shared device multimainframe complex.

Device Index Table (DIT)

A table that is used to determine device usage in an independent shared device multimainframe complex.

Device Interface (DI)

CDCNET hardware for open system interconnection. The device interface houses processor boards in configurations that permit a network of various other data processing equipment.

Device Mask

An 8-bit quantity that identifies the group of users who have the particular device as their master device; that is, it identifies the device that contains their file catalogs, all their indirect access files, and possibly some or all of their direct access files.

DFD Utility

A dayfile dumping utility that dumps all or selected parts of the system dayfile to produce a listing.

DFLIST Utility

A utility that generates a printer listing of all permanent files created by the DFTERM utility.

DFTERM Utility

A utility that terminates an active or inactive dayfile and retains it as a direct access permanent file for later use.

DI

See Device Interface.

Diagnostic Operator (DOP)

An NPU operator who resides at a terminal and has permission only to check status and test the network.

Direct Access File

A NOS permanent file that can be attached to the your job. All changes to this file are made on the file itself rather than a temporary copy of the file (compare with Indirect Access File).

DIS (Job Display)

A system peripheral processor program similar to the system display (DSD) that provides communication between a job in central memory and the operator at the console, and permits the operator to control execution of the program through the console keyboard.

Disabled Job

An interactive service class job temporarily rolled out due to user break processing or encountering the end of its command stream. The job scheduler ignores disabled jobs.

Display Code

A 6-bit character code set that represents alphanumeric and special characters.

Disposition Code

A 2-character mnemonic indicating the destination queue and format for processing a file named on a ROUTE function.

DIT

See Device Index Table.

DLFP

See Debug Log File Processor.

DMPNAD

See Dump NAD Memory.

DOP

See Diagnostic Operator.

Downline

The direction of output flow, from host to terminal.

DRD

See Data Recording Drive.

DSD (System Display)

The operating system program that provides communication between the operator and the system by accepting control information typed on the console keyboard and by displaying information pertinent to all jobs known to the system. DSD is permanently assigned to peripheral processor 1.

DSDI Utility

A deadstart dump interpreter utility that is called by a batch command to convert selected portions of the binary information on an express deadstart dump file into reports to be listed.

Dump NAD Memory (DMPNAD)

A utility that reads the NAD memory and formats the data into an output file.

E

ECS

See Extended Core Storage.

EDD File

See Express Deadstart Dump File.

EDD Utility

See Express Deadstart Dump Utility.

EJT

See Executing Job Table.

EJT Ordinal

An index into the executing job table (EJT). It uniquely identifies an EJT entry. The acceptable range is from 0001 to 4095.

ELD Utility

A dayfile dumping utility that dumps all or selected parts of the error log dayfile to produce a listing.

End-of-Chain Flag (EOC)

In an MSE environment, EOC is a flag in the 7990 catalog that identifies the last volume in a chain of allocated AUs.

End-of-File (EOF)

A boundary within a sequential file, but not necessarily the end of a file that can be referenced by name. The actual end of a named file is defined by EOI. For labeled tape, EOF and EOI (denoted by the EOF1 label) are the same. For multifile tape files, EOF and EOI do not correspond. In the product set manuals, an end-of-file is also referred to as an end-of-partition.

End-of-Information (EOI)

The end of data on a file. Information appearing after this point is not considered part of file data. In card decks, a card with a 6/7/8/9 multiple punch in column 1. On mass storage devices, the position of the last written data. On labeled tape, it is the EOF1 label. CYBER Record Manager defines end-of-information in terms of file residency and organization.

End-of-Record (EOR)

An indicator that marks the end of a logical record. Also referred to as end-of-section.

Entry Point

A location within a program or procedure that can be referenced from other programs. Each entry point has a unique name with which it is associated.

EOC

See End-of-Chain Flag.

EOF

See End-of-File.

EOI

See End-of-Information.

EOR

See End-of-Record.

EQPDECK

The equipment description deck used during deadstart to define the system equipment configuration.

Equipment Access Levels

A range of access levels specified for each equipment on a secured system. In order for a file to be stored or output on a given equipment, the file's access level must be within the equipment access levels for that equipment.

Equipment Number

A number from 0 to 7 that identifies the setting on a peripheral device controller.

Equipment Status Table (EST)

A central memory resident table listing all the defined equipments, parameters affecting their operation, and the status of the equipments.

ESM

See Extended Semiconductor Memory.

EST

See Equipment Status Table.

EST Ordinal

The number designating the position of an entry within the equipment status table (EST) established at each installation. Devices are identified in operator commands by EST ordinals. The EST ordinal is sometimes referred to as equipment number.

Ethernet

A baseband local area network protocol developed by the XEROX, Intel, and Digital Equipment Corporations. CDCNET is an Ethernet-compatible network.

Exchange Package

A table that contains information used during job execution. It is printed as part of the output when a job aborts.

Executing Job Table (EJT)

A central memory resident table that contains a 4-word entry for all executing jobs including interactive service class jobs. It is used to control jobs that are executing at a control point and jobs that are rolled out. Every executing job in the system has an EJT entry.

Execution

An input job is in execution after it is selected by the operating system and assigned to a control point. A job remains in execution until terminated, but it can be temporarily swapped or rolled out by the operating system.

Express Deadstart Dump (EDD) File

A file that is generated on magnetic tape by the express deadstart dump utility. This file contains a dump of memory, executing exchange packages, hardware registers, and controller memory.

Express Deadstart Dump (EDD) Utility

A utility that may be run at deadstart time after a system malfunction has occurred. It generates the express deadstart dump file on magnetic tape.

Extended Core Storage (ECS)

A type of extended memory that is an option available for 6000 Computer Systems, CYBER 70 Computer Systems, CYBER 170 Computer Systems (except model 176), and CYBER 180 Computer Systems. The maximum size of ECS is two million words. See Extended Memory.

Extended Memory (EM)

An additional portion of memory that is available as an option. This memory can be used for program and data storage, but not for program execution. Special hardware instructions exist for transferring data between central memory and extended memory. Extended memory consists of either extended core storage (ECS), extended semiconductor memory (ESM), large central memory extended (LCME), or unified extended memory (UEM).

Extended Memory File Space

The portion of extended memory that is defined as a mass storage device.

Extended Memory I/O Buffers

The portion of extended memory that is used for buffering I/O data to and from disks. Disks requiring I/O buffers are: 819, 885-42, 887, 895, and 9853.

Extended Semiconductor Memory (ESM)

A type of extended memory that is an option available for 6000 Computer Systems, CYBER 70 Computer Systems, CYBER 170 Computer Systems (except model 176), and CYBER 180 Computer Systems. The maximum size of ESM is 16 million words. See Extended Memory.

F

Facility Interface Program (FIP)

A program consisting of routines and buffers that are loaded into each application program's field length. This program is the interface between the application program and RHF.

Family Device

A mass storage permanent file device associated with a specific system. A family may consist of from 1 to 63 logical devices. Normally, a system runs with one family of permanent file devices available. However, additional families may be introduced during normal operation. This enables users associated with the additional families to access their permanent files via the alternate family.

Family Name

Name of the permanent file storage device or set of devices on which all of a user's permanent files are stored. When a user requests a permanent file, the system looks for it on this family (group) of devices. Usually a system has only one family of permanent file devices, but it is possible to have alternate families in the system. At login, the user may have to specify which family he/she is using. A user gets a family name from his/her employer, instructor, or computer center personnel.

Family Ordinal Table (FOT)

A table that maintains the relationship between family ordinals and family names.

FCT

See File and Cartridge Table.

FET

See File Environment Table.

Field Length

The area in central memory allocated to a particular job; the only part of central memory that a job can directly access. Also the number of central memory words required to process a job.

File

A collection of information referred to by a file name (from 1 to 7 alphanumeric characters). You can create a file at the terminal or retrieve a file from permanent file storage for use during a terminal session.

File Access Category

A property of a permanent file used by the creator of the file on a secured system to restrict access of the file to a particular group of users. A secured system supports up to 32 access categories, and a user is authorized to use some, all, or none of those categories. Refer also to System Access Categories.

File Access Level

A property of each file on a secured system used to indicate the sensitivity of information contained on the file. A file is assigned the current job access level by default when it is created or stored; the file creator may specify any access level for that file that is within the set of access levels valid for the job, the system, the file creator, and (for interactive jobs) the communication line to the host mainframe. If a user accesses a file on a secured system, that user must be validated for the access level of the file. Refer also to Access Level, Job Access Level, and Job Access Level Limits.

File and Cartridge Table (FCT)

Table that has an entry for each cubicle assigned to the subfamily from a given SM.

File Category

Each permanent file is assigned a category of private, semiprivate, or public.

File Count

A maximum number of permanent files allowed each user.

File Environment Table (FET)

A table within a program's field length through which the program communicates with operating system input/output routines. One FET exists for each file in use by the program.

File Name Table (FNT)

A system-managed table that contains the local file name, the file type, and other job control information. All active files in the system have an FNT entry.

File Status Table (FST)

A system-managed table that contains information pertaining to the file's location in mass storage and other job control information. Each active file in the system has an FST entry. See also File Name Table.

FIP

See Facility Interface Program.

First Level Peripheral Processor (FLPP)

The processor that is connected directly to the CYBER 170 Model 176 mainframe and operates synchronously with the mainframe.

FLPP

See First Level Peripheral Processor.

FNT

See File Name Table.

Forms Code

An attribute of output files and output devices. The user can specify special forms required for output, and then mount the special forms and use the FORM command to let the system process his/her output.

FOT

See Family Ordinal Table.

FST

See File Status Table.

Function Processor

A system CPU or PP program that the user can call by placing a request in location RA+1. Function processors perform input/output, local and permanent file manipulations, and so on.

H

Hardware Initialization and Verification Software (HIVS)

The software package that assists CTI during deadstart and provides deadstart confidence-level testing (HVS).

Hardware Verification Sequence (HVS)

HVS is a member of HIVS. It tests the ability of memory to hold patterns of data and execute instructions. The user can choose to test central memory, extended memory, PP memory, and central processor memory.

Head-of-Chain Flag (HOC)

In an MSE environment, HOC is a flag in the 7990 catalog that identifies the first volume in a chain of allocated AUs.

Header

A word or set of words at the beginning of a block, record, file, or buffer that contains control information for that unit of data.

HIVS

See Hardware Initialization and Verification Software.

HOC

See Head-of-Chain Flag.

HOP

See Host Operator.

Host

The computer that controls a network, executes the application programs, and processes network messages.

Host Operator (HOP)

The administrative operator who resides at the system console, initiates NAM, and controls NPUs and network related host elements.

HVS

See Hardware Verification Sequence.

I

IAF

See Interactive Facility.

Inactive Queued File Table (IQFT)

A table of file entries that has been removed from the queued file table. An IQFT file is on each mass storage device on which one or more inactive queued files reside.

Incremental Dump

An incremental dump copies those permanent files modified after a specified date. Each incremental dump writes a catalog image record at the beginning of the archive file on which the permanent files are dumped.

Incremental Load

An incremental load builds up an accumulation of the most recently modified versions of the files extracted from the archive files for loading. A series of archive files is read in reverse order of creation. The CIR is read and checked against the archive files. If a file matches an entry on the CIR, that file is a candidate for loading.

Indirect Access File

A NOS permanent file that you access by making a temporary copy of the file (GET or OLD command). You create or alter it by saving or substituting the contents of an existing temporary file (REPLACE or SAVE command).

Input File

The system-defined file that contains the entire job the user submits for processing. It is also known as the job file.

Input/Output Unit (IOU)

A collection of all PPs, PP channels, and related hardware for models 865 and 875 and CYBER 180-class machines.

Interactive Facility (IAF)

An application that provides a terminal operator with interactive processing capability. The Interactive Facility makes terminal input/output and file input/output appear the same to an executing program.

Interactive Transfer Facility (ITF)

A network application that allows the user to connect an interactive terminal to a remote CYBER 200 computer system linked to a host mainframe by a loosely coupled network.

IOU

See Input/Output Unit.

IQFT

See Inactive Queued File Table.

ITF

See Interactive Transfer Facility.

J

Job Access Level

On a secured system, each job has an access level. This is the default access level that is assigned to files that are created or stored in the job. A job's initial access level is the lower access level limit for the job. The job's access level is automatically raised to the access level of any file from which information is read. The user can also change the job access level. Refer also to Job Access Level Limits.

Job Access Level Limits

An upper limit and a lower limit that determine the range of access levels that are valid for a particular job on a secured system. All files used in a given job must have an access level within the job's access level limits.

Job Sequence Name (JSN)

The unique, system-defined name assigned to every executing job or queued file. The JSN is a string of four alphabetic characters.

Job Status

A job attribute kept in the job's executing job table (EJT) entry. It is used by the system to determine if a job is rolled in or rolled out. If the job is rolled out, job status indicates why it was rolled out.

JSN

See Job Sequence Name.

L

LAN

See Local Area Network.

Large Central Memory Extended (LCME)

A type of extended memory that is an option available for model 176. Refer to Extended Memory.

LCF

See Local Configuration File.

LCME

See Large Central Memory Extended.

LCN

See Loosely Coupled Network.

LDLIST Utility

A utility that generates a printer listing of queued files present on a dump tape produced by the QDUMP utility.

LFG

See Load File Generator.

LID

See Logical Identifier.

LISTPPM Utility

A PIP dump analyzer program that converts all available PIP dump binary records on th PIP memory dump file into a report to be listed in byte format.

Load File Generator (LFG)

A utility program that reformats communications control program files for subsequent use by the network supervisor of NAM to load network processing units.

Load Point

Metallic strip marking the beginning of the recordable portion of a magnetic tape. Data, including labels, is written after the load point. A rewind positions a single file volume to the load point.

Load Sequence

A sequence of load operations that encompasses all of the loader's processing from the time that nothing is loaded until the time execution begins. It includes initialization, specification of specified loader requests, and completion of load.

Local Area Network (LAN)

A privately owned network that interconnects data processing equipment to provide high-speed communications. It allows users and services to exchange messages and share resources.

Local Batch Job

A batch job submitted at the central computer site through a card reader or terminal.

Local Configuration File (LCF)

A file in the host computer system containing information on the logical makeup of the communication elements of the host. The file contains a list of the application programs available for execution in the host computer, and the users that can access it. This is a NOS direct access permanent file.

Local File

Any file that is currently associated with a job. Local files include all temporary files and attached direct access files.

Local File Name

The file name assigned to a file while it is local (assigned) to a job. The name is contained in the local file name table.

Local NAD

A 380-170 NAD connected to the host mainframe using a channel and configured in the EST.

Local NPU

An NPU that is connected to the host via a coupler. A local NPU always contains a host interface program for processing block protocol transfers across the host/local NPU interface.

Logical Identifier (LID)

A 3-character alphanumeric string used to identify a particular mainframe. LIDs are identified by the user's site.

Loosely Coupled Network (LCN)

A network of physically connected computer systems. The LCN environment allows jobs, data files, and messages to be transmitted from one computer system to another.

M

Machine Identification (MID)

The identifier that associates a specific machine with its access to a shared device.

Machine Recovery Table (MRT)

A table that provides the information needed to recover the mass storage space and interlocks of a machine that shares a mass storage device.

Machine Recovery Utility (MREC)

A utility that clears interlocks held by the machine to be deadstarted that have not been cleared by CPUMTR. It also recovers mass storage space on a shared device that is currently not accessible because of a machine interruption.

Macro

A sequence of source statements that is saved and then assembled whenever needed through a macro call.

Mainframe Device Interface (MDI)

A device interface that is configured to connect a CYBER mainframe to Ethernet.

Mainframe to Terminal Interface (MTI)

A terminal interface that is configured to connect a CYBER mainframe to a terminal for support of terminal-to-network communications.

MAINLOG Utility

A dayfile dumping utility that dumps all or selected parts of the binary maintenance log to produce an output file in binary format.

Maintenance Logging Transfer Utility (MLTF)

A utility that controls logging NAD error logs into the binary maintenance log.

Maintenance Register

A hardware register used in error detection, logging, and recovery procedures. Maintenance registers are used on models 865 and 875 instead of status/control registers. Refer to Status/Control (S/C) Register.

MAP

The Matrix Algorithm Processor.

Mass Storage Device

An extended memory or disk unit that has defined logical attributes such as family, file residency, and so on.

Mass Storage Extended Subsystem (MSE)

MSE is the product consisting of the 7990 hardware, the channel interface, the diagnostics, and the operational software. MSE stores data on the 7990 and moves it to disk upon request for access by an authorized user.

Mass Storage Table (MST)

A table that contains an entry for each logical device in the configuration of mass storage devices currently available to the system.

Master Device

A disk device that contains the user's permanent file catalog entries; all of the user's indirect access files; and all, part, or none of the user's direct access files.

MCT

See Memory Control Table.

MDI

See Mainframe Device Interface.

MID

See Machine Identification.

Memory Control Table

A central memory table used in allocating central memory and extended memory to user jobs.

MLIA

See Multiplex Loop Interface Adapter.

Monitor

The system routine that coordinates and controls all activities of the computer system. It occupies peripheral processor 0 and part of central memory. It schedules the use of the central processor and the other peripheral processors.

MREC

See Machine Recovery Utility.

MRT

See Machine Recovery Table.

MSE

See Mass Storage Extended Subsystem.

MST

See Mass Storage Table.

MTI

See Mainframe to Terminal Interface.

Multimainframe Operation

An operation that provides mechanisms by which more than one computer can share mass storage devices.

Multiplex Loop Interface Adapter (MLIA)

The hardware portion of the multiplex subsystem that controls the multiplex loops (input and output) as well as the interface between the NPU and the multiplex subsystem.

Multispindle Device

A logical mass storage device that includes from two to eight disk units.

Multiterminal Job

A job that does one specific task for many terminals while being scheduled into the system only once.

N

NAD

See Network Access Device.

NAM

See Network Access Method.

NCF

See Network Configuration File.

NCTF

See Network Description File.

NDA

See NPU Dump Analyzer.

NDI

See Network Device Interface.

NDL Processor

See Network Definition Language Processor.

NDR

See Network Driver.

Negative Field Length (NFL)

Central memory assigned to a control point that physically precedes the job's reference address (RA).

NETLOG

A program that uses the network configuration file to determine which remote NADs should be logged.

NETOU

See Network Operator Utility.

Network

An interconnected set of network elements consisting of a host and one or more NPUs and terminals.

Network Access Device (NAD)

The primary element in a loosely coupled network. Each NAD connects a computer system to the network.

Network Access Method (NAM)

A software package that provides a generalized method of using a communications network for switching, buffering, queuing, and transmitting data. NAM is a set of interface routines used by a terminal servicing facility for shared access to a network of terminals and other applications, so that the facility program does not need to support the physical structures and protocols of a private communication network.

Network Configuration File (NCF)

A network definition file in the host computer containing information on the network elements and permissible linkages between them. The status of the elements described in this file is modified by the NPU operator in the course of managing the network. This is a NOS direct access permanent file.

Network Definition Language (NDL) Processor

The network software module that processes an NDL program as an offline batch job to create the network definition files and other NDL program output.

Network Description File (NCTF)

A file that must be present if the Transaction Facility is used. The file is prepared by the site analyst.

Network Device Interface (NDI)

A device interface that is configured to transfer data between networks; for example, LAN to LAN, LAN to PDN, PDN to PDN, or LAN to communication lines.

Network Driver (NDR)

A program that executes in a dedicated peripheral processor unit. It communicates with the network access devices using a host computer data channel, and is the interface between RHF and the communication network.

Network Invocation Number (NIN)

A 1- to 3-digit decimal number. NIN is incremented by 1 every time NAM is brought up.

Network Load File (NLF)

An output file generated by the load file generator utility for use by the network supervisor.

Network Operator Utility (NETOU)

A group of programs residing in a host computer and in a mainframe device interface connected to the mainframe that allow a network operator to access, monitor, control, and configure a CDCNET network from the host console or a remote terminal. NETOU allows commands from network operators to be sent through the CDCNET network to specific device interfaces or all of the DIs in the network.

Network Processing Unit (NPU)

The collection of hardware and software that switches, buffers, and transmits data between terminals and host computers.

Network Supervisor (NS)

A portion of the network software written as a NAM application program. NS dumps and loads NPUs upon request.

Network Terminal

A terminal that communicates with the operating system through the network.

Network Validation Facility (NVF)

A portion of the network software, written as a NAM application program. The network validation facility performs application validation and all connection validation processing and supports login dialog with the terminal user.

NFL

See Negative Field Length.

NIN

See Network Invocation Number.

NLF

See Network Load File.

Nonincremental Load

A nonincremental load does no CIR checking and uses only parameter options specified on the PFLOAD call, if any, to select candidates for loading.

NOP

See NPU Operator.

NPU

See Network Processing Unit.

NPU Dump Analyzer (NDA)

A utility program that produces a readable printout from the NPU dump files.

NPU Operator (NOP)

The administrative operator who resides at a terminal and controls NPUs.

NS

See Network Supervisor.

NVF

See Network Validation Facility.

O

Object Code

The machine language version of a program that has been translated (compiled) from source code written in a higher-level language.

Operating System

The set of system programs that controls the execution of computer programs and provides scheduling, error detection, input/output control, accounting, compilation, storage assignment, and other related services.

Origin Type

A job attribute that indicates how a job entered the system. The four origin types are interactive origin, batch origin, remote batch origin, and system origin.

Output File

The system-defined file that contains the output from job processing. It is also known as the print or punch file.

P

PACKER Utility

A utility that provides the capability to manage the holes within the indirect access permanent file chain on a permanent file device.

Paging (Screen)

The process of filling a CRT display with data and holding additional data for subsequent displays. Changing the page display is an operator-controlled function if the page-wait option is selected.

Parity

In writing data, an extra bit is either set or cleared in each byte so that every byte has either an odd number of set bits (odd parity) or an even number of set bits (even parity). Parity is checked on a read for error detection and possible recovery.

Partial Dump

A partial dump copies permanent files according to any specified options, except those defining a full or incremental dump.

Password

A name or word the user enters during login to provide extra security for his/her user name. A unique password ensures that no one else can log into the system with someone else's user name and access that user's files. A user's password is given to him/her by that user's employer, instructor, or computer center personnel.

PCP

See Pseudo-control Point.

PDN

See Public Data Network.

Peripheral Interface Package (PIP)

The interface package between the PPU of the CYBER computer and the network application.

Peripheral Processor (PP)

The hardware unit within the host computer that performs physical input and output through the computer's data channels.

Peripheral Processor Unit (PPU)

First level peripheral processor. A PPU is contained in the mainframe in a multimainframe environment and operates synchronously with the mainframe. Sometimes referred to as FLPP.

Permanent File

A mass storage file that is cataloged by the system so that its location and identification are always known to the system. Permanent files cannot be destroyed accidentally during normal system operation. They are protected by the system from unauthorized access according to privacy controls specified when they are created.

Permanent File Catalog Entry (PFC)

A 16-word entry that the system maintains and uses to determine the file name, owner, identification, disk pointers, alternate storage pointers, and other attributes of a permanent file.

Permanent File Family

The permanent files that reside on the family devices of a specific system.

Permanent File Manager (PFM)

PFM identifies the master device and catalog track information when a user submits a job.

Permanent File Supervisor (PFS)

The PFS processes parameters in utility commands and loads the correct processing overlays.

Permanent File Transfer Facility (PTF)

PTF is an application program initiator started by the user using an MFLINK command. It is responsible for initiating and completing (with the help of its servicing application, PTFS on another host) a permanent file transfer.

Permanent File Transfer Facility Servicer (PTFS)

PTFS is an application program servicer started by RHF or NAM when requested by a PTF on another host. The PTFS application assists the PTF application in completing the file transfer by performing those permanent file functions requested by the user and then transferring the file between PTF and PTFS.

Permanent File Utility (PFU)

A utility that manages the catalogs, permits, data allocation on a device, and the data transfer between the device and the overlay.

Permission Mode

A mode of operation that a user is allowed for a particular permanent file, such as write, modify, append, read, and so forth.

PFATC Utility

A utility that produces a cataloged directory of file information derived from an archive file previously created by the PFDUMP utility.

PFC

See Permanent File Catalog.

PFCAT Utility

A utility that produces a cataloged directory of file information derived from catalog tracks on a master device.

PFCOPY Utility

A utility that extracts files from an archive file and copies them to one or more files at a control point.

PFDUMP Utility

A utility that dumps permanent files to an archive file. Dumps can be reloaded by the PFLOAD utility and can be accessed by the PFATC and PFCOPY utilities for cataloging and copying.

PFLOAD Utility

A utility that loads archived files produced by the PFDUMP utility back into the permanent file system. The load can reestablish the permanent file system exactly as it was at the time of the dump, or can load only a desired subset of files on the archive file.

PFM

See Permanent File Manager.

PFREL Utility

A utility that releases disk space for permanent files that have copies on alternate storage.

PFS

See Permanent File Supervisor.

PFU

See Permanent File Utility.

Physical Identifier (PID)

The unique 3-character identifier of a specific host.

Physical Record Unit (PRU)

The amount of information transmitted by a single physical operation of a specified device. For mass storage files, a PRU is 64 central memory words (640 characters); for magnetic tape files, the size of the PRU depends upon the tape format. A PRU that is not full of user data is called a short PRU; a PRU that has a level terminator but no user data is called a zero-length PRU.

PICB

See Program Initiation Control Block.

PID

See Physical Identifier.

PIP

See Peripheral Interface Package.

PP

See Peripheral Processor.

PPS

The Peripheral Processor Subsystem.

PPU

See Peripheral Processor Unit.

Preserved File

A mass storage file that is recovered on all levels of system deadstart. Preserved files include permanent files, queued files, and system dayfiles.

Primary File

A temporary file created with the OLD, NEW, LIB, (interactive jobs only), or PRIMARY command. The primary file is assumed to be the file on which most system operations are performed unless another file is specified. There can be only one primary file associated with your job.

Primary VSN

The volume serial number (VSN) of a single reel destage dump tape or the VSN of the first reel of a multireel set of destage dump tapes. The VSN must be two alphanumeric characters followed by a four-digit decimal number; the number must be in the range 0000 through 4095.

Privileged Analyst

A user with AW=CPAM validation (refer to the NOS Version 2 Administration Handbook for information on MODVAL validation). Such a user can read system status information (such as the system dayfile, account file, and error log) using a nonsystem-origin job if PRIVILEGED ANALYST MODE is enabled (via IPRDECK or DSD command). Note that PRIVILEGED ANALYST MODE cannot be enabled on a secured system.

PROBE Utility

A utility that traps and measures particular interval events in the system. PROBE generates a report from the data collected by the system.

Procedure

A user-defined set of instructions that can be referenced by name. The instructions consist of procedure directives and system commands.

Program Initiation Control Block (PICB)

A sequence of commands that initiates NPU load and dump operations for a specific NPU variant. Several PICBs may exist on the network load file. Each PICB is a separate record with a unique NPU variant name as its record name.

Programmable Format Control

Spacing and format control for 580 line printers provided by the use of software and a microprocessor instead of a carriage control format tape.

Project Number

An alphanumeric identifier that may be required at a user's installation for accounting and billing to a specific project. If it is required, the project number is entered during the login procedure. It is given to the user by personnel at his/her installation.

Protocol

A set of conventions or rules that must be used to achieve complete communication between entities in a network. A protocol can be a set of predefined coding sequences such as the control byte envelopes added to or removed from data exchanged with a terminal; a set of data addressing and division methods, such as the block mechanism used between an application program and NAM; or a set of procedures used to control communication, such as the supervisory message sequences used between an application program and NAM.

PRU

See Physical Record Unit.

Pseudo A Register

A software register used by DSD to function channels and to manipulate peripheral hardware devices from the operator's console.

Pseudo-control Point (PCP)

The portion of central memory that is assigned to a job that has been pseudo-rolled. Pseudo-control points are the same as control points except that a job at a pseudo-control point cannot have any activity (such as PP assignment or CPU assignment).

Pseudo-control Point Number

The number of the pseudo-control point to which a job is assigned while the job resides in central memory. The actual number of pseudo-control points is an installation parameter.

Pseudo-rollout

The removal of jobs from a control point to a pseudo-control point so that the control point and central memory can be assigned to another job. A job is rolled out to a pseudo-control point when its control point is needed by a higher priority job.

Pseudo-rollin

The return of jobs from a pseudo-control point to a control point.

PTF

See Permanent File Transfer Facility.

PTFS

See Permanent File Transfer Facility Servicer.

Public Auxiliary Device

An auxiliary device that is available for access by all validated users knowing the correct pack name. Additional validation is required to create or replace files on an auxiliary device.

Public Data Network (PDN)

A commercial packet-switching network that supports the interface described in the CCITT protocol X.25.

Q

QALTER Utility

A utility that displays, lists, and/or alters routing and other information about active queued files. It selects files for processing according to specified criteria. QALTER can also purge selected files from the system.

QDUMP Utility

A utility that dumps selected queued files from a single device, a family of devices, or all devices on the system. These queued files can be dumped either to a tape or disk. QDUMP also provides a listing of all files dumped with information about each file processed.

QFSP

See Queue File Supervisor Program.

QFT

See Queued File Table.

QFTLIST Utility

A utility that displays and/or lists routing and other information about active queued files. Its operation is similar to that of QALTER, except file alteration or purging is not allowed.

QLIST Utility

A utility that lists inactive queued files, which may include all inactive queued files in the system or a selected subset based on options specified when the utility is called.

QLOAD Utility

A utility that processes the dump files generated by QDUMP or other utilities using the same format. QLOAD can selectively load the queued files from these dump files. QLOAD can also list the contents of a dump file without loading any files.

QMOVE Utility

A utility that moves queued files from one disk device to another. It also produces a listing of all files moved with information about each file processed.

QREC Utility

A utility that deactivates or activates selected queued files and purges selected inactive queued files.

QTF

See Queue File Transfer Facility.

QTFS

See Queue File Transfer Facility Servicer.

Queue File Supervisor Program (QFSP)

A program that provides control for the queue file utilities.

Queue File Transfer Facility (QTF)

QTF is an application program initiator that periodically scans the I/O queues searching for files to transfer. When it finds a file to transfer, it initiates and completes the queue file transfer with the help of its servicing application, QTFS, on another host.

Queue File Transfer Facility Servicer (QTFS)

QTFS is an application program servicer started by RHF or NAM when requested by a QTF on another host. The QTFS application assists the QTF application in completing the transfer by receiving the queue file and placing it in the I/O queue.

Queue Priority

An attribute associated with input and output files. If all other factors are equal, queue priority is used to select the best file for processing.

Queued File

An input, print, plot, or punch file that has an entry in the QFT, is not assigned to an EJT entry, and is waiting to be selected for processing.

Queued File Table (QFT)

A central memory resident table that contains a 4-word entry for all active input and output queue files.

R

Random Access

An access method by which any record in a file can be accessed at any time. Random access applies only to mass storage files with an organization other than sequential. Refer to Sequential Access.

RCFGEN

See RHF Configuration File Generation.

RCL

See Resident Central Library.

RDF

See Release Data File.

Recall

The state of a program when it has released control of the central processor until a fixed time has elapsed (periodic recall) or until a requested function is completed (auto recall). Recall is a system action request as well as an optional parameter of some file action requests.

Record

A unit of information. In CYBER Record Manager and its language processors, a record is a unit of information produced by a single read or write request.

Eight different record types exist within CRM. The user defines the structure and characteristics of records within a file by declaring a record format.

Regulation Level

A number that indicates to NAM the existence of a logical link and indicates what types of information exchange are possible on that logical link.

Release Data File (RDF)

A file created by PFDUMP that identifies those 7990-resident files that are pointed to by PFC entries at the time of the dump.

Remote Batch Job

A job submitted from a remote batch terminal.

Remote Host Facility (RHF)

A central processor program that executes at a system control point. It performs data buffering and switching, and is the intermediary between application programs and the network.

Remote NAD

Any 380 NAD accessible to a local NAD using a loosely coupled network trunk.

Remote NPU

A network processing unit linked to a host computer through other network processing units.

Removable Device

A disk storage device that can be physically detached from the disk drive.

Resident Central Library (RCL)

An area in central memory resident that central library routines specified by the *CM directive reside.

Resident Peripheral Library (RPL)

An area in central memory resident that peripheral library routines specified by the *CM directive reside.

RHF

See Remote Host Facility.

RHF Configuration File Generation (RCFGEN)

A utility that reads configuration definition statements to create a permanent file that RHF uses for the network description and access.

Rollout

The removal of jobs from central memory to mass storage before execution is complete, so the control point and central memory can be assigned to another job. A job is rolled out when it is waiting for an external event, when its control point and/or central memory is needed by a higher priority job, or when it exceeds its central memory time slice.

Rollout File

A file containing a job (and system information) that has been temporarily removed from the main processing area of the system.

RPL

See Resident Peripheral Library.

S

SC

See Service Class.

S/C Register

See Status/Control Register.

Scheduling Priority

An attribute associated with an executing job available for job scheduling. Scheduling priority is used to select the best executing service class job for processing.

SCOPE 2 Station Facility (SSF)

A NOS subsystem that allows a NOS user to submit jobs (including batch jobs that require interactive I/O) to a linked SCOPE 2 system. The submitted job uses standard SCOPE 2 commands to access NOS files stored on the originating NOS system.

SCP

See Subcontrol Point.

Screen Management Facility (SMF)

A subsystem that alters the performance characteristics of the Full Screen Editor (FSE). The absence or presence of SMF is not detectable by the FSE user. Performance can be optimized by disabling SMF for small mainframes and interactive work loads, and by enabling SMF for large configurations and heavy work loads.

SCRSIM

See Status/Control Register Simulator.

SDF

See System Deadstart File.

SECDED

See Single Error Correction Double Error Detection.

Secondary Mask

An 8-bit quantity used to identify groups of users who can place direct access files on a particular device.

Secondary VSN

The VSN of a single reel destage dump tape or the VSN of the first reel of a multireel set of destage dump tapes having a sequence number (the last four characters of the VSN) in the range 5000 through 9095. The existence of a secondary VSN assumes the existence of a primary VSN having the same two-character prefix and a sequence number of 5000 less. The primary and secondary VSN tapes or multireel sets of tapes are assumed to contain identical data although the individual reels of multireel sets cannot be assumed identical.

Secured System

A system in which a mandatory security mechanism has been enabled during deadstart. A secured system protects information by enforcing restrictions based on access levels and access categories, and restricts many sensitive system functions to security administrators.

Security Administrator

A secured system prevents users and operators from performing certain functions that could result in the unauthorized disclosure or modification of information. These functions can only be performed by someone who is designated a security administrator. A security administrator is always authorized to access the highest level of information stored on the system. This person performs functions in the areas of installation, user validation, system operation, and system maintenance.

Security Unlock Status

This status of the system console applies only to a secured system and must be set by a security administrator. The console must be in security unlock status in order for the security administrator to perform certain functions that are restricted on a secured system.

Sequential Access

A method in which only the record located at the current file position can be accessed. Refer to Random Access.

Sequential (SQ) File

A file in which records are accessed in the order in which they occur. Any file can be accessed sequentially.

Service Class (SC)

An attribute associated with a queued file or executing job. The service class determines how the system services the job.

SFS

See Special File Supervisor.

Single Error Correction Double Error Detection (SECDED)

A hardware technique that detects and corrects single bit errors in memory. Double bit errors are detected by not corrected.

SM

See Storage Module.

SM Map

See Storage Module Map.

SMF

See Screen Management Facility.

Source Code

Code input to the computer for later translation into executable machine language instructions (object code).

Special File Supervisor (SFS)

A program that provides routines, table management, data manipulation, and I/O processing for special system jobs.

SQ File

See Sequential File.

SRU

See System Resource Unit.

SSALTER Utility

A utility that displays the current 7990 hardware configuration and allows an analyst to change the status of certain elements in the 7990 configuration.

SSBLD Utility

A utility that processes statements consisting of 7990 component-oriented mnemonics that define the logical mapping within the 7990 configuration. SSBLD generates a direct-access permanent file which SSEXEC uses as its unit device table.

SSDEBUG Utility

A utility that allows an analyst to update appropriate entries in the SM maps and/or 7990 catalogs and thereby resolve inconsistencies reported by the SSVAL utility. SSDEBUG can also be used to copy data from selected 7990 files or cartridges to disk.

SSDEF Utility

A utility that creates the system files (SM maps and 7990 catalogs) that are necessary for MSE processing.

SSEXEC Program

The main processing program that controls MSE activities, such as destaging files from disk to the 7990, purging unneeded 7990 files, labeling or relabeling cartridges, updating SM maps, and updating 7990 catalogs.

SSF

See SCOPE 2 Station Facility.

SSLABEL Utility

A utility that manages cartridge assignment and cubicle allocation in a storage module.

SSMOVE Utility

A utility that manages disk and 7990 residence. SSMOVE determines which files to leave on disk, which files to release from disk and move to 7990, and which files should reside both on disk and on 7990.

SSSLV Program

A program that runs on each slave mainframe and communicates with the SSEXEC program to retrieve files from the 7990 in response to ATTACH requests by jobs running on the slave mainframes.

SSUSE Utility

A utility that reads data in the 7990 catalogs and SM maps and produces reports on the availability of space on 7990 cartridges and the allocation of cubicle space within an SM.

SSVAL Utility

A utility that either performs release processing or reports on problems with the current MSE system files.

Staging

The process of restoring file data to disk residence from a copy on alternate storage. Staging is initiated when a user executes an APPEND, ATTACH, GET, or OLD command for a file that is not currently disk resident.

Status

Information relating to the current state of a device, line, and so forth. Service messages are the principal carriers of status information. Statistics are a special subclass of status.

Status/Control Register Simulator (SCRSIM)

A program that enables the user to set status/control register bits in order to aid in the testing of error logging and error recovery procedures.

Status/Control (S/C) Register

A hardware register used in error detection, logging, and recovery procedures. This register is present on all CYBER 170 Computer Systems. For models 865 and 875, the S/C register is replaced by a maintenance register. Refer to Maintenance Register.

Step Mode

A protected or debugging mode for the operating system monitor. The keyboard spacebar must be pressed to process each PP request.

Stimulator

A collection of central memory and peripheral processor programs that enters a hypothetical work load into the system to analyze the effects of such a load on response time and system reliability.

Storage Module (SM)

An MSE hardware unit that houses up to 312 usable data cartridges, a cartridge accessor unit that picks cartridges from and puts cartridges in their cubicles, and one or two data recording drives.

Storage Module (SM) Map

A direct access permanent file that contains information indicating the cartridges that reside in the SM.

Subcontrol Point (SCP)

A division of a central memory control point. You can set up a control point to contain two or more programs; one of the programs is the executive, and monitors the other programs executing at the subcontrol points.

Subfamily

Each permanent file family consists of eight subfamilies, subfamily 0 through subfamily 7. The lower 3 bits of the user index identify the subfamily to which a user belongs.

Suspended Job

An interactive job placed in a inactive state. Processing is stopped immediately and recovery information is copied to the rollout file. Processing is resumed as if no interruption took place, if the job's EJT entry is recovered.

System Access Categories

On a secured system, a set of access categories are set during level 0 deadstart. This set may consist of some, all, or none of the 32 possible access categories. While the system is running in security mode, you may only use access categories that are within the set of system access categories.

System Access Levels

On a secured system, a range of access levels is set during level 0 deadstart. This range may contain some or all of the eight possible access levels. While the system is running, users may only use access levels that are within the range of system access levels.

System Deadstart File (SDF)

A file that is a copy of the deadstart tape that resides on a disk storage deadstart device. When the system is deadstarted from disk, this file is read to generate copies of the running system.

System Library (SYSTEM)

The collection of tables and object language programs that reside in central memory or on mass storage and are necessary for running the operating system and its product set.

System Origin Job

A job entered at the system console.

System Resource Unit (SRU)

A unit of measurement of system usage. The number of SRUs includes the central processor time, memory usage, and input/ output resources used for a given job.

T

TAF

See Transaction Facility.

Tape Alternate Storage

Magnetic tape used as an alternate storage medium for permanent files. See Alternate Storage.

TCU

See Trunk Control Unit.

TDI

See Terminal Device Interface.

Temporary File

A file associated with a job that is not a permanent file. Temporary files no longer exist when the user logs off the system or releases the files.

Terminal Device Interface (TDI)

A device interface that is configured to support terminal-to-network communications.

Timed/Event Rollout

A condition in which an executing job has been temporarily removed from central memory but will be rolled back into central memory when a specified event (such as a file is no longer busy) or a specified time period has elapsed.

TRACER Utility

A utility that monitors the system's activity and gathers data periodically for statistical analysis of the system.

Track Link

An address of the next track that is a logical continuation of a file.

Track Reservation Table (TRT)

A table that describes the physical layout of data on a device and is the key to allocating information on the device.

Transaction Facility (TAF)

An application program that provides the transaction terminal with access to a data base. A terminal using TAF can enter, retrieve, and modify information in the data base.

TRT

See Track Reservation Table.

Trunk

The communication line connecting two network processing units.

Trunk Control Unit (TCU)

The hardware part of a network access device (NAD) that interfaces with a network trunk.

U

UDT

See Unit Device Table.

UEM

See Unified Extended Memory.

Unified Extended Memory (UEM)

A type of extended memory that is available as an option for CYBER 180-class machines and models 865 and 875. UEM differs from other types of extended memory in that it is a portion of central memory and not a separate memory unit. See Extended Memory.

Unit Device Table (UDT)

A table that defines the logical mapping of the components within a 7990 configuration.

Unit Number

The setting of a hardware device. The unit number is used when more than one hardware unit can be connected to a controller.

Unsecured System

A system in which the multilevel security mechanism has not been enabled during deadstart. The restrictions based on access levels and access categories are not enforced on an unsecured system.

Upline

The direction of input flow from terminal to host.

User Break 1 Sequence

The character or sequence of characters that causes an executing program to be interrupted (also called the interruption sequence).

User Break 2 Sequence

The character or sequence of characters that causes an executing program to be terminated (also called the termination sequence).

User Index

A unique 17-bit identifier that is associated with each user name. The user index is used by the permanent file manager to identify the device and catalog track for the user's permanent files.

User Job Name (UJN)

A 1- to 7-character alphanumeric name you specify to replace the system defined JSN for a queued file or executing job.

User Name

A name given to the user by his/her employer, instructor, or computer center personnel. A user name has certain resources and privileges assigned to it. When logging in to the system, a user specifies his/her user name as identification, so that it knows that this person is an authorized user and what resources this user is entitled to use. A user name also represents a specific catalog in the permanent file system. All files a user makes permanent are associated with that user's name and this catalog.

V

Validation File

A file that contains validation information for all users (user names, passwords, resources allowed, and so on).

Volume Serial Number (VSN)

A from 1- to 6-character identifier that identifies the volume of magnetic tape to the system.

VSN

See Volume Serial Number.

W

Word

A group of bits (or 6-bit characters) between boundaries imposed by the computer system. A word is 60 bits in length. The bits are numbered 59 through 0 starting from the left. A word is also composed of five 12-bit bytes, numbered 0 through 4 from the left.

Write Mode

A mode that allows a user to write, modify, append, read, execute, or purge the file (modify permission applies only to direct access files).

Z

Zero-Length PRU

A PRU that contains system information but no user data. Under NOS, a zero-length PRU defines EOF.

7990

A hardware product that is a large capacity mass storage device. The 7990 is a cost-effective extension to the disk file storage system and an alternative to conventional magnetic tape storage.

7990 Catalog

A disk-resident direct access permanent file that contains information describing which AUs of each cartridge assigned to a particular subfamily are allocated to 7990 files and which AUs are available for allocation.

Scope 2 Station Facility — C

The SCOPE 2 Station Facility (SSF) is a NOS software subsystem that allows a NOS user to submit jobs and transfer files to a linked SCOPE 2 system residing in a 7000 Computer Systems mainframe. With SSF, a NOS user can route jobs to a linked SCOPE 2 system for processing. While processing the job, the SCOPE 2 system can access permanent files and tape files at the NOS host.

SSF provides these services:

- Provides NOS users with local batch capabilities at a linked SCOPE 2 system.
- Optionally serves as the system operator station for SCOPE 2.
- Provides the NOS user with remote batch capabilities, through RBF, at a linked SCOPE 2 system.
- Allows simultaneous transfer of multiple files.
- Optionally furnishes the deadstart file for SCOPE 2.
- Provides a means of obtaining absolute SCOPE 2 dumps.
- Allows a job executing on a SCOPE 2 system to use SCOPE 2 commands (GETPF, SAVEPF, and PURGE) to access permanent files at a linked NOS system.
- Allows a job executing on a SCOPE 2 system to access magnetic tape files at a linked NOS system.

Equipment Configuration

To establish a NOS-SCOPE 2 link using the SSF Subsystem, the NOS and SCOPE 2 mainframes must be attached to a loosely coupled network (LCN) by a 6683 and 7683 Satellite Coupler, respectively. An equipment status table (EST) device type entry of CC describes the coupler link. The EST table appears on the E,A display.

SSF File Transfers

SSF file transfers between NOS and SCOPE 2 are of two types:

- Staged file transfers.
- Spooled file transfers.

Staged file transfers process one file at a time and handle all tape file and permanent file transfers. Spooled file transfers handle multiple I/O file transfers. Normally, operator intervention is not required for either type of transfer.

SSF initiates a special type of job, called a spunoff task (SPOT) job, to perform file transfers. Staged file SPOT jobs differ from spooled file SPOT jobs as described in this appendix. The primary functions of any SPOT job are performing user validation and queuing tasks required to send a file across the link. All SPOT jobs are executed on the NOS system.

Installation parameters or operator commands can be used to limit the number of file transfer activities that can be active concurrently. Independent limits can be set on the following activities:

- Purging NOS files.
- Reading tape files from NOS to SCOPE 2.
- Writing tape files from SCOPE 2 to NOS.
- Reading permanent files from NOS to SCOPE 2.
- Writing permanent files from SCOPE 2 to NOS.
- Reading input files from NOS terminals to SCOPE 2.
- Writing output files from SCOPE 2 to NOS terminals.

The operator commands used to set limits are described in section 8, K-Display Utilities.

Staged File Transfers

Staged file transfers are essentially the same for both tape and permanent file transfers. In response to an SSF query, SCOPE 2 indicates to SSF that an executing SCOPE 2 job requires access to a NOS file. The SCOPE 2 request can specify either a NOS to SCOPE 2 file transfer, also called a prestaging transfer; or a SCOPE 2 to NOS (poststaging) transfer. In either case, SSF normally responds by creating a SPOT job to perform the file transfer. The SPOT job name is of the form xyyy, where x is the last character of the PID of the SCOPE 2 mainframe, and yyy is the JCB ordinal of the SCOPE 2 job. SSF places the newly created SPOT job into the NOS input queue, and the job executes on NOS in the same manner as any other job.

During execution, the job checks that the user has the appropriate file access permission. If so, the SPOT job initiates the file transfer.

For staged file transfers, each SPOT job terminates upon completion, and the job's dayfile (containing all processing and error messages) is sent to SCOPE 2 to be included in the dayfile of the job that requested the file transfer. SSF initiates a new SPOT job for each staged file transfer request it receives.

SSF periodically queries SCOPE 2 about file staging activities that should be dropped. For example, the SCOPE 2 system operator may have dropped a job for which staging was active. If any file staging activities are to be dropped, SSF requests information to identify the staging activity and then drops the associated SPOT job. No operator intervention is required.

Spooled File Transfers

Spooled file transfers use a single SPOT job to perform all input/output file transfers between SSF and SCOPE 2. As soon as communications are established between SSF and SCOPE 2, SSF creates a SPOT job to handle input/output file transfers. The job is called xSTA, where x is the PID of the SCOPE 2 mainframe. This SPOT job can be swapped out during periods of low activity, but the job is not terminated until the SSF-SCOPE 2 link is dropped.

An input file transfer begins when the spooling SPOT job selects a file from the NOS input queue that is destined for the SCOPE 2 system. The SPOT job queries SCOPE 2 for a system file table (SFT) entry for the input file to be transferred. When an entry is assigned, the spooling SPOT job transfers the input file to SCOPE 2. Upon completion of the transfer, the SPOT job requests that the input file be removed from the NOS input queue.

For each output file transferred from SCOPE 2, the spooling SPOT job transfers the output file from the SCOPE 2 output queue to a local NOS file. When the transfer is complete, the SPOT job routes the local file to the NOS output queue (with the appropriate disposition code, forms code, or terminal ID). The spooling SPOT job then requests the removal of the output file from the SCOPE 2 output queue.

Error Logout

A loss of communications between SSF and SCOPE 2 results in an error logout condition. Error logout occurs as a result of a STOP command entered by the system operator, or a software or hardware error. When SSF detects an interruption in communications, it logs out of the SCOPE 2 system after:

- All partially transmitted files are rewound on the sending end of the link and are deleted at the receiving end.
- All SPOT jobs are dropped.

No files are lost as a result of the communications break. After logging out of the SCOPE 2 system, SSF periodically queries SCOPE 2 to determine if the link has been restored.

Status/Control Register Simulator D

Error logging on a CYBER 170 Computer System enables you to detect and log errors identified in the status/control (S/C) register. The status/control register simulator (SCRSIM) allows you to set S/C register bits in order to aid in the testing of error logging and error recovery procedures.

SCRSIM does not run on CYBER 180-class machines. Models 865 and 875 use maintenance registers instead of S/C registers. Throughout this appendix all references to S/C registers also apply to the model 865 and 875 maintenance registers.

SCRSIM runs on CYBER 170 Computer Systems using the S/C register on channel 16 and, if more than 10 PPs are available on the system, the S/C register on channel 36. On CYBER 70 Computer Systems, SCRSIM uses the interlock register on channel 15.

With the aid of a K display, you can specify commands to set and clear bits, set bytes, and set lines and areas in holding registers. This allows both S/C registers to be set up completely. The contents of the holding register can then be transferred to the S/C registers (64 or 128 bits are transferred to the interlock register of a CYBER 70 Computer System).

The bits set through this simulator are logged in the error log if an error bit is set, thus aiding in testing and software checkout. (Refer to the appropriate hardware reference manual for a complete description of the significance of each S/C register bit.) The simulator job dayfile lists all simulator commands entered. This error logging does not occur in a CYBER 70 Computer System, however, unless an ENABLE,SCRSIM IPRDECK entry has been made.

NOTE

Be careful when using the simulator. Improper use may result in serious system malfunctions.

Refer to the NOS Version 2 Operations Handbook for descriptions of messages produced by SCRSIM.

Using the Simulator

Error logging is always enabled on a CYBER 170 Computer System except models 815, 825, 835, 845, and 855. Error logging is enabled on a CYBER 70 Computer System only if the ENABLE,SCRSIM IPRDECK entry is present.

Console Operation

The simulator is called from the console by entering:

```
X.SCRSIM.
```

The simulator K display (refer to figure D-1) appears on the left screen after entering:

```
K,jsn.
```

Entry	Description
jsn	The job sequence name of SCRSIM noted on the B,O display.

This K display shows the contents of the temporary holding registers, as well as a central memory buffer. The buffer contains:

- A history of all error status bits since the last level 0 deadstart. If an error status bit has been set in the S/C register, it remains set in the buffer, even though it may have been cleared in the actual S/C register. This history may be useful in diagnosing system malfunctions.
- All other bits in the buffer reflect actual values in the S/C register at the time the last error bit was set. Each time an error bit is set, the entire buffer is updated.

Unless the simulator is running on a CYBER 170 Computer System with more than 10 PPs, the message

```
CHANNEL 36 NOT AVAILABLE
```

also appears. This indicates that no channel 36 S/C register is present on the machine, and thus, no simulation need be done for it.

By entering

```
KK.
```

the simulator commands K display (refer to figure D-2) appears on the right screen. This display gives a brief description of the commands available.

Commands can be entered on the K display by entering:

```
K.command.
```

Entry	Description
command	One of the commands shown in figure D-2.

```
                 TEMPORARY HOLDING REGISTER CONTENTS

         CHANNEL 16 REGISTER - LINES 0-3


BITS  59-  0                                              BYTE 00
000000000000 000000000000 000000000000 000000000000 000000000000
    0000         0000         0000         0000         0000


BITS 119- 60                                              BYTE 05
000000000000 000000000000 000000000000 000000000000 000000000000
    0000         0000         0000         0000         0000


BITS 179-120                                              BYTE 10
000000000000 000000000000 000000000000 000000000000 000000000000
    0000         0000         0000         0000         0000


BITS 203-180                                              BYTE 15
                                       000000000000 000000000000
                                           0000         0000


                    CHANNEL 36 NOT AVAILABLE




                 ACTUAL S/C REGISTER ERROR BUFFER

         CHANNEL 16 REGISTER - WORDS 113-116
000000000000 000000000000 000000000000 000000000000 000000000000
000000000000 000000000000 000000000000 000000000000 000000000000
000000000000 000000000000 000000000000 000000000000 000000000000
                                       000000000000 000000000000
```

Figure D-1. Simulator K Display (Left Screen)

SIMULATOR COMMANDS

COMMAND	DESCRIPTION
AREA,A,M,Y.	SET M BITS FROM A TO OCTAL VALUE Y
BYTE,XX,YYYY.	SET BYTE XX TO OCTAL VALUE YYYY
CLEAR,A,B,...,Z.	CLEAR BITS A,B,...,Z
CYCLE,X,T,R.	SET BIT X EVERY 16*T MS. R TIMES
END.	END CYCLE COMMAND BEFORE R REACHED
LINE,X,Y.	SET LINE X TO OCTAL VALUE Y
READ.	READ S/C REGISTER INTO HOLDING REGISTER
SET,A,B,...,Z.	SET BITS A,B,...,Z
+	CHANGE REGISTER BEING USED AND K DISPLAY
GO.	ENTER HOLDING REGISTERS IN S/C REGISTERS
STOP.	END THE SIMULATOR

ALL BIT, BYTE, AND LINE NUMBERS ASSUMED DECIMAL.
TIME VALUES ASSUMED DECIMAL
Y AND YYYY VALUES MUST BE OCTAL.

Figure D-2. Simulator Commands K Display (Right Screen)

Batch Input

The simulator may also be called from batch input by using the SCRSIM command. The input file must have a record containing the commands to be processed, one command per card. The system must be in debug mode and the user must be validated for system origin privileges.

Simulator Commands

The simulator commands are described next. You must enter the entire command keyword and only one command may be entered at a time. Each command, except +, must end with a terminator. In all cases, a null argument is assumed to be zero.

Holding Register Commands

The following commands, except GO., affect only the holding register currently displayed on the left screen. These commands are used to set up the entire 204 bits in the holding registers. GO. transfers the holding register contents to the actual S/C register. (On a CYBER 70 Computer System, the channel 16 S/C register is simulated by the interlock register. GO. transfers the first 64 of 128 bits of the holding register to the interlock register.) The current contents of the holding register is displayed in binary and octal on the left screen (refer to figure D-1). The contents of the actual S/C register are also displayed in binary on the left screen.

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>AREA,a,m,y.</td><td>Set m bits in the holding register, from bit a to bit a+m-1, to the octal value y.
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>a</td><td>Starting bit number; from 0 to 203. a is assumed to be decimal, but a postradix of D or B may also be included.</td></tr>
<tr><td>m</td><td>Number of bits to be set. m is assumed to be decimal, but a postradix of D or B may also be included.</td></tr>
<tr><td>y</td><td>Value to which the bits are to be set. y may be up to m bits of octal value.</td></tr>
</table></td></tr>
<tr><td>BYTE,xx,yyyy.</td><td>Set byte xx in the holding register to the octal value yyyy.
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>xx</td><td>Byte number; from 0 to 16. xx is assumed to be decimal, but a postradix of D or B may be included.</td></tr>
<tr><td>yyyy</td><td>Value to which byte xx is to be set. yyyy may be up to 12 bits of octal value.</td></tr>
</table></td></tr>
<tr><td>CLEAR,a_1,a_2,...,a_n.</td><td>Clear bits a_1,a_2,...,a_n in the holding register. If more than 30 bit numbers are entered, only the first 30 are processed. All others are ignored.
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>a_i</td><td>Bit number to be cleared; from 0 to 203. A decimal value is assumed, but a postradix of D or B may be included.</td></tr>
</table></td></tr>
</table>

Command	Description
LINE,x,y.	Set line x of the holding register to the octal value y.

Parameter	Description
x	Line number of the holding register shown on the left display screen (refer to figure D-1); from 0 to 3. Line 0 is positioned at the top and line 3 is positioned at the bottom.
y	Value to which line x is to be set. y may be up to 60 bits of octal value.
READ.	Transfer the contents of the actual S/C register to the holding register.

Command	Description
SET,a_1,a_2,...,a_n.	Set bits a_1,a_2,...,a_n in the holding register. If more than 30 bit numbers are entered, only the first 30 are processed. All others are ignored.

Parameter	Description
a_i	Bit number to be set; from 0 to 203. A decimal value is assumed, but a postradix of D or B may be included.

Command	Description
GO.	Transfer the contents of the holding register to the actual S/C register or to the interlock register of a CYBER 70 Computer System. No bits in the holding register are changed by this command.

Cycle Commands

Command	Description
CYCLE,x,t,r.	Set bit x every t periods of time, a total of r times. This command assumes control of the simulator for the total time period specified. During this time, no command is accepted except END. Bit x is set in the holding register by this command.

Parameter	Description
x	Bit number to be set; from 0 to 203.
t	Number of periods of time for each cycle; one period is 16 milliseconds. t=32 is approximately 0.5 second.
r	Number of times to set bit x. r may not exceed 4095.

Command	Description
END.	End CYCLE command processing before r is reached. Control of the simulator is returned to the operator.

+ And Stop Commands

Command	Description
+	Toggle the K display between the channel 16 and channel 36 register displays and also change the register currently being worked on, if the simulator is being operated on a CYBER 170 Computer System with two S/C registers. If two S/C registers are not present on the machine, no action is taken. The channel 16 register is assumed when the simulator begins. The only holding register that is affected by the holding register and cycle commands is the one currently displayed.
STOP.	End simulator processing.

580 Programmable Format Control E

You control the spacing and format on 580 line printers by using carriage control format tapes or programmable format control. The carriage control format tape is punched to indicate particular format channels for each frame. A printer with programmable format control does not use carriage control format tapes; instead, it contains a microprocessor plus memory. Programmable format control arrays are loaded into this memory, performing the same function as the format tape. This appendix describes how to create and load programmable format control arrays. A description of format tapes is included in the NOS Version 2 Reference Set, Volume 3.

Certain 580 printers are not equipped with a carriage control tape; instead, a microprocessor plus memory called programmable format control is used. Instead of a tape controlling the page format, software is used. This software is called a programmable format control array. A programmable format control array consists of numbers from 0 to 12 and 17_8. Each nonzero character represents a channel. A 0 (zero) specifies that no channel is selected. An array is similar to a format tape since each number in a programmable format control array corresponds to a line on the print form. As each line of a page is printed, the next number in the programmable format control array buffer is addressed. A carriage control character in column 1 of the output line, indicating a skip to a particular channel, causes the memory in the programmable format control array buffer to be sequentially addressed until the particular number is found. The paper is spaced a similar number of spaces.

A programmable format control array differs from a format tape because only one channel can be specified per line. A number of channels can be specified per line using a format tape. Also, programmable format control arrays are accessed in pairs, one for 6 lines per inch (lpi) printing and one for 8-lpi printing. The 8-lpi array is usually larger, allowing more lines to be printed on the same size page.

Building Programmable Format Control Arrays

Observe the following rules when building a programmable format control array.

- Enter only valid numbers (from 0 to 12 and 17_8) in the programmable format control array.
- A 1 must be the first number in the programmable format control array, indicating a top-of-form position.
- A 12 should always indicate the last line of the form (bottom of page).
- A 17_8 should appear as the last number in the array, denoting the end of valid numbers for a given array. This number does not correspond to any particular line on the form.
- Maximum length programmable format control arrays (132 for 6 lpi and 176 or 136 for 8 lpi) must include a 9 only at location 132 for 6 lpi and location 176 or 136 for 8 lpi. Improper paper alignment may occur if a 9 is placed elsewhere.

To properly load the appropriate programmable format control buffer for a particular form, it is necessary to assemble data that will contain, when transmitted and stored in the programmable format control array buffer, as many numbers as lines on the form. As stored within the programmable format control array buffer, each number is a 4-bit code used to represent channels (1 through 12) or a null code (no channel selected).

NOTE

Channels 13 and 14 are valid channels but they are not selectable. No programmable format control error occurs when loading these numbers into the programmable format control buffer.

The maximum capacity of the 6-lpi programmable format control buffer is 132 numbers (22-inch form maximum) plus the last line number whereas the 8-lpi programmable format control buffer has a capacity of 176 or 136 numbers (22-inch or 17-inch form maximum) plus the last line number. Fewer than the maximum amount of numbers may be loaded into the programmable format control buffer, but an excessive amount of numbers will cause a programmable format control overflow error.

Adding Programmable Format Control Arrays

Programmable format control arrays must occur in pairs (one 6-lpi array and one 8-lpi array); therefore, when a particular array is specified with the SC option of the ROUTE command, it is possible to switch from 6- to 8-lpi spacing except when using 8.5-inch forms, which are always printed at 8-lpi spacing. Four pairs of arrays are provided with the operating system, two pairs of arrays for 11-inch forms and two pairs of arrays for 8.5-inch forms (refer to table E-1). The numbers are omitted for 6-lpi 8.5-inch forms because this combination is not selectable. The form or paper size is defined in the EQPDECK unit record equipment EST entry.

To add additional programmable format control arrays, BIO must be modified to contain these additional arrays as overlays in QAP. The overlay names and corresponding ROUTE command SC parameter options must be added to PFCO, and the programmable format control overlay table in QAP. The first entry in this table specifies the default programmable control format array. This array is used when the SC parameter is not specified. If an added array is to be the default array, the PFCO entry for it should be placed at the beginning of the table. The value that must be specified with the SC parameter is included in this entry. This value can be any 6-bit binary value. The arrays that are also added to QAP must follow the same format, structure, and labeling conventions as the arrays already provided in QAP.

To generate the numbers to be loaded into the programmable format control buffer, the DPFC (define programmable format control code) macro is used with the following format. Only the lower 4 bits of each 6 bits are used as programmable format control numbers.

Location	Operation	Variable
	DPFC	a,b,c,d

Variable	Description
a, b, c, d	Programmable format control numbers for 6 or 8 lpi. These numbers can be any value from 0 to 12 and 17_8.

NOTE

In each macro call, all four parameters must be specified (0 denotes no channel is selected). If all four are not required to complete an array (the number of lines on the form is not a multiple of 4), zeros should be used for the remaining parameters.

Table E-1. Released Programmable Format Control Arrays

Location	11 in.[1]	8.5 in.[1]	11 in.[2]	8.5 in.[2]	11 in.[3]	8.5 in.[3]	11 in.[4]	8.5 in.[4]
1	1		1		1	1	1	1
2	6		6		6	6	6	6
3	0		0		0	0	0	0
4	2		2		0	2	0	2
5	0		0		2	0	2	0
6	0		0		0	0	0	0
7	3		3		0	3	0	3
8	0		0		0	0	0	0
9	0		0		3	0	3	0
10	4		4		0	4	0	4
11	0		0		0	0	0	0
12	0		0		0	0	0	0
13	5		5		4	5	4	5
14	0		0		0	0	0	0
15	0		0		0	0	0	0
16	2		7		0	2	0	7
17	0		0		5	0	5	0
18	0		0		0	0	0	0
19	3		8		0	3	0	8
20	0		0		0	0	0	0
21	0		0		2	0	7	0
22	4		10		0	4	0	10
23	0		0		0	0	0	0
24	0		0		0	0	0	0
25	5		11		3	5	8	11
26	0		0		0	0	0	0
27	0		0		0	0	0	0
28	2		2		0	2	0	2
29	0		0		4	0	10	0
30	0		0		0	0	0	0
31	3		3		0	3	0	3
32	0		0		0	0	0	0
33	0		0		5	0	11	0
34	4		4		0	4	0	4
35	0		0		0	0	0	0

1. 6 lpi default (default is SC=).

2. 6 lpi alternate (alternate is SC=1).

3. 8 lpi default.

4. 8 lpi alternate.

(Continued)

Table E-1. **Released Programmable Format Control Arrays** *(Continued)*

Location	11 in.[1]	8.5 in.[1]	11 in.[2]	8.5 in.[2]	11 in.[3]	8.5 in.[3]	11 in.[4]	8.5 in.[4]
36	0		0		0	0	0	0
37	5		5		2	5	2	5
38	0		0		0	0	0	0
39	0		0		0	0	0	0
40	2		7		0	2	0	7
41	0		0		3	0	3	0
42	0		0		0	0	0	0
43	3		8		0	3	0	8
44	0		0		0	0	0	0
45	0		0		4	0	4	0
46	4		10		0	4	0	10
47	0		0		0	0	0	0
48	0		0		0	0	0	0
49	5		11		5	5	5	11
50	0		0		0	0	0	0
51	0		0		0	0	0	0
52	2		2		0	2	0	2
53	0		0		2	0	7	0
54	0		0		0	0	0	0
55	3		3		0	3	0	3
56	0		0		0	0	0	0
57	0		0		3	0	8	0
58	4		4		0	4	0	4
59	0		0		0	0	0	0
60	0		0		0	0	0	0
61	5		5		4	5	10	5
62	0		0		0	0	0	0
63	0		0		0	0	0	0
64	12		12		0	2	0	7
65	0		0		5	0	11	0
66	0		0		0	12	0	12
67	1		1		0	0	0	0
68	0		0		0	0	0	0
69	0		0		2	1	2	1
70	2		2		0	0	0	0
71	0		0		0	0	0	0

1. 6 lpi default (default is SC=).

2. 6 lpi alternate (alternate is SC=1).

3. 8 lpi default.

4. 8 lpi alternate.

(Continued)

Table E-1. Released Programmable Format Control Arrays *(Continued)*

Location	11 in.[1]	8.5 in.[1]	11 in.[2]	8.5 in.[2]	11 in.[3]	8.5 in.[3]	11 in.[4]	8.5 in.[4]
72	0		0		0	2	0	2
73	3		3		3	0	3	0
74	0		0		0	0	0	0
75	0		0		0	3	0	3
76	4		4		0	0	0	0
77	0		0		4	0	4	0
78	0		0		0	4	0	4
79	5		5		0	0	0	0
80	0		0		0	0	0	0
81	0		0		5	5	5	5
82	2		7		0	0	0	0
83	0		0		0	0	0	0
84	0		0		0	0	0	7
85	3		8		12	0	12	0
86	0		0		0	0	0	0
87	0		0		0	3	0	8
88	4		10		0	0	0	0
89	0		0		1	0	1	0
90	0		0		0	4	0	10
91	5		11		0	0	0	0
92	0		0		0	0	0	0
93	0		0		2	5	2	11
94	2		2		0	0	0	0
95	0		0		0	0	0	0
96	0		0		0	2	0	2
97	3		3		3	0	3	0
98	0		0		0	0	0	0
99	0		0		0	3	0	3
100	4		4		0	0	0	0
101	0		0		4	0	4	0
102	0		0		0	4	0	4
103	5		5		0	0	0	0
104	0		0		0	0	0	0
105	0		0		5	5	5	5
106	2		7		0	0	0	0
107	0		0		0	0	0	0

1. 6 lpi default (default is SC=).

2. 6 lpi alternate (alternate is SC=1).

3. 8 lpi default.

4. 8 lpi alternate.

(Continued)

Table E-1. Released Programmable Format Control Arrays *(Continued)*

Location	11 in.[1]	8.5 in.[1]	11 in.[2]	8.5 in.[2]	11 in.[3]	8.5 in.[3]	11 in.[4]	8.5 in.[4]
108	0		0		0	2	0	7
109	3		8		2	0	7	0
110	0		0		0	0	0	0
111	0		0		0	3	0	8
112	4		10		0	0	0	0
113	0		0		3	0	8	0
114	0		0		0	4	0	10
115	5		11		0	0	0	0
116	0		0		0	0	0	0
117	0		0		4	5	10	11
118	2		2		0	0	0	0
119	0		0		0	0	0	0
120	0		0		0	2	0	2
121	3		3		5	0	11	0
122	0		0		0	0	0	0
123	0		0		0	3	0	3
124	4		4		0	0	0	0
125	0		0		2	0	2	0
126	0		0		0	4	0	4
127	5		5		0	0	0	0
128	0		0		0	0	0	0
129	0		0		3	5	3	5
130	12		12		0	0	0	0
131	0		0		0	0	0	0
132	9		9		0	2	0	7
133	17_8	17_8			4	0	4	0
134					0	12	0	12
135					0	0	0	0
136					0	9	0	9
137					5	17_8	5	17_8
138					0		0	
139					0		0	
140					0		0	
141					2		7	
142					0		0	
143					0		0	

1. 6 lpi default (default is SC=).

2. 6 lpi alternate (alternate is SC=1).

3. 8 lpi default.

4. 8 lpi alternate.

(Continued)

Table E-1. Released Programmable Format Control Arrays *(Continued)*

Location	11 in.[1]	8.5 in.[1]	11 in.[2]	8.5 in.[2]	11 in.[3]	8.5 in.[3]	11 in.[4]	8.5 in.[4]
144					0		0	
145					3		8	
146					0		0	
147					0		0	
148					0		0	
149					4		10	
150					0		0	
151					0		0	
152					0		0	
153					5		11	
154					0		0	
155					0		0	
156					0		0	
157					2		2	
158					0		0	
159					0		0	
160					0		0	
161					3		3	
162					0		0	
163					0		0	
164					0		0	
165					4		4	
166					0		0	
167					0		0	
168					0		0	
169					5		5	
170					0		0	
171					0		0	
172					0		0	
173					12		12	
174					0		0	
175					0		0	
176					9		9	
177					17_8		17_8	

1. 6 lpi default (default is SC=).

2. 6 lpi alternate (alternate is SC=1).

3. 8 lpi default.

4. 8 lpi alternate.

533/536, 537, and 585 Printer Electronic Vertical Format Unit (EVFU) F

Upon startup, the printer support utility (PSU) looks for an EVFU load file which is a public file named EVFULFN under the network user name NETOPS. The EVFU load file contains definitions which describe all 533, 536, 537, and 585 printers to be serviced by PSU. In addition, for 533/536 printers it contains directives that describe the actions to be taken while printing files that have format control characters in column 1 and the page length and print density specified. EVFU information for 537 and 585 printers is supplied by CDCNET, and is documented in the CDCNET Configuration and Site Administration Guide.

PSU supports up to 12 printers with any mix of 533/536, 537, and 585 printers. The printer directives in the released EVFU file specify PRINT01 through PRINT04 as 533/536 printers; PRINT05 and PRINT06 as 537 interactive printers and PRINT07 and PRINT08 as 537 batch printers; and PRINT09 through PRINT12 as 585 printers. If these printer names are not correct for the printers being used, the EVFU file will have to be changed accordingly. If 533/536 printers are used, the 533/536 printer definitions must follow all other printer definitions, which are then followed by the 533/536 EVFU load data. If a mismatch occurs between the EVFU load file printer definition and the connection received from the network, the connection is placed in a HOLD state and a K-display message provides the pertinent information. If the K-display is not assigned when this occurs, a flashing B-display message requests that the K-display be assigned to PSU.

EVFU Directives for 533/536 Printers

EVFU directives follow the printer definitions in the EVFU load file. The presence of a character in column 1 initiates a new EVFU directive. A line with a blank in column 1 is a continuation of the line that started the current EVFU directive. Blank lines are not permitted in the EVFU load file. An EVFU directive terminates with a semicolon or end-of-line. You can add comments to an EVFU directive line following the semicolon.

An EVFU load file contains two types of EVFU directives:

- Format control directives.
- EVFU load directives.

Format Control Directives

The character in column 1 of each line of a print file is a format control character. PSU translates this character into a string of one or more data characters that are output instead of the format control character.

A format control directive consists of a single format control character followed by the associated data for PSU to output in place of the format control character specified in the directive. At least one space must separate the format control character from the data characters. There are two formats for the data characters, and you can intermix the two formats. The first data format is pairs of hexadecimal digits representing ASCII characters. These pairs of digits may be separated from each other by spaces for readability. The second data format is ASCII strings delimited by quotation marks, for example, "data".

You can specify a maximum of six data items in a format control directive. They can be hexadecimal digit pairs or ASCII strings or both, but not more than a total of six items representing ASCII characters. Character data should not extend over line boundaries.

PSU supports the following format control characters:

Character	Description
1	Eject page before printing.
2	Skip to last line of form before printing.
+	Skip zero lines before printing (overprint).
space	Single space.
0	Skip one line before printing (double space).
-	Skip two lines before printing (triple space).
8	Skip to top of page (channel 1) before printing.
7	Skip two lines (channel 2) before printing.
6	Skip three lines (channel 3) before printing.
5	Skip four lines (channel 4) before printing.
4	Skip five lines (channel 5) before printing.
3	Skip one page (channel 6) before printing.
9	Skip seven lines (channel 7) before printing.
X	Skip eight lines (channel 8) before printing.
Y	Skip three lines (channel 9) before printing.
Z	Skip ten lines (channel 10) before printing.
W	Eject page (channel 11) before printing.
U	Eject page (channel 12) before printing.
A	Eject page after printing.
B	Skip to last line of form after printing.
/	Skip zero lines after printing (overprint).
H	Skip to top of page (channel 1) after printing.
G	Skip two lines (channel 2) after printing.
F	Skip three lines (channel 3) after printing.
E	Skip four lines (channel 4) after printing.
D	Skip five lines (channel 5) after printing.
C	Skip one page (channel 6) after printing.
I	Skip seven lines (channel 7) after printing.
J	Skip eight lines (channel 8) after printing.
K	Skip three lines (channel 9) after printing.
L	Skip ten lines (channel 10) after printing.
M	Eject page (channel 11) after printing.
N	Eject page (channel 12) after printing.
Q	Clear auto page eject.
R	Set auto page eject.
S	Set 6-lpi print density.
T	Set 8-lpi print density.

If column 1 of a format control directive contains a blank or any character that has not been defined, PSU performs a line termination and line feed sequence. Thus, the concept of an invalid format control character does not exist. Lines with something other than a defined character in column 1 will have that character replaced by a blank. Following the line feed, processing of the rest of the line will continue as normal.

Note that Q, R, S, and T may appear either as format control characters or as EVFU load characters. However, if both are specified, the format control character takes precedence and the EVFU load character will not be used.

The level number mentioned earlier corresponds to various combinations of page size and print density or spacing codes that can be specified. The following EVFU level numbers are defined for the default EVFU load file:

Level Number	Description
1	Defines a 12-inch form length with a 6-lpi print density.
2	Defines a 12-inch form length with an 8-lpi print density.
3	Defines an 11-inch form length with a 6-lpi print density.
4	Defines an 11-inch form length with an 8-lpi print density.
5	Defines an 8.5-inch form length with a 6-lpi print density.
6	Defines an 8.5-inch form length with an 8-lpi print density.

If you specify a spacing code or request a forms length/density for which no load data exists, PSU will issue a message to the printer, then rewind the output file, and print it using the default EVFU level of 3.

EVFU Load File Example

Figure F-1 provides an example of an EVFU Load File. It shows the required order of directives for a 533/536 printer EVFU load file; printer definitions are first, format control directives are next, and EVFU load directives are last.

In this example, 1B 5C is the trailer for all EVFU data except for the print density selection. For format control directives (skip to channel), the leader is 1B 50 22. For EVFU load directives, the leader is 1B 50 23. However, the printer paper must be positioned at the top of form any time the EVFU is changed because that is how the printer determines the top of form (when the EVFU is changed). Hence, all EVFU load data starts with 0C (form feed). The 1B 63 in the V load directive is a printer reset. A T specifies the total number of lines on the form (T66). An L specifies where to set a stop in a particular line; for example L41;4;5 sets stops at positions 4 and 5 in line 41. Similarly, a C clears a stop. Items following the leader 1B 50 23 and preceding the trailer 1B 5C are separated by semicolons.

```
PRINT09,PC=CDC585,DOWN.     ┐
PRINT10,PC=CDC585,DOWN.     │
PRINT11,PC=CDC585,DOWN.     │
PRINT12,PC=CDC585,DOWN.     │
PRINT05,PC=C537INT,DOWN.    │
PRINT06,PC=C537INT,DOWN.    } Printer Definitions
PRINT07,PC=C537BAT,DOWN.    │
PRINT08,PC=C537BAT,DOWN.    │
PRINT01,PC=C533536,DOWN.    │
PRINT02,PC=C533536,DOWN.    │
PRINT03,PC=C533536,DOWN.    │
PRINT04,PC=C533536,DOWN.    ┘
8 1B 50 22 "1" 1B 5C        ┐
7 1B 50 22 "2" 1B 5C        │
6 1B 50 22 "3" 1B 5C        │
5 1B 50 22 "4" 1B 5C        │
4 1B 50 22 "5" 1B 5C        │
3 1B 50 22 "6" 1B 5C        │
H 1B 50 22 "1" 1B 5C        } Format Control
G 1B 50 22 "2" 1B 5C        │ Directives
F 1B 50 22 "3" 1B 5C        │
E 1B 50 22 "4" 1B 5C        │
D 1B 50 22 "5" 1B 5C        │
C 1B 50 22 "6" 1B 5C        ┘
Q3 0C 1B 50 23 "C64;2;L66;2" 1B 5C                   ┐            ;CLEAR AUTO PAGE EJECT
R3 0C 1B 50 23 "C66;2;L64;2" 1B 5C                   │            ;SET AUTO PAGE EJECT
S3 0C 1B 50 23 "T66;L64;2;C65;4;C66;5" 1B 5C         │            ;FF, 66 LPF, BOF
   1B 5B 31 32 30 20 47                              │            ;SELECT 6 LPI
T4 0C 1B 50 23 "T88;C64;2;7;L65;4;L86;2;7" 1B 5C     │            ;FF, 88 LPF, BOF
   1B 5B 39 30 20 47                                 │            ;SELECT 8 LPI
V3 0C 1B 63 1B 50 23 "T66;"                          │            ;FF, RESET, LINES/FORM
   "L1;1;3;4;5;6;7;L3;7;L4;3;L5;4;7;"                │            ;LINES 1 - 5
   "L6;5;L7;3;7;L9;4;7;L10;3;"                       │            ;LINES 6 - 10
   "L11;5;7;L13;3;4;7;L15;7;"                        │            ;LINES 11 - 15
   "L16;3;5;L17;4;L19;3;7;"                          │            ;LINES 16 - 20
   "L21;4;5;7;L22;3;L23;7;L25;3;4;7;"                │            ;LINES 21 - 25
   "L26;5;L27;7;L28;3;L29;4;7;"                      │ EVFU       ;LINES 26 - 30
   "L31;3;5;7;L33;4;7;L34;3;L35;7;"                  } Load       ;LINES 31 - 35
   "L36;5;L37;3;4;7;L39;7;L40;3;"                    │ Directives;LINES 36 - 40
   "L41;4;5;7;L43;3;7;L45;4;7;"                      │            ;LINES 41 - 45
   "L46;3;5;L47;7;L49;3;4;7;"                        │            ;LINES 46 - 50
   "L51;5;7;L52;3;L53;4;7;L55;3;7;"                  │            ;LINES 51 - 55
   "L56;5;L57;4;7;L58;3;L59;7;"                      │            ;LINES 56 - 60
   "L61;3;4;5;7;L63;7;L64;2;3;"                      │            ;LINES 61 - 65
   "L67;3;7;L69;4;7;L70;3;"                          │            ;LINES 66 - 70
   "L71;5;7;L73;3;4;7;L75;7;"                        │            ;LINES 71 - 75
   "L76;3;5;L77;4;7;L79;3;7;"                        │            ;LINES 76 - 80
   "L81;4;5;7;L82;3;L83;7;L85;3;4;7;"                │            ;LINES 81 - 85
   "L86;2;5"                                         │            ;LINES 86 - 88
   1B 5C                                             ┘            ;TERMINATE
```

Figure F-1. EVFU Load File

EVFU Load Image

Table F-1 shows the released EVFU load image for 533/536 printers. The channel numbers in the format control character descriptions correspond to the channel numbers in table F-1. For example, format control character Y causes the printer to skip three lines before printing. It does this by using channel 9 in the EVFU load image.

Table F-1. Released EVFU Load Image: Print Lines and Channel Numbers

Print Line	1	2	3	4	5	6	7	8	9	10	11	12
1	X	X	X	X	X	X	X	X	X	X		
2												
3	X											
4		X						X				
5	X		X									
6				X								
7	X	X						X				
8						X						
9	X		X				X					
10		X						X				
11	X			X					X			
12												
13	X	X	X					X				
14												
15	X					X						
16		X		X				X				
17	X		X				X					
18												
19	X	X						X				
20												
21	X		X	X					X			
22		X				X		X				
23	X											
24												
25	X	X	X				X	X				
26				X								
27	X											
28		X						X				
29	X		X			X						
30												
31	X	X		X				X	X			
32												
33	X		X				X					
34		X						X				
35	X											
36				X		X						
37	X	X	X					X				
38												

(Continued)

Table F-1. Released EVFU Load Image: Print Lines and Channel Numbers *(Continued)*

Print Line	1	2	3	4	5	6	7	8	9	10	11	12
39	X											
40		X						X				
41		X		X	X			X		X		
42												
43		X	X				X		X			
44												
45		X		X								
46			X		X				X			
47		X										
48												
49		X	X	X				X	X		1	1
50							X					
51		X			X					X		
52			X						X			
53		X		X								
54												
55		X	X						X			
56					X							
57		X		X			X	X				
58			X						X			
59		X										
60												
61		X	X	X	X				X	X		
62												
63		X										
64			X				X		X		2	2
65		X		X				X				
66					X						3	3
67		X	X						X			
68												
69		X		X								
70			X						X		4	4

1. Set for 8.5 inch form at 6 lines per inch.
2. Set for 11 inch form at 6 lines per inch.
3. Set for 8.5 inch form at 8 lines per inch.
4. Set for 11 inch form at 8 lines per inch.

(Continued)

Table F-1. Released EVFU Load Image: Print Lines and Channel Numbers *(Continued)*

Print Line	1	2	3	4	5	6	7	8	9	10	11	12
71		X			X		X			X		
72												
73		X	X	X				X	X			
74												
75		X										
76			X		X				X			
77		X		X								
78							X					
79		X	X						X			
80												
81		X		X	X			X		X		
82			X						X			
83		X										
84												
85		X	X	X			X		X			
86					X						[1]	[1]
87		X										
88			X						X			
89		X		X				X				
90												
91		X	X		X				X	X		
92							X					
93		X		X								
94			X						X		[2]	[2]
95		X										
96					X							

1. Set for 12 inch form at 6 lines per inch.

2. Set for 12 inch form at 8 lines per inch.

881/883 Pack Reformatting Utility G

Each 881 disk pack used in the 844 disk contains factory-recorded flawing information on cylinder 632_8 (410), track 0, sectors 0, 1, and 2. Each 883 pack contains this information on cylinder 1466_8 (822), track 0, sectors 0, 1, and 2. The following information is included on the cylinders.

- Cylinder 632_8 (or 1466_8 for 883 packs), track 0, sector 0 contains the factory-recorded manufacturing data. This data consists of the pack serial number and the manufacturing date.
- Cylinder 632_8 (or 1466_8), track 0, sector 1 contains the factory map. This map contains a list of all factory-detected flaws, both correctable and uncorrectable.
- Cylinder 632_8 (or 1466_8), track 0, sector 2 contains the utility map. This map originally contains all factory-detected uncorrectable flaws. This map is updated by the reformatting utility.

FORMAT is a CPU program that operates in conjunction with FDP, a PP program, to maintain and reformat 881/883 disk packs. It is used to perform the following functions.

- Factory-recorded manufacturing data, factory-recorded flaw data, and utility flaw data can be retrieved from a factory-formatted disk pack.
- Sector and track flaws can be set or cleared on a factory-formatted disk pack.
- Address fields of a previously factory-formatted disk pack can be restored. (This function is used only in the event that addresses on the pack are lost.)

In order to function, FORMAT requires that the factory-recorded data [sectors 0 and 1 of cylinder 632_8 (or 1466_8)] be correct and readable. The pack cannot be processed if this data is unreadable. If packs are available that do not contain this factory-recorded information, consult a customer engineer to have this information placed on the packs. Also, the correct level of controlware must be present in order for FORMAT to function. To determine the controlware level and for procedures to install this controlware, refer to the NOS Version 2 Installation Handbook. Since the operating system requires that the utility map contain the physical flaw information in order for automatic logical flawing to be performed, it is important that the utility map be properly maintained.

The operating system automatically sets logical flaws when initializing 844 equipment. This is done by reading the utility map of the 844 units involved, and mapping this physical flaw information into the corresponding logical track addresses. Logical track flaw reservations are then made in the track reservation table (TRT) for the 844 equipment being installed. For example, if the 844 equipment being initialized consists of two physical units (such as a DI-2 configuration), the logical flaws set in the TRT are obtained from the physical flaw information recorded in the utility maps of both units making up the DI-2 configuration. This automatic flawing occurs when an equipment is initialized, regardless of whether the initialization is done during deadstart or online. Automatic flawing also occurs when an X.FLAW request is made from the console.

The operating system allows for setting and clearing flaw information. The SLF and CLF APRDECK entries set or clear logical track reservations in the TRT of the equipment. (Refer to section 3, Deadstart Decks, for information concerning these entries.) If the device is then checkpointed, this flaw information is preserved in the TRT portion of the device label. The SLF entry can be made during deadstart, during online initialization, or by using the FLAW utility (as described in section 8, K-Display Utilities). In any case, the flawing done via these entries is only logical; the flaw information remains only in the TRT and is discarded on subsequent deadstart initialization. This information is also lost during online initializations if it was not possible to recover the equipment. Using SLF does not cause any additional information to be recorded in the utility map; only FORMAT is capable of updating the utility map data. Caution should be used if attempts are made to cancel a logical flaw that was made during automatic flawing, since the physical disk sector is still marked as flawed and attempts to access that sector yield error conditions.

The use of the FORMAT utility for maintaining and reformatting 881/883 disk packs is described next.

FORMAT Command

Processing maintenance operations on an 881/883 type disk pack is initiated by the FORMAT command. This program interfaces with you as required. The format of this command is:

FORMAT, p_1, p_2, . . . , p_n.

Each p_i is a keyword or a keyword equated to a value.

p_i	Description
G=m	Relative unit of a multispindle device. This value is checked for validity within the device. For example, if the device is a DI-2 and G=2 is specified, an error results.
I=infile	File on which input directives and data are written.
I	Same as I=INPUT.
L=outfile	Output file on which the information extracted from the disk pack is to be written. (Refer to Output Formats later in this appendix.)
L	Same as L=OUTPUT.

pi	Description
MODE=mode	Operational mode for FORMAT.

mode	Description
ALTER	The input file contains directives to control the set or clear flaw operations (refer to Input Formats later in this appendix).
FETCH	The factory-recorded flawing information contained on cylinder 632_8 (or 1466_8), track 0, sectors 0, 1, and 2 is obtained and copied to the output file (and optional output file, if available).
RESTORE	The addresses, flawed sectors, and tracks are restored according to information given in the utility flaw map. If the utility flaw map is not intact, the program aborts.

pi	Description
MODE	Same as MODE=FETCH.
O=filename	Optional output file to contain the output extracted from the disk pack.

NOTE

If output files other than OUTPUT or optional output files are specified, they should be created prior to the initiation of FORMAT. If they are not, they are destroyed upon completion of FORMAT processing.

pi	Description
P=serialnumber	Pack serial number in decimal of the pack to be processed. If serialnumber does not match the serial number recorded on the disk pack at the factory, processing does not occur.
P	Same as P=0.
U=est	EST ordinal of the 844 drive on which the disk pack is mounted. The unit is checked to ensure that it is available for formatting (refer to Accessing Disk Devices later in this appendix).
V	Specifies that the utility is to verify the addresses recorded on the disk pack. This parameter is valid only if MODE=FETCH or MODE=RESTORE is specified.

Example:

If all default values are used, the following call is made.

```
FORMAT,I=INPUT,L=OUTPUT,MODE=FETCH,P=0.
```

At least the U and the P parameter must be correctly specified to initiate processing.

Input Formats

Input to FORMAT consists of control directives and data statements. Control directives specify the type of operation to be performed. Data statements specify locations on the pack where the operations are to be performed. A number of data statements may follow each control directive. Control directives and data statements are contained on the input file. This file is accessed only when MODE=ALTER has been specified on the FORMAT command. The input file (and therefore, control directives and data statements) has no significance when MODE=FETCH or MODE=RESTORE is specified.

Control Directives

Control directives begin in column 1. The format is:

```
directive
```

The following are acceptable directives.

Directive	Description
SET	Declares that the following data statements contain the addresses of flaws to be set and entered in the utility flaw map.
CLEAR	Declares that the following data statements contain the addresses of flaws to be cleared and deleted from the utility flaw map.
FINIS	Declares the end of the input. No information following this directive is processed. This directive is optional.

SET and CLEAR directives may be intermixed in the input file. However, all CLEAR operations are performed before any SET operation. Any attempt to alter the factory map or to set or clear sector flaws in a previously flawed track results in an error.

Data Statements

Data statements begin in column 1. The format is:

```
x,cccc,tt,ss
```

Parameter	Description
x	Type of flaw to be set or cleared. Acceptable values are: **x** — **Description** S — Specifies that the SET or CLEAR directive applies to a sector. T — Specifies that the SET or CLEAR directive applies to a track.
cccc	Octal number specifying the cylinder; from 0 to 632_8 (or 1466_8 for 883 packs).
tt	Octal number specifying the track; from 0 to 22_8.
ss	Octal number specifying the sector; from 0 to 27_8. This field is ignored for track flaws (x=T).

All input data is checked to ensure that the values are within range. Any errors in input result in the termination of the utility before the disk is accessed. Any attempt to alter the factory map, or to set or clear sector flaws in a previously flawed track results in an error.

A maximum of 157 data statements can appear in the input stream.

Output Formats

Output generated by FORMAT is placed on the output file (L=filename on the FORMAT command). This file, for all modes of operation (ALTER, FETCH, and RESTORE), contains:

- A listing of the input stream, if any.
- The pack serial number and date of factory formatting on cylinder 632_8 (or 1466_8), track 0, sector 0.
- A listing of the factory flaw map contained on cylinder 632_8 (or 1466_8), track 0, sector 1.
- A listing of the utility flaw map contained on cylinder 632_8 (or 1466_8), track 0, sector 2.
- A listing of the utility flaw map following any changes resulting from SET or CLEAR directives. This listing appears only when MODE=ALTER is specified on the FORMAT command.
- A listing of the flawed sectors and tracks as read from the disk during address verification. This listing appears only when MODE=FETCH or MODE=RESTORE, and the V parameter are specified on the FORMAT command.

The output generated by FORMAT can be directed to an optional output file (O=filename). This file can then be used as input to another program, or it can be punched or printed.

The following three examples of standard output illustrate a series of reformatting operations performed on the same pack.

Example 1:

A RESTORE operation is performed on an 881 pack. A command similar to the following was entered.

```
FORMAT,U=est,P=819545,MODE=RESTORE.
```

Figure G-1 illustrates the resulting output.

```
DISK PACK REFORMATTING UTILITY              -VERSION 1.1 -  82/01/25.
MODE = RESTORE


           DISK PACK SERIAL NUMBER
            819545

           DATE OF ORIGINAL FACTORY FORMATTING
            74/04/30

           FACTORY FLAW MAP
           (C=CORRECTABLE ERROR,S=SECTOR FLAW,T=TRACK FLAW)
           S,632, 00, 00
           S,632, 00, 01
           S,632, 00, 02
           T,302, 16, 00
           T,362, 01, 00
           T,373, 21, 00
           S,626, 15, 15

           UTILITY FLAW MAP
           (S=SECTOR FLAW,T=TRACK FLAW)
           MAP EMPTY

PACK FORMATTING COMPLETE, VERIFICATION FOLLOWS
           S,632, 00, 00
           S,632, 00, 02
           S,632, 00, 01

ADDRESS VERIFICATION COMPLETE
```

Figure G-1. FORMAT Output, MODE=RESTORE

Example 2:

The flaws noted in the factory flaw map from example 1 (refer to figure G-1) are now set in the utility flaw map.

Input similar to the following was entered.

```
FORMAT,U=est,P=819545,MODE=ALTER.
--EOR--
SET
S,626,15,15
T,302,16,0
T,362,01,00
T,373,21,00
FINIS
--EOI--
```

Figure G-2 illustrates the resulting output.

```
DISK PACK REFORMATTING UTILITY                   -VERSION 1.1 -      82/01/25.
MODE = ALTER

INPUT DATA
SET
S,626, 15, 15
T,302, 16, 00
T,362, 01, 00
T,373, 21, 00
FINIS


               DISK PACK SERIAL NUMBER
                819545

               DATE OF ORIGINAL FACTORY FORMATTING
                74/04/30

               FACTORY FLAW MAP
               (C=CORRECTABLE ERROR, S=SECTOR FLAW, T=TRACK FLAW)
               S,632, 00, 00
               S,632, 00, 01
               S,632, 00, 02
               T,302, 16, 00
               T,362, 01, 00
               T,373, 21, 00
               S,626, 15, 15

               UTILITY FLAW MAP
               (S=SECTOR FLAW, T=TRACK FLAW)
               MAP EMPTY

               UTILITY FLAW MAP    (ALTERED)
               (S=SECTOR FLAW, T=TRACK FLAW)
               S,626, 15, 15
               T,302, 16, 00
               T,362, 01, 00
               T,373, 21, 00
```

Figure G-2. FORMAT Output, MODE=ALTER

Example 3:

A FETCH with verification operation is performed to ensure proper reformatting. A command similar to the following was entered.

```
FORMAT,U=est,P=819545,MODE=FETCH,V.
```

Figure G-3 illustrates the resulting output.

```
DISK PACK REFORMATTING UTILITY              -VERSION 1.1 -      82/01/25.
MODE = FETCH

           DISK PACK SERIAL NUMBER
            819545

           DATE OF ORIGINAL FACTORY FORMATTING
            74/04/30

           FACTORY FLAW MAP
           (C=CORRECTABLE ERROR, S=SECTOR FLAW, T=TRACK FLAW)
           S,632, 00, 00
           S,632, 00, 01
           S,632, 00, 02
           T,302, 16, 00
           T,362, 01, 00
           T,373, 21, 00
           S,626, 15, 15

           UTILITY FLAW MAP
           (S=SECTOR FLAW,T=TRACK FLAW)
           S,626, 15, 15
           T,302, 16, 00
           T,362, 01, 00
           T,373, 21, 00


ADDRESS VERIFICATION FOLLOWS
           T,302, 16, 00
           T,362, 01, 00
           T,373, 21, 00
           S,626, 15, 15
           S,632, 00, 00
           S,632, 00, 02
           S,632, 00, 01


ADDRESS VERIFICATION COMPLETE
```

Figure G-3. FORMAT Output, MODE=FETCH

Accessing Disk Devices

Special procedures must be used to access the 844 drive used in the reformatting utility. Since certain FORMAT operations (ALTER and RESTORE) can change addresses on the pack, user access to the pack must be restricted.

FORMAT can operate on the pack as follows.

- A read operation (FETCH) obtains formatting information from the pack. The integrity of the pack is maintained.
- Read and write operations (ALTER and RESTORE) can set and clear flaws, and addresses can be rewritten. Users cannot place permanent files on the pack when these operations occur. The integrity of the data on the pack is lost, so a full initialization of the pack must occur before system usage occurs.

Access for Read Operations

Accessing the pack for read operations requires that you specify the U parameter on the FORMAT command with the correct EST ordinal of the device containing the pack. In this case, the device must be a single-spindle device unless you also specify the G parameter. You must also specify the P parameter with the correct pack serial number.

In addition, FORMAT must be called from one of the following.

- A system origin job (from the console).
- A system privileged job (in this case, engineering mode must have been selected on the system console).

Access for Read and Write Operations

In addition to the information specified for read only operations, the following additional steps must be taken to access a device when write operations (ALTER and RESTORE) are to be performed.

1. The pack to be accessed should be mounted on a removable disk device.

2. One of the following conditions is required.

   - The device should not be a shared device. (Refer to section 13, Multimainframe Operations.)

   - If the device is shared, a global unload should be set.

3. The device must be declared logically off. Use the OFF DSD command or the OFF parameter in the EQPDECK EQ entry.

4. The disk status display (E,M) must show that the device is not in use.

5. One of the following conditions is required.

   - The disk status display (E,M) must show that the device is unavailable for permanent file access.

   - The following conditions are required.

     - The full initialize status and the format pending status must be set. Use the DSD command, INITIALIZE,FP,est.

     - The direct access file user count should be equal to zero. The family status display (E,F) gives this information.

If all the necessary conditions are satisfied, FORMAT is able to access the pack for reformatting purposes. FORMAT repeatedly checks to ensure that these conditions are satisfied throughout the FORMAT operation.

At the end of the FORMAT process the disk must be initialized (INITIALIZE,AL,est) to copy the flaw information to the TRT and label. Otherwise, the flaw map changes will have no effect on NOS until the next time the disk is recovered. At that time, unreserved tracks marked as flaws in the flaw map will be flawed in the appropriate NOS tables.

NOS/VE Address Formats H

When doing an analysis of a NOS/VE dump tape, you can specify NOS/VE addresses to the DSDI utility in any of several formats. These formats allow a flexible specification of the address range in which you are interested.

Address Format	Description
nn.n	A real memory address (RMA).
asid#nn..n	A system virtual address (SVA).
seg#nn..n#exch	A process virtual address (PVA) based on the exchange package that contains the address.
reg#exch	The register and exchange package that contains the address.
nn..n#reg#exch	A PVA formed from the segment field of the specified register pointed to by the specified exchange package. In this format nn..n is the byte offset from the beginning of the segment.
PVA#exch	The address in the pseudo register PVA of the specified exchange package.
exch	The address as the beginning of the specified exchange package.

The number sign (#) character is used in address formats to separate parts of the address parameter. Commas cannot be used because they delimit parameters within DSDI directives.

The symbols used for the various parts of the address formats are described as follows:

Symbol	Description
nn..n	A number (octal, decimal, or hexadecimal as required by the directive) of from 1 to 8 digits.
seg	A hexadecimal number of from 1 to 3 digits specifying a segment number.
asid	A hexadecimal number of from 1 to 4 digits specifying an actual segment identifier.

Symbol	Description
reg	Any one of the following registers:

reg	Description
P	Specifies the processor P register.
RA0 to RAF	Specifies the corresponding address register 0 to F hexadecimal.
RN1 to RN15	Specifies the top of stack address for the corresponding ring 1 to 15.
UTP	Specifies the address in the untranslatable pointer register.
TP	Specifies the address in the trap pointer register.

Symbol	Description
exch	Either a hexadecimal number of from 1 to 6 digits specifying an RMA of an exchange package (used to get the register values, segment table address, and segment table length needed to change the real address into a PVA), or one of the following keywords:

exch	Description
MPS	Address pointed to by the monitor process state register.
JPS	Address pointed to by the job process state register.
RMA	Real memory address calculated using the SETRMA DSDI directive.
PXP	Specifies that the current processor exchange package be used to obtain the P register address.

You can modify each address by adding or subtracting a hexadecimal number of bytes from the specified address. If the number added or subtracted is entered with no address, the last address specified is used as the base address.

The following illustrates some examples of valid address formats.

Example	Address Format
25E101	nn..n
MPS	exch
RA0#JPS−40	reg#exch
1000#RN1#MPS	nn..n#reg#exch
P#PXP+40	PVA#exch

Management Of Storage Media Defects — I

This appendix describes the process of detecting failing devices that are suspected of having storage media defects and the process of dealing with known storage media defects.

Detecting Failing Devices

This aspect of storage media management is the process of detecting failing mass storage devices and making informed decisions regarding the proper course of action. The action may be to place a failing disk in idle mode, change its threshold values, or set its device state to OFF or DOWN. These actions are discussed in the following paragraphs.

When a mass storage device or channel begins to fail, the system directs new file assignments away from the device and inhibits access to files on the device that are being used by executing jobs.

The operator is alerted to the device failure through the A,OPERATOR display. Also the E,E and E,H displays provide an overview of the mass storage devices that have abnormal conditions present. The E,E display shows that a device error has occurred on a particular device and the current state of that device. The E,H display shows the current mass storage threshold values and the current count associated with each threshold value. Refer to the NOS Version 2 Operations Handbook for additional information about the E,E and E,H displays.

You can use the IDLE command in conjunction with the THRESHOLD command to maintain some degree of flexibility in deciding when to repair the failing device. For more information, refer to the IDLE and THRESHOLD commands in section 5, DSD Commands.

NOTE

The decision to continue using a failing device should be made under the advisement of a customer engineer. If there is a chance of data corruption due to the nature of the failure, it may be best to leave the device OFF or DOWN until it can be repaired.

Disk Idle Mode

When a disk is in idle mode, the system temporarily eliminates access to the device when it starts to fail. Nonsubsystem jobs accessing the device will be rolled out so that the rest of the system and jobs not accessing the device can continue running. A verification algorithm will be used to attempt to isolate the problem to a particular part of the hardware (media, drive, controller, or channel). If the cause of the problem can be isolated to a failing channel or drive, then the system will attempt to DOWN that particular hardware resource.

Mass Storage Thresholds

The initial threshold values for a mass storage device are determined by THRESHOLD EQPDECK entries in the deadstart file. You can use the DSD THRESHOLD command to change the following threshold values for any mass storage device:

- Verification failure threshold
- Restricted activity threshold
- Low space threshold
- Recovered error threshold
- Unrecovered error threshold

These threshold values are described under the DSD THRESHOLD command in section 5.

When the space available on a mass storage device falls below the low space threshold, the following message appears on the A,OPERATOR display:

```
LOW SPACE ON MASS STORAGE DEVICE.
```

Dealing With Known Media Defects

This aspect of storage media management is the process of dealing with known media defects on mass storage devices. This process includes keeping an accurate record of all known defects and preventing the system from using defective areas. You should follow these general steps when you or the system encounter a media defect:

1. Record the media defect in a logbook.
2. Use the FLAW utility to reserve (flaw) the defective area and prevent the system from using that space.
3. Modify or create an APRDECK for the device containing the defect to include a command to flaw the defective area.
4. Modify the EQPDECK entry for the device to include a reference to the APRDECK entry.

This appendix describes general procedures for storage media management and the software tools you will need to perform these procedures. Later in this appendix an example shows the specific steps you should perform given a defined media defect problem.

Keeping An Accurate Record of Media Defects

Your site should maintain a logbook that documents all of the known mass storage media defects. This logbook should be kept by the system console so that an operator can record any media defect messages displayed. The information can then be used to reserve defective areas and thus prevent the system from using them.

Preventing The System From Using Flawed Disk Space

NOS allocates space on a mass storage device in units of logical tracks. For example, when you write a file to disk, NOS assigns a logical track to the file and then writes the file information to that logical track. The logical track remains assigned to the file until you release the file. That is, as long as the logical track is assigned to your file, it cannot be assigned to another file.

To prevent NOS from assigning a defective logical track to a file, you must reserve the track. This is called flawing a track. However, you cannot flaw a track while it is in use; that is, if the logical track with the defect is currently assigned to a file, you cannot flaw the track until the file is released.

If a media defect is encountered on a logical track assigned to a type of file that will not be released (such as, the system file, dayfiles, permanent file catalogs, and system checkpoint files) you will have to perform special procedures to explicitly move these files and then flaw the track. For example, you may need to dump and reload permanent files, reinitialize a disk, or deadstart the system. For a discussion of dealing with media defects encountered on areas assigned to these files, refer to Releasing Special Files and Reinitializing a Device later in this appendix.

Media Management Tools

You can use the following software tools to help with storage media management:

- Error log messages for 887, 895, and 9853 disks
- FLAW utility
- APRDECK entries
- EQPDECK entries
- FORMAT utility for 881/883 disk packs

Error Log Messages for 887, 895, and 9853 Disks

NOS automatically detects the presence of a media defect on an 887, 895, or 9853 disk. NOS automatically flaws the logical track that contains the media defect when the file containing the logical track is released.

When NOS encounters a media defect on an 887, 895, or 9853 disk, it issues a SEE A,ERROR LOG message to the operator. The operator should then use the A,ERROR LOG command to view the error log screen. The following message describes the media defect encountered:

```
dtest,TKnnnn,MEDIA DEFECT ENCOUNTERED
```

where dt is the device type, est is the EST ordinal of the device, and nnnn is the number of the logical track containing the media defect.

The operator should copy this message to a logbook for future use. This message indicates that NOS has detected a media defect; it does not indicate that the defective area has been automatically flawed. When the file assigned to the logical track containing the defect is released, NOS will then automatically flaw the track.

However, because media defects are rare (they should occur only once per device every two years), NOS stores only one media defect; the last defect encountered. Thus, your operator should always log the media defects encountered so that the information will not be lost if multiple defects occur.

When NOS automatically flaws a logical track, it again issues a SEE A,ERROR LOG message to the operator and displays this message on the error log screen:

```
dtest,TKnnnn,MEDIA DEFECT FLAWED
```

The operator should copy this message to a logbook for future use.

FLAW Utility

If a media defect occurs during normal system operation, use the FLAW utility to reserve (flaw) the defective track. (You can also use the FLAW utility to clear any flaws previously set.) To flaw a defective logical track, you must know the EST ordinal of the device containing the defect and the logical track number of the defect. For more information, refer to FLAW K Display in section 8, K-Display Utilities.

APRDECK Entries

In addition to using the FLAW utility to flaw a defective area of mass storage, you should also modify the APRDECK (auxiliary mass storage parameter deck) for the device containing the defect.

An APRDECK is a text record on the deadstart file that is used when the device is initialized. APRDECK entries identify areas of mass storage that are unusable (flawed) and prevent the system from accessing them. The system uses the information in the APRDECK entries to build the track reservation table (TRT) that resides in central memory resident (CMR) and also in the mass storage device label for each device.

By making APRDECK entries, you cause the system to flaw the media defects without your intervention during the next deadstart initialization of the device. For more information, refer to APRDECK in section 3, Deadstart Decks.

EQPDECK Entries

An EQPDECK (equipment deck) is a text record on the deadstart file that declares all the devices that comprise your computer system. Each device is defined to the system with an EQPDECK entry. An optional parameter on the EQPDECK entry declares an APRDECK for the device. When you create an APRDECK entry for a device, modify the EQPDECK entry for the device to include the reference to the APRDECK entry. For more information, refer to EQPDECK in section 3, Deadstart Decks.

FORMAT Utility for 881/883 Disk Packs

When you use the FLAW utility or an APRDECK entry to flaw an area of an 881 or 883 disk pack, this flawing is only logical. That is, the flaw information is kept in the TRT and is discarded on subsequent deadstart initializations of the device. This information is also lost during online initializations if it was not possible to recover the equipment.

For 881 and 883 disk packs, which contain factory-recorded utility maps, you have the option of storing flaw information on the utility map on the disk. To modify or add information to the utility map, use the FORMAT utility. For more information, refer to appendix G.

Media Defect Example

Suppose a job encounters a media defect on a local file. The file is located on an 895 disk device with EST ordinal 15 and the bad track is 5010. Use the following procedure to record the media defect and flaw the defective area.

1. When the system displays the message SEE *A,OPERATOR* use the A,OPERATOR command to view the operator screen. This message is displayed:

   ```
   13 SEE A,ERROR LOG.
   ```

2. To view the error log, enter this command:

   ```
   A,ERROR LOG.
   ```

   The following message is displayed on the error log screen:

   ```
   EQ015,TK5010,MEDIA DEFECT ENCOUNTERED.
   ```

3. Record the information in the media defect logbook.

4. Enter this command to tell the system you are taking care of the problem:

   ```
   LOG,13.
   ```

5. When the job using the local file releases the file, the following message is displayed on the error log screen:

   ```
   EQ015,TK5010,MEDIA DEFECT FLAWED.
   ```

6. Record the information in the media defect logbook.

7. Suppose there is no APRDECK for the device with EST ordinal 15. Add the AP=nn parameter to the EQPDECK entry for device 15. Here is an example of what the EQPDECK entry should look like:

   ```
   EQ015=type,ST=ON,CH=ch,AP=00.
   ```

   This entry assigns the first APRDECK to device 15.

8. Create the APRDECK for device 15:

   ```
   APRD00.
   SLF=5010.
   ```

9. Use the LIBEDIT command to place the APRDECK entry and the updated EQPDECK entry on the deadstart file. (Remember that the number of the APRDECK record is determined by its position in relation to the APRINST record and not by the number on the record name; it is a good idea to name the records according to their position.)

Releasing Special Files

Because a media defect cannot be flawed automatically by the system until the file assigned to the defective track is released, you may have to release certain file types:

File	Description
Direct access file	When a flaw is detected on a track assigned to a direct access file, purge the file so that the system can perform automatic track flawing for the bad track.
Dayfile	Job dayfiles are released at job termination; system dayfiles are released when they are initialized or if you perform a DFTERM and then purge the files created by DFTERM.
System file	The system file is released at deadstart. Thus, at deadstart a defective track assigned to the system file will be flawed automatically.

Reinitializing A Device

Some track chains are not released unless the device is reinitialized:

- Indirect access file chain.
- Disk deadstart file.
- Catalog track chain.

If a storage media defect is encountered in one of these files, perform the following steps:

1. Create or update an APRDECK for the device.
2. Modify the EQPDECK entry for the device to include the APRDECK reference.
3. Dump the files on the device to tape.
4. Reinitialize the device.
5. Reload the files to the device.

Display Disk File (DDF) Utility J

The Display Disk File (DDF) utility provides the capability to read, display, change, and print the contents of physical disk sectors. It is intended to be used by analysts as an aid when working on system problems that involve data stored on mass storage devices.

CAUTION

Changing disk table contents should be done carefully and only by analysts who understand the effects of the changes. Unless extreme care is exercised, use of DDF can lead to system hangs or loss of permanent files.

Bringing Up DDF

The DSD display for the DDF utility is brought up on the system console by entering the following command:

```
X.DDF.
```

NOTE

Since DDF allows access to all disk space without regard for security access levels and categories, use of DDF in a secured system requires that the system be in SECURITY-UNLOCK state (refer to Secured System Control Commands in section 5, DSD Commands).

DDF Display Left Screen

The left screen displays the following disk and control point information:

- Disk EST ordinal and equipment type.
- Current disk track and sector address.
- Central memory address and byte number of the TRT link byte for the current track. The link byte is intensified if it is not consistent with the sector linkage bytes.
- TRT reservation, interlock, and preserved file status for the current track.
- Family name, pack name, control point number, and JSN of the associated job.
- Track numbers for the first track of the permit chain and the first track of the indirect access file data chain for the specified family/pack.
- First and second control point messages, if issued. The second control point message is usually a disk error and is intensified.
- Contents of DDF scratch areas, labeled A through F (refer to the DDF STORE command).
- The recovery track pointer and track recovery word, when pack recovery is in progress.
- Various status and errors may be displayed at the bottom of the screen.

DDF Display Right Screen

The right screen displays the contents of the specified disk sector. The byte numbers displayed in parenthesis after the central memory address are the actual byte numbers for the physical disk sector. The byte numbers are used in the memory changing commands.

You can change the right screen format with the right blank key on the CC545 console or the tab key on the CC634B console.

The following right screen displays are available:

- Five Bytes Per Line. The contents of the sector are displayed in 5 bytes per line. The CC545 console displays one half sector per page and uses the / to toggle between the first and second half of the sector. The CC598B and CC634B consoles display one quarter of a sector per page and uses the / to increment the addresses.
- Ten Bytes Per Line. The contents of the sector are displayed in 10_8 bytes per line with byte numbers. The CC545 console displays the entire sector on one page. The CC598B and CC634B consoles display one half sector per page and uses the / to toggle between the first and second half of the sector.
- Text Mode. The contents of the sector are displayed in text mode. The CC545 console displays the entire sector on one page. The CC598B and CC634B consoles display one half sector per page and uses the / to toggle between the first and second half of the sector.
- Interpreted mode. The data is formatted in one of three ways depending on the linkage bytes:
  - System Sector. The file name table (FNT) word and file status table (FST) word of the file are displayed along with the date and time when the sector was last updated. If the file type is PMFT, the copy of the permanent file catalog entry (PFC) contained in the system sector is also displayed.
  - PFC Entry or Data Sector. The contents of one permanent file catalog entry (20_8 words) are displayed. Some fields such as dates are decoded. The / is used to advance to the next PFC entry in the sector.
  - End of Information Sector. This display shows a short format of the system sector display.
- Legal Commands. This display shows a list of the DDF commands.

Keyboard Input

You can use the following keys to interact with the DDF display.

Key	Action Initiated
+	Read the next sector. If positioned at EOI or the end of the track, the current sector is reread.
-	Read the previous sector. If positioned at the beginning of the track, the current sector is reread.
.	Advance to the track specified in the control bytes and set the sector to 0. If the control bytes do not contain a track link, the current sector is reread.
,	Advance to the next track in the TRT chain. The sector number is not changed. If currently at the last track in the chain, the current sector is reread.
=	Read the next sector, continuing past EOI. If positioned at the end of an EOI track, the current sector is reread.
CR or NEXT	Initiate processing of an entered command. This key also sets REPEAT ENTRY if a complete command has not yet been entered.
Space bar	Read the current sector into the buffer. The current sector is the sector to which the current EST ordinal, track, and sector values point.
*	Toggle between DSD and DDF.
8	Increment the track number by one and read the sector.
9	Decrement the track number by one and read the sector.
Right blank, ⟶| , or Tab	Change the right screen displays on the CC545 console. This function is done by the Tab key on the CC598B console and the ⟶| (tab) key on the CC634B console.
/	Toggle or increment the sector displays on the right screen.
BKSP, ⟵ , or Back Space	Delete the previous character typed on the CC545 console. This function is done by the Back Space key on the CC598B console and by the back arrow ⟵ (backspace) key on the CC634B console.
Left blank, |⟵ , or Esc	Delete the current line being entered on the CC545 console. This function is done by the Esc key on the CC598B console and the |⟵ (back tab) key on the CC634B console.

Console Messages

The following messages may appear at the control point.

Message	Description
WAITING FOR MEMORY	DDF is waiting for central memory to store the display and command processors.
READING EQxxx TPxxxx CTxxxx	This message shows the status of checkout during a pack recovery operation.

The following messages may appear above the command line at the bottom of the left screen.

Message	Description
FORMAT ERROR.	A format error has been detected during translation of the entry.
INCORRECT ENTRY.	The command is not valid.
INCORRECT EQUIPMENT.	EST is not mass storage, nor is it a null equipment.
INCORRECT PARAMETER.	The parameter in the entry is invalid or too long.
REPEAT ENTRY.	The entry will not be cleared after execution.
SYSTEM BUSY.	DDF is waiting for the system to process a request.
BYTE xxxx.	The data specified is in byte number xxxx.
LENGTH = nnnn.	The SKIPEI command has skipped nnnn octal sectors.

Some commands read several sectors. If a disk error is encountered while processing one of these commands, processing stops and an error code appears on the left screen.

DDF Commands

All DDF commands are displayed on the bottom of the left screen as they are entered. When you enter a command, it must be followed by a carriage return before any action is taken.

DDF commands are processed interpretively. After the period (.), any alphanumeric characters can be entered: however, the message FORMAT ERROR or INCORRECT PARAMETER will appear (after the carriage return is entered) if you make an error.

Most DDF commands do not actually cause a sector to be read and displayed. After entering a command that changes the EST ordinal, track, or sector, you must enter a SPACE key to read the sector. This allows data from one sector to be read and then written to another sector.

After entering a command that changes the equipment or track, the track and sector numbers are checked. If either the track or sector number is invalid, it will be set to zero.

The FIND commands scan the catalog track starting at the current position and search for the specified catalog entry. The EST ordinal and track number of the catalog track must first be set using the appropriate commands (FAMILY, PACKNAM, UI, etc.). The first sector of the catalog track must also be read, by entering the space bar, before the FIND command is used.

Command	Description
AUTOREAD.nnnn.	Read the sector every nnnn seconds. If nnnn is not entered, 1 is used. AUTOREAD is terminated by clearing the command entry with the left blank key or back tab key.
BLDEOI.	Creates an EOI sector in the buffer.
BOT.nnnn.	Back up one track. Search the TRT starting at track number nnnn for a track that points to the current track. If one is found, the current track is set to that value. If no track is found that points to the current track, the message INCORRECT PARAMETER is displayed. If nnnn is not entered, the search starts at track 0.
CTB.[1]	Clear the track interlock bit for the current track and equipment.
DEP.	Disable error processing for calls to the mass storage driver to read a sector. (By default, error processing is disabled.)
DIS.	Drop the DDF display and call DIS to the control point.
DROP.	Drop the DDF display and PP.
DTK.[1]	Drop tracks to the end of the chain starting with the current track.

1. The keyboard must be unlocked to use this command (refer to the UNLOCK command in section 5, DSD Commands).

Command	Description
DTK.ssss. [2]	Drop tracks starting with the current track and set the EOI sector in the TRT to the value ssss.
EEP.	Enable error processing for calls to the mass storage driver to read a sector. (By default, error processing is disabled.) If a disk error is encountered when reading or writing a sector with error processing enabled, the screens will not be refreshed until the driver has finished performing error processing. This may result in the screens being blank for many seconds.
EJT.ejt.	Enter the disk address from EJT ordinal ejt (0 ≤ ejt ≤ largest EJT ordinal). The equipment, track, and sector are set to the beginning of the file. The message INCORRECT PARAMETER is displayed if the EJT entry is not used or the file does not have any tracks assigned.
EST.est.	Enter the EST ordinal est. The equipment must be a mass storage device.
FAMILY.familyname.	Use permanent file family familyname. The family is set internally in DDF. PFCW in the control point area is not changed. The family is initially set to the family name specified in PFCW.
FIND.pfn.userindex.	Search for permanent file pfn with user index userindex.
FIND.pfn..	Search for permanent file pfn with a zero user index (i.e., a hole).
FIND.pfn.	Search for permanent file pfn with any nonzero user index.
FIND..userindex.	Search for any permanent file with user index userindex. The file name is ignored.
FIND...	Search for any permanent file with a zero user index. The file name is ignored (i.e., search for any hole).
FINDISS.userindex.	Starting at the current position, search for the next system sector with user index userindex on the chain. If userindex is not specified, the user index is not checked and the next system sector is displayed. This command is intended to be used on the indirect access permanent file chain. Set the EST ordinal, track, and sector to point to a position within the indirect chain before entering this command.

2. The keyboard must be unlocked to use this command (refer to the UNLOCK command in section 5, DSD Commands).

Command	Description
FINDO.octalnum.	Search from the current position for the octal number specified. The number can be 1 to 20 digits and is right justified in as many bytes as necessary to hold the number. The search starts on a byte boundary and the byte number is displayed if the search is successful. If the first part of the number is found at the end of the buffer, the search terminates even though the entire number is not found. If the carriage return is entered again, the search continues at the location of the first match.
FINDS.string.	Search from the current position for the specified string of characters. If the first part of the string matches the end of the buffer, the search terminates even though the entire string is not found. The byte number of the beginning of the string is displayed. If the carriage return is entered again, the search continues at that point.
FINDSS.userindex.	Starting from the current track and searching to the end of the TRT, find and display the next system sector with user index userindex. If userindex is not specified, the user index is not checked.
FNT.fnt.	Enter disk information from the system FNT ordinal fnt ($0 \leq$ fnt $\leq$ largest FNT ordinal). The EST ordinal, track, and sector are set to the beginning of the file. The message INCORRECT PARAMETER is displayed if the FNT entry is not used or the file does not have any tracks assigned.
FNTL.fnt.	Enter the disk address from the local FNT ordinal fnt ($0 \leq$ fnt $\leq$ largest FNT ordinal). The EST ordinal, track, and sector are set to the beginning of the file. The message INCORRECT PARAMETER is displayed if the FNT entry is not used or the file does not have any tracks assigned.
FNTLC.fnt.	Enter the disk address from the local FNT ordinal fnt ($0 \leq$ fnt $\leq$ largest FNT ordinal). The EST ordinal, track, and sector are set to the current position of the file. The message INCORRECT PARAMETER is displayed if the FNT entry is not used or the file does not have any tracks assigned.
GETTRT.nnnn.	This command is intended to be used on the label track. The EST ordinal and track must be set for the label track before the command is used. GETTRT reads the sector that contains the checkpoint TRT information for track number nnnn and displays the byte number within the sector for that track.
HOLD.	Release the display and wait for the operator to reassign the display.
LOAD.c.	Reload the EST ordinal, track, sector, and display selection from scratch area c. The scratch area is specified by a single alphabetic character. Valid characters are A through F. (Refer to the STORE command.)

Command	Description
PACKNAM.packname.	Use permanent file pack packname. The pack name is set internally in DDF. PKNW in the control point area is not changed. The pack name is initially set to the name specified in PKNW.
PREAD.[3]	Read the current sector using the read protected sector function. This command is used instead of the space bar when reading protected sectors.
PTK.nnnn.[3]	Enter the protected track number nnnn. This command works the same as the TK command except the track number entered is not checked.
PWRITE.[3]	Write the current sector using the write protected sector function. This command should be used instead of the WRITE command when writing protected sectors.
QFT.qft.	Enter the disk address from QFT ordinal qft ($0 \leq$ qft $\leq$ largest QFT ordinal). The EST ordinal, track, and sector are set to the beginning of the file. The message INCORRECT PARAMETER is displayed if the QFT entry is not used or the file does not have any tracks assigned.
RANDOM.nnnn.addr.	Set the track and sector for random address addr using number nnnn as the first track. The current track is used as the first track if nnnn is not specified. The message INCORRECT PARAMETER is displayed if the random address is not on the chain.
RANDOM..addr.	Set the track and sector for random address addr using the current track as the first track. The message INCORRECT PARAMETER is displayed if the random address is not on the chain.
RANDOM.c.addr.	Set the track and sector for random address addr using the track specified by scratch area c as the first track. The scratch area is specified by a single alphabetic character. Valid characters are A through F. (Refer to the STORE command.) The message INCORRECT PARAMETER is displayed if the random address is not on the chain.
SC.nnnn.	Enter the sector number nnnn. The message INCORRECT PARAMETER is displayed if the sector number is too large.
SC.*.	Enter the sector number of the last sector on the track.
SCAN.	Scan from the current position until the EOI control bytes are encountered. Scanning stops if, at some point, the control bytes are incorrect or the track is not reserved.

3. The keyboard must be unlocked to use this command (refer to the UNLOCK command in section 5, DSD Commands).

Command	Description
SCAN.*.	Scan from the current position until the end of information indicated in the TRT is reached. This is intended to be used for the indirect access permanent file chain. Note that when PFM delinks a track in the middle of the chain, it does not update the track pointer in the preceding track; which causes SCAN to stop at this point and display an error.
SHOWPF.	Display the permanent file whose catalog entry is currently displayed. The PFC display must be on the right screen to use this command. The device, track, and sector from the PFC currently displayed are used to display the file. If the device is not present, the message INCORRECT PARAMETER is displayed.
SKIPEI.	Set the track and sector to EOI based on the current position and information in the TRT. The number of sectors skipped is displayed on the left screen.
SKIPF.	Read the file starting at the current position until an EOF is encountered.
SKIPR.	Read the file starting at the current position until an EOR or EOF is encountered.
STB.[4]	Set the track interlock bit for the current track.
STORE.c.comment	Store the current EST ordinal, track, sector, and display selection into scratch area c. The scratch area is specified by a single alphabetic character (valid characters are A through F). The comment is copied to the scratch area and displayed on the left screen. The comment is for convenience only and is truncated after 10 characters. The EST ordinal, track, sector, and display selection can be reloaded from the scratch area using the LOAD command.
TK.nnnn.	Enter the track number nnnn. The message INCORRECT PARAMETER is displayed if the track number is too large.
UI.userindex.	Set the EST ordinal and track for the catalog entries for user index userindex (userindex ≤ 377777). The message INCORRECT PARAMETER is displayed if the catalog is not found (this may be the result of entering the wrong family name or pack name). If either the family name or pack name is changed after entering the UI command, the UI command must be reentered.
WRITE.[4]	Write the contents of the buffer to the sector currently displayed.

4. The keyboard must be unlocked to use this command (refer to the UNLOCK command in section 5, DSD Commands).

Print Sector Data Commands

The following commands are used to print the contents of the disk sector and manipulate the listing file. These commands use a CPU resident helper program called DDFILE. If DDF is unable to load DDFILE (due to system activity or other constraints), the use of these commands will be disabled.

Command	Description
LISTING.filename.	Set the listing file name to filename. The default listing file name is OUTPUT.
OUT.	Release the listing file to the output queue.
PRINT.	Print the current equipment type, track, sector, TRT information, and the contents of the sector.
PRINT.num.	Print the next num (octal) sectors starting with the current sector. Printing stops if EOI is encountered before num sectors have been printed. One page of output is produced for each sector printed.
RETURN.	Return the listing file.
REWIND.	Rewind the listing file.
SETID.id.	Set the identifier for the listing file to id.
SKIPL.	Advance from the current position of the listing file to the end of the file.

Change Sector Data Commands

The following commands are used to change the data in the current sector. If the comma (,) is replaced by a plus (+), the byte number xxxx is incremented after the entry is processed. The actual data on the disk is not changed until you enter the WRITE command.

Command	Description
xxxx,yyyy.	Enter value yyyy into byte number xxxx.
xxxx,Dcc.	Enter display code characters cc into byte number xxxx.

DDF Examples

The following examples illustrate DDF usage.

Example 1:

Read the system sector for permanent file ABC under user index 1234 on family XYZ.

`X.DDF.`	Bring up the DDF display.
`FAMILY.XYZ.`	
`UI.1234.`	Set the EST ordinal, track, and sector of the first catalog entry on this catalog track.
Space bar	Read the first catalog sector.
`FIND.ABC.1234.`	Search the catalog track for the PFC entry.
`SHOWPF.`	Set the EST ordinal, track, and sector of the file.
Space bar	Read the system sector of the file.

Example 2:

Look at the PFC entries under user index 1234 on family XYZ.

`X.DDF.`	Bring up the DDF display.
`FAMILY.XYZ.`	
`UI.1234.`	Set the EST ordinal, track, and sector of the first catalog entry on this catalog track.
Space bar	Read the first catalog sector.
`FIND..1234.`	Search for the first file under the user index 1234.

Example 3:

While using DIS, look at the contents of a local file. Be sure to remember the FNT ordinal fnt of the file.

`DDF.`	Switch control from DIS to DDF.
`FNTL.fnt.`	Set the EST ordinal, track, and sector to the system sector of the file.
or	
`FNTLC.fnt.`	Set the EST ordinal, track, and sector to the current position of the file.
Space bar	Read the system sector of the file (if FNTLC was entered, read the sector to which the file is currently positioned).
`Use +, -, and /`	Look at different parts of the file.
`DIS.`	Switch control from DDF to DIS.

Example 4:

Display a system file (e.g., the VALIDUS file). Look at the DSD H display to determine the FNT ordinal fnt.

X.DDF.	Bring up the DDF display.
FNT.fnt.	Set the EST ordinal, track, and sector to the system sector of the file.
Space bar	Read the system sector of the file.
Use +, -, and /	Look at different parts of the file.

Example 5:

Determine the length of a file and display the EOI sector. Display the system sector of the file (see previous examples).

SKIPEI.	Set the track and sector to the file EOI and display the number of sectors skipped above the command.
Left blank	Clear the message and command.
Space bar	Read the EOI sector.

Example 6:

Verify that a file can be read and that the linkage bytes are correct. Display the first sector of the file (see previous examples).

SCAN.	Read every sector from the current position in the file to the EOI. If any errors are encountered, SCAN stops and displays a message. SCAN verifies that the track linkage bytes match the TRT.

Example 7:

Print the contents of sectors. Display the first sector (see previous examples).

PRINT.num.	Print the next num sectors (num must be an octal number) to a listing file. The default for num is 1. The default for the listing file is OUTPUT unless the LISTING command has been used to specify another file name.
OUT.	Put the listing file in the output queue to be printed.
DIS.	File OUTPUT contains the listing file of sectors.

Example 8:

Often it is necessary to look at several different places on the disk. The DDF utility has a facility to remember disk locations.

X.DDF.	Bring up the DDF display.
FAMILY.ABC.	
UI.113.	Set the EST ordinal, track, and sector of the first sector of the catalog track.
Space bar	Read the first sector of the catalog track.
FIND.XYZ.113	Find the PFC entry for the file XYZ on user index 113.
STORE.A.XYZ PFC	Save the current EST ordinal, track and sector in scratch area A. Save the characters XYZ PFC as a comment. This information is displayed on the left screen.
SHOWPF.	Set the EST ordinal, track, and sector for the system sector of the file.
Space bar	Read the system sector of the file.
STORE.B.XYZ SS	Save the current EST ordinal, track, and sector in scratch area B. Save the characters XYZ SS as a comment.
SKIPEI.	Skip to the EOI sector of the file.
Left blank	
Space bar	Read the EOI sector of the file.
STORE.C.XYZ EOI	Save the current EST ordinal, track, and sector in scratch area C. Save the characters XYZ EOI as a comment.
LOAD.A.	Reset DDF to display the PFC entry.
Space bar	Read the PFC sector.
LOAD.B.	Reset DDF to display the system sector.
Space bar	Read the system sector.
LOAD.C.	Reset DDF to display the EOI sector.
Space bar	Read the EOI sector.

Example 9:

Change data on the disk. Display the sector to be changed. Determine from the display the byte number of the byte to be changed.

CAUTION

Be very careful and be sure you know what you are doing before changing data on the disk. Unless extreme care is exercised, use of DDF to change data on the disk can lead to system hangs or loss of permanent files.

---

*	Return control to DSD.
UNLOCK.	Allow privileged commands to be entered.
*	Return control to DDF.
xxxx,yyyy.	Set byte number xxxx to value yyyy. If a mistake is made, press the space bar to reread the sector and start over.
WRITE.	Write what is displayed to the disk.

CAUTION

Be sure another program does not change the sector between the time DDF reads it and writes it; otherwise, the changes may be lost. Depending on what you are changing, it may be best to make changes only when the system is idle. For changes to permanent files and catalog entries, the STB and CTB commands can be used to prevent other accesses to the sector; however, an understanding of permanent file interlocking is advisable before you use these commands.

Example 10:

Display the disk flaw map. First, determine (from the appropriate hardware manual) the logical track and sector of the flaw map. This is not currently operational for buffered devices.

X.DDF.	Bring up the DDF display.
EST.est.	Set the EST ordinal est.
*	Return control to DSD.
UNLOCK.	Allow privileged commands to be entered.
*	Return control to DDF.
PTK.nnnn.	Set the protected track number nnnn. DDF does not verify that the track number is valid.
SC.nnnn.	Set the sector number nnnn.
PREAD.	Read the sector using a protected read function. (Do not attempt to read the sector using the space bar; this would cause a disk error because this sector is protected.)

Pack Recovery

In the event that the label sector and TRT on a device have been overwritten and no suitable backup is available, a special set of DDF commands are available to allow the reconstruction of these tables. Note that this process requires extensive analyst intervention (on the order of several hours of dedicated time).

To effectively use these commands, an analyst must have a thorough and detailed understanding of the NOS mass storage and permanent file table structures, both in CMR and on disk. Recovery achieved by these commands is not automatic; the analyst will have to make manual changes to the label sector and TRT during the process of the reconstruction.

Refer to Pack Recovery Hints later in this section for an outline of the procedure to be used.

Pack Recovery Display Changes

When pack recovery is in progress, the recovery track pointer and the track recovery word are displayed on the left screen.

Recovery Table Format

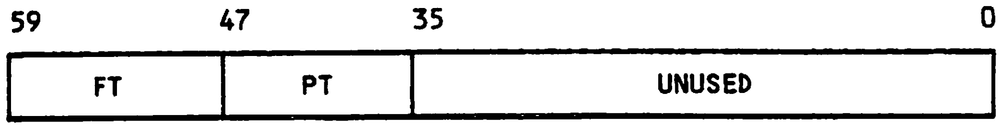
The pack recovery process uses a recovery table in central memory to record linkage information. The table is initialized by the BEGINR command. The other commands use or modify the data in the table to effect device recovery. The table (which is stored within DDF's central memory field length) is formatted as follows:

- Image of MST for the device (MSTL words long).
- FNSS/FTSS/DTSS words from the system sector of the current chain.
- Recovery pointer word.

59	47	35	23 0
FT	PT	SC	UNUSED

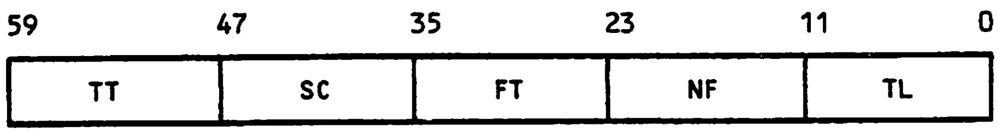
Field	Description
FT	First track of the current chain.
PT	Previous track in the chain.
SC	Current sector for the EOI search after an error.

- Indirect chain pointer word (not currently used).



Field	Description
FT	First track of the indirect chain.
PT	Previous track of the indirect chain.

- Build pointer word.



Field	Description
TT	Destination track for writing the TRT.
SC	Sector for the current piece of the TRT.
FT	First track for the current piece of the TRT.
NF	DAF file count.
TL	TRT length from the MST.

- Two reserved words.

- Track recovery table (one word per track).

59	47	35	23	11 0
FLAGS	FT	PT	NT	LC

Field	Description
FLAGS	The following flags are valid:

Flag	Bits	Description
TY	59-54	File type from system sector (for tracks beginning with a system sector).
RE	53	Read error detected in track..
SS	52	System sector in track at other than sector zero.
FL	51	Flawed track indicator.
	50	Not used.
EI	49-48	EOI status for track (0, 1, or 2). 0 No EOI. 1 Normal EOI in track. 2 EOI (written by IMS).

Field	Description
FT	First track of the chain linked to this track.
PT	First track encountered that is linked to this track. (PT equals the current track if at the start of the file.)
NT	Next track or EOI sector number.
LC	Linkage count. (Number of tracks linked to this track.)

Pack Recovery Commands

The BEGINR command must be entered as the first command of a pack recovery sequence; until the BEGINR command is entered, none of the other pack recovery commands will be accepted. All of these commands require that the keyboard be unlocked.

Command	Description
BEGINR.	Begin track recovery for the current selected equipment. The equipment to be recovered must be removable and unloaded. The MST in CMR for the selected equipment is used to initialize the MST for the label track built by the recovery commands. Be sure the equipment mnemonic in the E,M display matches that of the pack to be recovered. (The correct mnemonic can be set with the DSD INITIALIZE command.) The first use of this command initializes the recovery process by: • Setting the recovery track pointer to zero. • Reserving and clearing the central memory (CM) recovery table. • Setting up the MST skeleton in the recovery table. • Enabling the use of the other recovery commands. If the BEGINR command is used again, recovery restarts for the current equipment.
BLDSL.	Build a label sector image in the data buffer from data in the CM recovery table. Once the other recovery commands have completed, memory change commands can be used to fill in any missing data. The buffer can then be written to disk using the WRITE command.
BLDSTRT.	Build the next sector of the TRT data for the label track. The track and sector are set for the WRITE command. If any linkage errors are detected, the message ERROR IN TRACK STATUS is issued. After the last sector is built, the pointers are reset to the first sector again. You must manually generate the EOI sector for the label track using the BLDEOI command.
BLDSTRT.nnnn.	Build sector number nnnn of the TRT data for the label track (nnnn $\geq$ 1).

Command	Description
NEXTAT.	Locate the next available track on the current equipment and read the first sector of that track. The search begins at the displayed recovery track pointer +1. By resetting the track pointer to track zero and using this command, unreserved tracks can be found and checked for being part of the catalog track chain or the indirect access file chain.
NEXTSS.	Locate the next track in the recovery table that begins with a system sector. The search begins at the displayed recovery track pointer +1.
RECOVER.RECOVER.c.	Scan the entire device, saving linkage information in the track recovery table. The first sector of each track is read until a system sector is found. For non-SYFT system sectors and for SYFT files other than DATA, read each sector (following the sector linkage) until either an EOI sector, a multiple linkage, an embedded system sector, or a read error is encountered. For SYFT files LABEL, DATA, and PERMIT, set the ALGL track pointer in the MST skeleton. If the scan stops for any reason, it can be restarted by entering either the RECOVER or RECOVER.c command (where c is any character). Entering RECOVER.c resumes the linkage chain scanning from the sector after the one in which the scan stopped. Entering RECOVER causes DDF to go back to the last track on which a system sector was found, advance the track by one, and resume scanning the first sector of each track looking for another system sector. Once the RECOVER command has been used to scan the entire device (which may take several hours) the other commands can be used to resolve any linkage conflicts or other loose ends that remain. **NOTE** DDF releases the display to DSD while executing the RECOVER command to improve performance (minimize lost revolutions). The display is automatically rerequested when the scan stops for any reason.
SETRW.b.nnnn.	Change byte b of the current track recovery word to number nnnn. This is the method used to resolve linkage conflicts before using the BLDSL and BLDSTRT commands.
SETTP.nnnn.	Set the recovery track pointer for the RECOVER, NEXTAT, NEXTSS, and SETRW commands to track number nnnn.

Pack Recovery Hints

The general process for recovering a pack is as follows:

Example	Explanation
BEGINR.	
RECOVER.	Repeat as many times as required.
SETRW.	Repeat as necessary to correct linkage conflicts and oversights.
BLDSL.	
WRITE.	
BLDSTRT.	Repeat until TRT is complete.
WRITE.	Repeat until TRT is complete.
SC.nnnn.	Set EOI sector address.
BLDEOI.	
WRITE.	

There are, however, several manual operations required in addition to the process outlined above.

- Several fields in the MST image in the label sector must be entered manually after the BLDSL command and before the WRITE command. These include:
  - Family/pack name.
  - Device number.
  - Device masks.
  - Number of catalog tracks.
- The catalog tracks are not automatically recovered. The NEXTAT command can be used to search for the catalog tracks after the device has been completely scanned by the RECOVER command. In general, the catalog tracks are allocated on a device immediately prior to the first track of the indirect chain. (The first track of the indirect chain can be found by checking byte 0 of ALGL in the MST, built by the BLDSL command.) Once the catalog tracks are found, the track linkage information for these tracks must be set in the track recovery words by using the SETRW command.

  The label track must be linked to the first catalog track, which is linked to the second original catalog track, which is linked to the third original catalog track, and so forth. All overflow catalog tracks are linked after the last of the original catalog tracks.

  There can be one or more overflow catalog tracks for each of the original catalog tracks. These overflow tracks are linked from their corresponding catalog track via the sector linkage in the last sector of that track. This means that the sector linkage does not correspond to the TRT linkage for these tracks.

- The indirect chain is not automatically recovered. Tracks in the indirect chain can be located using the NEXTAT command and the RECOVER.c or FINDISS commands. Tracks for which bit 52 is set in the recovery word (which indicates that a system sector was found in a sector other than sector zero on this track) are probably part of the indirect chain. Examining the sector linkage of the last sector of each track that has been identified as containing indirect access file data can be used to establish the ordering of these tracks. Note that if the last sector of an indirect chain track is a system sector, byte NSSS contains the pointer to the next track. Alternatively, the PFCs on the catalog tracks can be examined to determine the tracks that are on the indirect chain and the ordering of these tracks.

- Instances in which multiple tracks are linked to a single track in the track recovery table must be resolved in order to correctly build the TRT. Note that the BLDSTRT command will inform you of these conflicts. You must inspect the contents of the various tracks and make an empirical judgment as to the correct linkage.

  The SETRW and SETTP commands must be used to eliminate the conflicting linkage. If a track is to be left unreserved, its track recovery word should be zeroed.

- Flawed tracks may be indicated by setting the flawed track indicator (FL flag) in the track recovery word for that track.

PACKER Utility K

The PACKER utility enables you to manage holes within the indirect access permanent file chain (IAPF chain) on a permanent file device. This chain is the set of tracks that contains all the indirect access permanent files on a permanent file device. Space within the chain is allocated to individual files.

Holes are the spaces created within the IAPF chain when indirect access permanent files are deleted. These spaces may be reused later for other indirect access permanent files; however, over time these spaces tend to become fragmented into smaller and smaller pieces. Eventually, a large amount of space may be in small, unusable holes.

The PACKER utility reorganizes the IAPF chain so that more of the holes are available for storage. It combines adjacent holes, moves files to allow holes to be combined, and releases entire tracks when possible. These actions make more tracks available on the device (relieving a TRACK LIMIT condition, if one exists) as well as making the remaining holes larger and thus more usable.

Operational Overview

A single call to the PACKER utility only processes a single permanent file device, but you can process multiple devices at the same time by calling a different copy of PACKER for each device.

NOTE

While PACKER is processing a device, no other files are allowed to access that device.

PACKER begins processing a device by reading all the permanent file catalog entries (PFC entries) on the specified device and building a set of tables that allows it to operate on the IAPF chain in sequential order. PACKER then scans the IAPF chain from the beginning, searching for holes.

Lost Space Processing

If PACKER encounters any lost space during its scan, it attempts to reclaim the space for future use. Lost space is a condition where space exists on the IAPF chain that is not pointed to by a permanent file catalog (PFC) entry. Without a PFC entry, the permanent file manager (PFM) is unaware of the space. Lost space is usually the result of a previous PACKER run being interrupted by a system failure.

If the lost space is preceded by or followed by a hole, PACKER merges the lost space with one of the holes. If the lost space is preceded by a file and not followed by a hole, PACKER creates a new PFC entry for it, transforming it into a normal IAPF hole. If the lost space is smaller than the absolute minimum IAPF hole size of three PRUs, it cannot be reclaimed and is left as lost space.

Hole Processing

When PACKER encounters a hole during its scan, a sequence of events begins. Unless the collection hole is already open, the newly found hole is turned into the collection hole. The collection hole is the focal point of the major operations of combining holes and moving files. The PFC pointing to the collection hole is changed into a purged direct access permanent file (DAPF) PFC. This PFC is changed back into an IAPF PFC pointing to the hole when the collection hole is changed back into a normal hole; until that occurs, the collection hole is lost space.

PACKER uses this procedure to provide for system failures during PACKER operation. If such a failure occurs, the system is unaware of the changes PACKER is making, all files are intact, and the lost space is recovered the next time PACKER is run.

The following sequence is repeated until the collection hole is closed.

1. Any holes following the collection hole have their PFCs changed into purged DAPF PFCs and their space is added to the collection hole. This combining of contiguous holes is continued until PACKER encounters a file.

2. When a file is encountered, PACKER checks to see if the collection hole is big enough for tracks to be delinked from it. If so, it delinks and returns the tracks to the system. This premove delinking limits the amount of data (files) that are moved by limiting the size of the collection hole.

3. The utility looks for the largest file that fits into the collection hole by starting from the end of the IAPF chain and scanning toward the current position. If it finds such a file, PACKER copies the file to the beginning of the collection hole, updates its PFC to point to the new position, and updates the collection hole's starting address and length. (This is called the fill-move technique.)

   If the collection hole is completely filled, PACKER considers it closed and begins scanning again. If the move is successful, PACKER repeats it until no further files can be moved using the fill-move technique.

4. The file adjacent to the collection hole (which was excluded from the fill-move search) is then examined to see if it can be slid across the collection hole. This file must completely fit within the collection hole, for the file would be destroyed if the file was copied on top of itself and the system failed during the copy.

   PACKER examines the sizes of all the files until the next hole to ensure that they fit in the collection hole (if any do not fit, the effect of the moves would only be to move the position of the hole, which in itself is of no benefit).

   If all the files fit, PACKER copies them, one by one, across the collection hole, and updates the address of the collection hole. (This is called the slide-move technique.) PACKER combines the newly adjacent hole with the collection hole and resumes scanning. If all the files between the collection hole and the next hole do not fit into the collection hole, PACKER closes the collection hole and scans for the next hole.

5. PACKER terminates the scan when it encounters the end of the IAPF chain. If the collection hole is open, PACKER drops it off the end of the IAPF chain by adjusting the EOI.

Dayfile Statistics

At termination, PACKER issues the following set of statistical dayfile messages. These messages document the original state of the device, changes made by PACKER, final state of the device, and resources used by PACKER.

```
ONSET FILES     NNNNNN,  SSSSSS PRUS.
ONSET HOLES     NNNNNN,  SSSSSS PRUS.
FILL MOVES      NNNNNN,  SSSSSS PRUS.
SLIDE MOVES     NNNNNN,  SSSSSS PRUS.
FILES MOVED     NNNNNN,  SSSSSS PRUS.
LOST SPACE      NNNNNN,  SSSSSS PRUS.
HOLES LEFT      NNNNNN,  SSSSSS PRUS.
HOLES FREED     NNNNNN,  SSSSSS PRUS.
TRACKS FREED    NNNNNN,  SSSSSS PRUS.
FLAW SPACE      NNNNNN,  SSSSSS PRUS.
SRUS                23456.890 UNTS.
CPU SECONDS         23456.890 SECS.
MAXIMUM CM FL       234567890  SCM.
MANAGED TABLE MOVES 234567890 UNTS.
```

Command Format

The PACKER command calls the utility that reorganizes the indirect access permanent files on a specified permanent file device. The command has the following format:

```
PACKER,p1=f1,p2=f2,...,pn=fn.
```

pi	Description
DN=devicenumber	Device number. This parameter is required if a family device is to be processed. It cannot be used with the PN parameter.
FM=familyname	Family name. This parameter cannot be used with the PN parameter. If you omit both the FM and PN parameters, PACKER processes the system default family.
NM	No moves. If you specify this parameter, PACKER does not move any files; it limits its work to the combining of contiguous holes and the reclamation of lost space. Since this parameter greatly decreases the time required to run PACKER, you can use it when attempting to relieve a TRACK LIMIT condition during production hours.
PN=packname	Pack name. This parameter is required if the device to be processed is an auxiliary pack. This parameter cannot be used with the FM or DN parameter.
TL=timelimit	Time limit. If you specify this parameter, PACKER stops running after the specified number of seconds (wall clock time). If a value is specified, the system assumes it is octal. If the TL parameter is specified but not equivalenced (that is, TL is specified without a value), the system uses a limit of 300D.

The following command parameters are not needed for normal operations, but may be useful in special situations.

pi	Description
EF	Exact fit. If you specify this parameter, an exact fit is required when moving a file with the fill-move technique. Use of this parameter might increase the number of files moved using the slide-move technique, which can result in more space being released to the operating system at the expense of moving an increased number of files.
IX=index	Index. This parameter specifies the starting index into PACKER's sorted PFC table. You can use it to skip the specified number of PFC entries before beginning processing. In some situations, this parameter provides a useful method of shortening the execution time of PACKER. If a value is specified, the system assumes it is octal. (The use of this parameter has an impact on the end-of-run statistics, since they do not reflect the portion skipped.)

pi	Description
MR=moveratio	Move ratio. This parameter specifies the maximum ratio between the total length of the files to be moved in a slide move and the size of the hole into which the files are to be moved. A value of 0 prevents all files from being moved using the slide-move technique; however, it is recommended that you use the NS parameter instead of MR=0. If a value is specified, the system assumes it is octal. If you omit the MR parameter, the system uses a value of 100B for the move ratio; if the MR parameter is specified, but not equivalenced, the system uses a value of 377777B.
NC	No changes. If you specify this parameter, PACKER determines what operations ought to be performed in response to the other specified parameters, but does not actually make any changes to the device. Since PACKER still issues dayfile messages at termination detailing the changes that would have been made, you may use this option to preview the potential effect of a parameter setting. You can compare alternative parameter combinations to determine the best way to reach a desired result.
NF	No fill moves. If you specify this parameter, PACKER does not move any files using the fill-move technique. Use of this parameter forces all files to be moved using the slide-move technique. This can result in more space being released to the operating system at the expense of moving an increased number of files.
NP	No premove delinks. If you specify this parameter, PACKER does not delink tracks before attempting to move files. Delinking tracks before moving files limits the size to which the collection hole is allowed to grow, which in turn limits the amount of data moved by PACKER in its attempts to fill the collection hole. These delinks might shorten the time PACKER executes at the expense of possibly leaving small holes where tracks were delinked. This parameter does not affect the delinking of tracks from holes that remain after PACKER has completed its work.
NS	No slide moves. If specified, PACKER does not move any files using the slide-move technique. This might increase the speed of PACKER execution by leaving more holes unfilled. You may also use the MR parameter to control slide moves in a more general fashion; however, to totally disable slide moves, you should use the NS parameter rather than specifying MR=0.

PACKER Examples

The following examples illustrate how you can use the PACKER utility to perform periodic maintenance on permanent file devices.

Example 1:

For normal periodic maintenance, the PACKER utility can be executed when the production load is light and there is little interactive usage. For example, a site might want to use the following command every day in the early morning:

```
PACKER(FM=ffffff,DN=nn)
```

Example 2:

To get some space on a device that is at or near track limit, PACKER can be run quickly even during active production hours by specifying the NM parameter. Specifying this parameter reduces the amount of space reclaimed, but also greatly reduces the amount of time required to run PACKER. The following command may be entered:

```
PACKER(FM=ffffff,DN=nn,NM)
```

Example 3:

If a site determines that daily PACKER runs have too great an impact on production, the site can specify the NM parameter (as in example 2) when running PACKER on weekdays, and only do full PACKER runs (as in example 1) once a week.

Index

Index

D

E

G

H

I

M

N

O

P

R

S

7

8

9

Comments (continued from other side)

Please fold on dotted line;
seal edges with tape only.

FOLD

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
First-Class Mail Permit No. 8241 Minneapolis, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CONTROL DATA
Technical Publications
ARH219
4201 N. Lexington Avenue
Arden Hills, MN 55126-9983

We would like your comments on this manual to help us improve it. Please take a few minutes to fill out this form.

Who are you?

- ☐ Manager
- ☐ Systems analyst or programmer
- ☐ Applications programmer
- ☐ Operator
- ☐ Other ______________________

How do you use this manual?

- ☐ As an overview
- ☐ To learn the product or system
- ☐ For comprehensive reference
- ☐ For quick look-up
- ☐ Other ______________________

What programming languages do you use? ______________________

How do you like this manual? Answer the questions that apply.

Yes	Somewhat	No	
☐	☐	☐	Does it tell you what you need to know about the topic?
☐	☐	☐	Is the technical information accurate?
☐	☐	☐	Is it easy to understand?
☐	☐	☐	Is the order of topics logical?
☐	☐	☐	Can you easily find what you want?
☐	☐	☐	Are there enough examples?
☐	☐	☐	Are the examples helpful? (☐ Too simple? ☐ Too complex?)
☐	☐	☐	Do the illustrations help you?
☐	☐	☐	Is the manual easy to read (print size, page layout, and so on)?
☐	☐	☐	Do you use this manual frequently?

Comments? If applicable, note page and paragraph. Use other side if needed.

Check here if you want a reply: ☐

Name ______________________ Company ______________________

Address ______________________ Date ______________________

Phone ______________________

Please send program listing and output if applicable to your comment.

Quick Reference Index