Jacquard Systems

System II Utilities

V2-005

SYSTEM II

UTILITIES

Publication V2-005-4

July 1979

REVISION HISTORY

Number:	Date:	Notes:
V2-005-0	12/78	First release.
V2-005-1	1/79	Revision.
V2-005-2	2/79	Revision.
V2-005-3	4/79	Replacement. Release 8.0 of System II.
V2-005-4	7/79	Revision. See page 0-3.

NOTICE

PREFACE

This publication describes a standard set of System II utility programs. A series of execution requests for these programs is the equivalent of a system command language. Also provided is an overview of certain System II concepts and terminology.

! REVISION DETAILS

The character "!" in the left margin of a chapter title or a section heading indicates substantially new material throughout. The same character in the left margin of a text line indicates a significant spot change. Corrections of minor typographical or stylistic errors on update pages are not indicated.

The pages in the latest revision are listed below. Each page is a replacement or an insertion by page number, unless a deletion is specifically indicated. Note that each page includes its own revision level at the end of the publication number. When a two-sided page is distributed, the text on one side may be unchanged.

Title, 0-2. Publication number updated.

0-3, 0-4 (blank). Section added.

TC-1, 2. EDIT write-up added.

2-1, 2-2 (blank). Reference to separate Editor manual deleted.

BLDCPY-3, 4. Corrections.

DISCON-1, 2 (blank). /B switch added.

DOCCRT-1, 2. Corrections; example added.

EDIT-1 to 14; 15, 16 (blank). New write-up, replacing publication V1-021.

FCMP-1, 2. Corrections; messages added.

FORMAT-1, 2. Corrections.

LSPY-1, 2. References to embosser deleted.

MONITR-1, 2. Corrections.

SIIDEF-1, 2. Response added.

UPDATE-1, 2. Corrections.

TABLE OF CONTENTS

ACUP	Asynch Characteristics Update
ALTER	Alter Contents of Memory
ANSWER	Auto-Answer Modem Monitor
BCUP	BSC Characteristics Update
BLDCPY	Build a Command File
BOOT	Execute J100 System Bootstrap
CHATR	Change File Attributes
CHBUFS	Change Buffer Pool
CHNOM	Change Nominal Disk
CHPRI	Change Primary Disk
CHRGEN	J500 Character Generator
COPY	Copy a File
CREATE	Create a File
DELETE	Delete a File
DISCON	Disconnect Communications Line
DOCCRT	Document a Screen
DSKCHK	Disk Structure Check
DSKOPY	Copy an Entire Disk
DSPLAY	Display and Edit Disk Data
! EDIT	File Editor
EXEC	Execute a Command File
FCMP	File Compare
FILES	FIle Status Inquiry
FLOPIZ	Initialize Floppy Disk
FORMAT	Format a Disk
GTAD	Time and Date Inquiry
HCREAT	Hash File Creation
HDELT	Hash File Deletion
HKOPY	Hash File Copy
J5LDAB	J500 Absolute Loader
JOBS	Job Status Inquiry
KILL	Terminate All Copies of a Program
LDABS	J100 Absolute Loader
LSPY	Line Driver Monitor
MONITR	Remote Terminal Monitor
MTUTIL	Disk/Tape Dump and Load
PARTS	Partition Inquiry
PAUSE	Pause in Command File
PCRT	"Print" on Screen
PRINT	Print a File
REMOVE	Remove a Disk
RENAME	Rename a Disk File
RLDR	Relocating Loader
RTDUMP	Real-Time Memory Dump/Edit
SDUMP	Stand-Alone Memory Dump
SI5DEF	J500 System Definition Inquiry
SIIDEF	J100 System Definition Inquiry
SORT	File Sort
SPRCAT	Disk Catalog Inquiry
STAD	Set Time and Date
TDSPLY	Tape Display
TRANSL	ASCII/EBCDIC Translation
UPDATE	Source File Update

Chapter 1: INTRODUCTION

This manual describes the standard utility programs for any Videocomputer controlled by System II software. It applies both to the J100 and to the J500, although a few programs are restricted to just one type of machine.

Every computer installation needs programs for copying disks, for printing files, for sorting data, and for similar general-purpose operations. These "utilities" are intended to lessen the need for "custom" or "application" or "user" programs.

You may be familiar with a computer system which has "control commands" or a "job control language." On the Videocomputer, the standard utilities serve this purpose. For example, when this line is entered at a terminal keyboard --

```
COPY APRIL MAY
```

-- it appears to be a complete command to the system, but it is, in fact, simply a request for the execution of a specific program -- namely, the COPY utility program. When that program begins execution, it analyzes the rest of the line itself, and takes action accordingly.

This manual often refers to a utility program execution request as a "System II command" for simplicity. A clear-cut distinction is needed only if your installation does not keep all the utility programs on a disk which is always on-line and ready for use.

Files and Directories

A "file" is a collection of related data which can be referred to by a name. When data must be sent to a line printer, for example, the name "LPT1" is typically used. The association between a name -- some letters and numerals in a command -- and a device -- a real piece of machinery cabled to the computer -- is set up during a process called "system generation" (or "sysgen"), as described in a separate manual.

A floppy or rigid disk can hold a number of separate files, so each disk includes a directory --also called a catalog -- which allows the system to locate and update each file. The SPRCAT utility can display a listing of all the files on a disk.

Names

One of these formats is used when a command must refer to a file:

```
device
device:filename
device:filename.extension
```

Each of the words in these formats is a "dummy" for a real device (such as LPT1 and FP00), a real filename (such as INFO and J1639), or a real extension (such as RB or 37).

For a reference to a device with no directory, such as a printer, there is no filename or extension. For disks, a specific filename, perhaps with an extension, must appear after the device name. As explained later, files located on certain disks (the "primary" and the "nominal" disk) may be referred to without actually naming the device.

A device name consists of 1 to 6 letters and numerals, the first of which must be a letter. Exactly the same rule applies to a filename. An extension -- a way of giving similar but not identical names to closely related files -- consists of either one or two characters, chosen from any combination of letters and numerals.

Note that a period -- and no other character -- is required to separate a filename from an extension. The colon used between a device and a filename may be replaced with a semicolon, as in these examples:

```
FP00;INFO
DP01;J1639.RB
```

The commands in System II all begin with filenames that agree with these rules. As mentioned already, each command is actually a request for access to a disk file containing an executable program.

Nominal Device

For each terminal on the system, at any given point in time, a specific device -- either a floppy or a rigid disk drive, and nothing else -- is designated as that terminal's "nominal" device. There is a command described later in this manual -- CHNOM -- which allows this designation to be changed at any time.

The nominal device may be different for each terminal on the system, and often is. If a given device is the nominal for a given CRT, files resident on it can be accessed from that CRT without specifying any device name. This allows considerable simplification in programming and in operational procedures.

For example, if the nominal device of CRT1 has been designated to be FP01, the following two references are equivalent in any command entered from CRT1:

```
FP01:AFILE
AFILE
```

Use of the "nominal device" convention has been built into many of the system commands. For example, some commands generate temporary files on the nominal device during their operation. If this is the case, the description of the command says so.

Primary Device

For every terminal in the system, at any given point in time, a particular disk device is designated to be the "primary" device. The CHPRI command can change this designation as needed, subject to certain constraints.

The primary device is always the same for all terminals; changing the primary at any terminal changes it everywhere.

When a particular disk is the primary, files on it can be referenced from any terminal by replacing the device name (and the colon) with a "$" character.

For example, if the floppy disk drive FP00 is currently the primary for the system, then the following references are equivalent in a command from any terminal:

```
FP00:GTAD
$GTAD
```

You will find, in later sections, some important rules about dealing with overlaid operating systems, with the software modules that drive peripheral devices, and with the system initialization command file SYSICF. Those rules are imposed because certain files must be on the primary device.

File Attributes

Every file in the system has certain "attributes." In the case of non-disk devices, these attributes are fixed, and reflect built-in characteristics of the devices. For example, a printer is obviously a "write-only" device.

Disk file attributes can be set to reflect properties of the data in the file, rather than of the device itself.

Since the system will support multiple users, it is prepared to arbitrate conflicting claims for files. Through its attributes, a file may be declared to be "read sharable," permitting several users to inspect it independently and simultaneously, or "write sharable," permitting several users to enter data in the file simultaneously.

Assembly language may be written to be reentrant, capable of supporting multiple users. The Super BASIC and Data-Rite compilers also generate such code. Programs may be given the attribute "execution sharable," so that a single memory-resident copy of the program can serve more than one user, thus saving central memory space.

Files may have other attributes -- they may be "permanent" or "batch" or "secure" or some combination of these and other attributes. Details appear under the SPRCAT utility. Another command, CHATR, gives you control over some attributes.

File Types

Under System II, there are two ways of organizing the data within a disk file: the "sequential" method and the "hash" method. The System II Reference Manual, and the publications for Super BASIC and Data-Rite, include details on the differences between these two file types.

Here it is only necessary to emphasize that the two types are distinct. Certain operations permissible and possible with respect to files of the sequential type are not permissible or possible with respect to files of the hash type, and vice versa. As a result, separate utility commands are provided -- for example, COPY and HKOPY; CREATE and HCREAT; DELETE and HDELT.

Special System Files

As noted above, all the commands described in this manual are actually program files which may reside on any disk device attached to your system.

For that reason, the "model" command line format given for each utility does not include any device indication -- neither a device name nor a dollar sign. If you must refer to a utility program -- or to any other disk file -- which is not on the current nominal device, then it is your responsibility to prefix its filename with a device and a colon, or with a dollar sign, as appropriate.

However, once they are invoked (loaded and executed), utility programs, other software packages (such as Type-Rite), and compiled programs may invoke additional software modules on their own -- programs which are "drivers" for peripheral devices, or which provide certain run-time functions, or which represent "overlays" for large programs.

With few exceptions, auxiliary modules of this kind have two uniform characteristics:

> Their names have the extension "SB" ("System Binary").
>
> THEY ARE ASSUMED TO RESIDE ON THE PRIMARY DISK.

If you are running with a fully resident operating system -- Configuration 1 -- you can easily change your primary device to a disk other than the one from which System II itself was loaded. You may invoke the utilities from that device or from any other. But you must have the required special system files on the new primary disk in order to operate successfully -- that is, without getting a series of mysterious "FILE NOT FOUND" error messages.

Device "driver" programs have names like MUXDRV.SB ("Multiplexer Driver"), DPRDRV.SB ("Diablo/Qume Printer Driver"), and so on. A complete list of such names may be found in the System II Reference manual. The only drivers needed on the current primary disk are, of course, those for devices which will actually be used at your installation. Disks and CRT terminals, incidentally, are handled by resident parts of System II, not by separate drivers.

Run-Time Support

In addition to device drivers, other system binary files may be needed on the current primary disk. The list below is not complete; Type-Rite, for example, has an associated set of ".SB" files, as described in the relevant manual.

SORTM.SB	Sort module, required by by SPRCAT, SORT, and other utilities, as noted in their descriptions.
SRTLIB.SB	Super BASIC Run-Time Library, required by any program which was written in that language, including the DOCCRT and BLDCPY utilities.
MTHLIB.SB	Super BASIC Mathematical Function Library, required by programs which use certain features.
BSORTM.SB	Super BASIC Sort Module, required when a program uses a SORT statement. This module, in turn, requires SORTM.SB for its operation.
DEINTR.SB	Data-Rite Interpreter, required for the execution of a program which was written in Data-Talk.
DEBUGX.SB	Debugger, as described in a separate manual.

Removing a Disk

While we're describing disk files, there's a very important rule you should know about --

ALWAYS ENTER A "REMOVE" COMMAND
BEFORE YOU REMOVE A DISK FROM ITS DRIVE.

If your terminal isn't ready to accept such a command at the moment, then you shouldn't be removing a disk; check your procedures. Further details may be found later in this manual, under REMOVE. (Incidentally, Type-Rite has its own command -- RM -- for this purpose.)

System Overlays

To save central memory space, you may be using an operating system which has "overlays," with only a portion of System II resident in memory continuously. Other parts are brought into memory (or wiped out) as needed. Overlays are files which must reside on the primary disk. If an overlay is needed, but is not available, the system freezes with the special message "SOVL" until the problem is corrected. This implies several rules:

1. The primary disk can be changed by a CHPRI command only if the disk to which it is changed contains an identical copy of the operating system.

2. The primary disk can be taken out after a REMOVE command, but only if it is immediately replaced with a disk containing an identical copy of of the operating system.

3. The primary disk cannot be taken out, even temporarily, if any programs are active anywhere in memory. If you do take it out, you must replace it in the unit before entering any more commands on the keyboard.

4. Utilities such as FORMAT and DSKOPY, which perform operations equivalent to "erasing" the contents of disks, should be used with care. In particular, they must not be requested to write onto the primary if that disk contains the overlays of a system in current use, because the overlays will be destroyed.

<RETURN> Key

You will find, as you study the rest of this manual, that some commands allow or require a large number of parameters. In the course of keying in a complex command line, you may make a mistake. But the computer will not take off on its own and attempt to execute your errors.

No command is ever executed until you press the key labeled <RETURN>. Until that moment, you may use the <HOME>, <TAB>, <BACKTAB>, and arrow keys to reposition the cursor. Then type over your mistake before executing the command with a <RETURN>.

Because it is the normal way of terminating any keyboard input for System II, the <RETURN> key is omitted from most of the model command line formats in Chapter 2.

Screen Roll-Down

If you accidentally execute a faulty command line by pressing <RETURN> before you have noticed your error, you may get an immediate error message. In this case, you can "retrieve" your command line with the down arrow key. Each time it's pressed, this key rolls down the entire screen by one line. When the rejected command line is back on the bottom line, you may type over your mistakes before once again executing the line with another <RETURN>.

<CANCEL> Key

If you execute a command line and then change your mind, you can usually press the <CANCEL> key to abort the utility you invoked. Some utilities, and some application packages written in BASIC, "hold off" termination at certain times.

A few utilities (JOBS and FILES, for example) put up full screen displays and do not return the cursor to the bottom line; for utilities of this kind, the <CANCEL> key is the normal way of terminating the program's operation.

Other utilities release the terminal after they load, and run "detached" (PRINT is an example). To terminate utilities of this kind, you must invoke the utility KILL.

If a task is in system code when the <CANCEL> key is pressed, it will not be aborted until it is back in program code. All of a job's assigned resources are released after the last task is terminated. There may be a delay of up to 30 seconds before the cursor returns after a <CANCEL>.

When a task is aborted, the instruction address being executed at that moment, and the task's CPU registers, are displayed:

```
***TASK ABORTED AT aaaa R0=wwww R1=xxxx R2=yyyy R3=zzzz
```

Executing a Utility

For each utility described in Chapter 2, there is a model command line format given. Though many of the command lines are quite complex, they can all be generalized in the following ways, each of which must, of course, be followed by a <RETURN>.

```
name
name/switches
name fields
name/switches fields
```

A System II command is simply a request for the execution of a program -- and that, in turn, is accomplished by simply typing in the name of a disk file which contains the program in object code format, ready to run. Previous sections explained how the name may or may not require an explicit device identifier.

If a non-disk input device, such as a magnetic tape, is positioned so that the next data accessed is a valid object file, then entering its device name alone is also acceptable as an execution request.

These same rules apply to an application program in any language, once it has been compiled and, if necessary, processed with RLDR. The reference manual for each language includes details on this point.

Except for the program name itself, everything on a command line may be optional or required, depending solely on the specific program. A considerable part of each program write-up is devoted to just this topic.

"Switches" are indicators appended directly to the program name, with no spaces between them. Each switch consists of a slash and a capital letter, like this:

```
NAME/A
NAME/A/B/X/Y
```

"Fields" are other items expected by the program; each field must be preceded by one or more spaces. A field may have its own switches:

```
NAME ALPHA
NAME BETA/Q/R GAMMA
NAME/S DELTA/T/U/V
```

To allow certain fields to appear in any order, some programs expect each one to consist of a capital letter, an equal sign, and a value -- perhaps a number, an English word, a file or device name -- with switches as needed:

```
NAME ALPHA BETA Z=13 S=MEMORY L=LIST.01/A GAMMA DELTA
```

Note that comments are not permitted on execution request lines.

Program Load Message

After a keyboard request has loaded a program into memory, and just before execution actually begins, the following message is displayed by the system:

```
yy/mm/dd BS=nnnn TS=nnnn BL=nnnn TL=nnnn SA=nnnn
```

The "yy/mm/dd" field is the creation date of the object program just loaded. The remaining fields, all of which represent hex words, may be useful to assembly language programmers:

BS and TS	BSECT and TSECT starting addresses
BL and TL	BSECT and TSECT lengths, in hex words
SA	Starting address for execution

Errors during program loading, and all fatal errors during program execution, generate standard screen messages. These are described in a separate publication -- the System II Error Messages manual.

Initial Command File

When System II is brought into memory by the "bootstrap" or "boot" process -- typically by pressing a switch on the Videocomputer -- one disk file, containing commands and comments, can be executed automatically.

The file must be on the primary disk -- from which the system was just loaded -- and it must have this fixed name:

```
SYSICF
```

If no such file is found, then CRT1 simply waits for commands from the keyboard, just as other terminals always do. If SYSICF is found, then CRT1 behaves as if this command had been entered from its keyboard:

```
$EXEC $SYSICF
```

Further details appear under the EXEC utility.

Remote Execution

When a given terminal is used to execute a program, the program normally runs "on" that terminal -- that is, any keyboard input or screen output operations use that terminal. However, any terminal (or command file, or internal "link" request) can initiate a program on a different terminal. This is done by prefixing the program name with the desired terminal name in angle brackets, as in this example:

```
<CRT3>PGM
```

Debugger Control

A program can be loaded without being executed, if the System II Debugger is given control at the same time. This is done by prefixing the program name with a "#" character. Examples:

```
#PGM
#FP01:APPL.JQ
```

Whenever the Debugger has control of its terminal, every user program in the system is frozen. As the operator of the Debugger terminal starts or stops the execution of the program being tested, all other programs start or stop as well.

A combination of remote execution -- "<terminal>" -- and Debugger control -- "#" -- is allowed, but only for a J100/J105 CRT, not for an asynchronous device. A typical command:

```
<CRT3>#PGM
```

Chapter 2: THE UTILITIES

This chapter contains a separate description for each standard utility program. To allow for continual updating, each program has its own set of page numbers, and the sets appear in strict alphabetical order by program name.

Most utilities are distributed on a diskette labelled "RnnGEN" -- "Release n.n General Files." This diskette also contains some programs which have their own manuals, such as the Debugger
! (DEBUGX.SB) and the Assembler (ASM). A few special-purpose utilities, such as LDABS, are distributed only on the diskettes for test programs or communications packages.

Command Line Formats

Each utility execution request is shown as a "command line format," in which a special notation is used. For example:

```
UTLPGM[/switches] device [L=fn[/switch]] file [file]...
```

This format covers, among many other possibilities, these lines:

```
UTLPGM FP00
UTLPGM/X/Y FP00 L=ABC.DE
UTLPGM FP00 ALPHA
UTLPGM FP00 ALPHA BETA GAMMA DELTA
UTLPGM FP00 L=ABC.DE/R ALPHA BETA GAMMA DELTA
```

Names and fields in uppercase (capital) letters are to be used exactly as indicated; so are slashes, equal signs, and colons. Field values in lowercase (small) letters represent "dummy" items, for which acceptable substitutions are described later.

A set of square brackets indicates an optional entry, subject to clarification in the write-up. An ellipsis ("...") indicates that the preceding entry may be repeated as often as needed. Neither brackets nor ellipses would appear in an actual command.

Switches may appear in any order. This is also true of fields prefixed with capital letters and equal signs. Order is sometimes significant for fields without prefixes; this will be noted in the write-ups.

A System II command line is limited to 80 characters at most.

ACUP -- Asynch Characteristics Update

ACUP can alter some generated characteristics for an asynchronous communications device or printer which is not currently open. These alterations are lost when System II is reloaded.

Command Line Format

ACUP[/switch] device [characteristics] [values]

One optional switch may be appended to the utility's name:

/S Set the characteristics specified, adding to the rest. This is assumed if no switch appears in the command.

/C Clear only the characteristics specified; leave the rest.

/R Replace the entire set of current characteristics with just those specified in the command; if there are none, clear all of the current characteristics.

The optional series of device characteristics may include any of those allowed during system generation -- TS, IT, HD, and so on. Note that ACUP can't reassign a device to a different driver.

Each optional value is a letter, an equal sign, and a number. In the following lists, "h" and "d" represent the same hex and decimal values described in the system generation manual.

J100 or J500

A=h	Address of device
B=h	Buffer size
L=d	Lines per page
C=d	Columns per line
T=d	Timeout limit, in tenths of a second; T=0 indicates an infinite time
R=d	Baud rate (bits per second) -- 50, 75, 110, 134 (meaning 134.5), 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200

J500 Only

H=d	Bits per character (excluding parity), from 5 to 8
P=d	Parity generation or checking -- none (P=0); odd (P=1); even (P=2)
S=d	Stop bits -- 1 (S=1); 1.5 (S=2); 2 (S=3)
X=d	Clocking control, from 0 to 9

Examples

ACUP ASY1 TM EM R=2400

This command sets the characteristics TM and EM for device ASY1, without disturbing other characteristics set previously. At the same time, the baud rate is set to 2400.

ACUP/R ASY1

This command clears all characteristics for device ASY1.

ACUP/C ASY1 TM

This command clears just one characteristic, TM, without disturbing any others.

ALTER -- Alter Contents of Memory

ALTER allows the contents of the Videocomputer's memory to be modified from a terminal keyboard or a command file.

This utility may be used to apply patches to System II, typically through a command file executed immediately after bootstrapping. (See the section on the SYSICF file in Chapter 1.)

For more elaborate manipulation of memory, two other programs are available -- the RTDUMP utility (described in this manual), and the System II Debugger (described in its own separate manual).

Command Line Format

```
ALTER  addr, value [,value]...
```

addr — is the hex address to be altered.

value — is the new hex value for one word. A decimal prefix -- D' -- is also be recognized. If you supply multiple values, they are stored into successive words -- "addr" and "addr+1" and so on.

An attempt to alter read-only memory, or memory not actually installed, is simply igored, with no error message.

Example

```
ALTER 50, 1, 2, FF28, D'255
```

This command stores, starting at location X'0050, the values X'0001, X'0002, X'FF28, and D'255 (that is, X'00FF).

ANSWER -- Auto-Answer Modem Monitor

ANSWER allows a command line to be executed automatically when an incoming call is received through an auto-answer modem.

Command Line Formats

```
ANSWER[/D][/T]   [C=dev]   F=cmd
ANSWER/K         [C=dev]
```

/D is optional; it executes ANSWER in background mode, detached from all terminals. If this switch is omitted, ANSWER remains in foreground mode.

/T is optional, but recommended. Without /T, the "F=cmd" field is executed after the first ring. If /T is specified, ANSWER instead waits up to 10 seconds for a second ring. If one occurs during that time, then "F=cmd" is executed; otherwise, the first ring is ignored -- presumably it was noise on the line -- and ANSWER continues to monitor the device.

C=dev is an optional field; it specifies the communications device to be monitored. The default is C=BPE1.

F=cmd is a required field unless /K appears. The command to be executed when a call is received is everything after "F=" through the end of the line.

/K kills (cancels immediately) the ANSWER job which is monitoring the specified device.

Error Processing

JOB WAS NOT LOADED

It was not possible to execute the command in the F= field.

NO F=

An F= field is required unless /K appears.

TABLE OVERFLOW

The system can handle a maximum of 10 concurrent ANSWER jobs.

Examples

```
ANSWER/D/T C=BPE1 F=BRJE BPE1 L=LPT1
```

ANSWER runs detached, monitoring BPE1. When a call is received, and two rings have been detected within ten seconds, this System II command is executed:

```
        BRJE BPE1 L=LPT1
```

```
ANSWER/K BPE1
```

The ANSWER job monitoring BPE1, and only that job, is killed.

BCUP -- BSC Characteristics Update

BCUP can alter some of the generated characteristics for a BSC device. These alterations are lost when System II is reloaded. BCUP can be used only with a device which is not currently open.

This System II command is used to execute the utility:

BCUP[/switch] device [characteristics] [values]

One optional switch may be appended to the utility's name:

/S Set the characteristics specified, adding to the rest. This is assumed if no switch appears in the command.

/C Clear only the characteristics specified; leave the rest.

/R Replace the entire set of current characteristics with just those specified in the command; if there are none, clear all of the current characteristics.

The optional series of device characteristics may include any of those allowed at SYSGEN time -- TT, PP, MC, and so on. Note that BCUP can't reassign a device to a different driver -- that is, neither the line control code (ASCII or EBCDIC), nor the operating mode (point-to-point or multipoint) can be altered.

Each optional value is a letter, an equal sign, and a number. In the following lists, "h" and "d" represent the same hex and decimal values described in the system generation manual.

A=h	Hex address of device
B=d	Block size
C=d	CU for multipoint
D=d	DN for multipoint
L=d	Line error retry limit
P=d	Poll/select retry limit
T=d	Device type
X=d	Clocking control for J500
R=d	Baud rate (bits per second) -- 50, 75, 110, 134 (meaning 134.5), 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200

Examples

BCUP BPE1 TT PP T=0

This command sets the characteristics TT and PP for device BPE1, without disturbing other characteristics set previously. At the same time, the device type is set to 0.

BCUP/R BPE2 C=0 D=31

This command clears all characteristics for device BPE2. It also specifies a CU of 0 and a DN of 31.

BCUP/C BPE1 DE

This command clears just one characteristic, DE, without disturbing any others.

BLDCPY -- Build a Command File

This utility program can generate a command file for various operations, including:

> Logically copying selected files -- sequential only, hash only, or both -- from one disk to another.
>
> Clearing the attributes of selected files.
>
> Deleting selected files.

BLDCPY itself is a BASIC program. The run-time support package SRTLIB.SB must be available on the current primary disk whenever BLDCPY is executed.

This program can be used interactively, or with a command line which supplies all the needed fields. In either case, BLDCPY expects to use a file created by a previous execution of this System II command:

```
SPRCAT/D device L=filename
```

That is, BLDCPY's input is a sorted directory listing on a disk file.

A typical use for BLDCPY is in copying all of an AM Jacquard software package from the supplied floppy to a system disk -- in the following example, from FP01 to DP00:

```
$CHNOM DP00
SPRCAT/D FP01 L=CAT.JQ
BLDCPY I=CAT.JQ O=EX.JQ F=FP01 T=DP00
EXEC EX.JQ
```

Interactive Command Line Format

```
BLDCPY  <RETURN>
```

The omission of command line fields indicates that BLDCPY is to operate interactively. This question will appear first:

```
WHAT IS THE NAME OF THE INPUT FILE (SPRCAT LIST)?
```

Key in the filename used for the "L=" field of the SPRCAT/D command executed previously.

```
WHAT IS THE NAME OF THE OUTPUT FILE?
```

Key in a file name of your choice for the output of BLDCPY.

```
IF THIS FILE ALREADY EXISTS,
DO YOU WANT THE NEW OUTPUT APPENDED?  (DEFAULT IS NO)
```

If you take the default, BLDCPY will automatically delete and replace the output file you named, if it is present.

```
WILL YOUR OUTPUT RUN UNDER "EXEC" OR "COPY"?
(DEFAULT IS EXEC)
```

If you key in COPY in response to this question, BLDCPY will generate an indirect file of paired file names that you can only use in a command of the form "COPY filnam/I." If you simply press <RETURN>, BLDCPY will assume that you want a command file using more than one utility, and that you want to copy hash files, too, if they are present.

```
DO YOU WISH TO CHATR FILES? (DEFAULT IS NO)
DO YOU WISH TO DELETE FILES? (DEFAULT IS NO)
```

Key in YES to each of these questions if you want your command file to include a CHATR or a DELETE for each selected file. Otherwise, press <RETURN> for each question.

```
DO YOU WISH TO COPY FILES? (DEFAULT IS YES)
```

Key in NO if you want a command file that only executes CHATR or DELETE; press <RETURN> to take the default.

```
DO YOU WISH THE /R (REPLACE) FOR COPY? (DEFAULT IS YES)
```

If you take the default, the "outfil" member of each pair of file names generated by BLDCPY will be followed by "/R." COPY will automatically CHATR and DELETE the files on the target disk if they are present and if their attributes are the same. There is one exception to this rule: files that are both attribute protected and permanent will not be replaced.

```
    WHEN COPYING HASH FILES
    BETWEEN DISK DEVICES OF DIFFERENT TYPES
    YOU MAY OPTIONALLY CHANGE THE FILE PARAMETERS.
    IF YOU WISH TO DO SO ENTER 'YES',
    IF NOT JUST PRESS <RETURN>.
```

If you press <RETURN> in response to this question, BLDCPY will skip over the next question.

```
    ENTER FILE PARAMETERS IN THE FORMAT 'K=F:T'
    (FOR MORE INFO SEE SYS II MANUAL)
```

If you are copying hash files between disks of different kinds, you may want to redefine sector size so that the number of nodes in the file remains approximately the same after the copy. "K" is the ratio of words per sector for the "from" disk versus the "to" disk. Here are the words per sector for some device types: 64 (FPnn); 256 (FAnn, FDnn, DDnn, DPnn); 512 (DAnn, BDnn).

In any case, the next question is:

```
    WHAT IS THE DISK DEVICE
    FROM WHICH TO LOAD CHATR, DELETE, COPY,...?
    (DEFAULT IS $)
```

Key in a disk device name if you want to override the default (which will look for all utilities on the primary disk).

```
    WHAT IS THE DISK DEVICE TO COPY FROM? (DEFAULT IS $)
```

Key in a disk device name if you want to override the default (which will look for "infil" in the logical copy commands on the primary).

```
    WHAT IS THE DISK DEVICE TO COPY TO? (DEFAULT IS NOMINAL)
```

Key in a disk device name if you want to override the default (which will assume that you want to transfer files to the nominal; attribute changes and deletions will also occur on the nominal if you take the default response to this question).

```
    DO YOU WISH TO COPY ZERO-LENGTH FILES? (DEFAULT IS YES)
```

There may be zero-length files on the disk which were created deliberately (with the CREATE utility, for example). On the other hand, they may be leftover temporary files, and not worth
! copying. Choose your response accordingly. Neither the Volume
! Identification (Volid) file, nor any other zero-length file with
! the attributes "AP,PF" is ever selected for processing.

ENTER FILE NAME PATTERN FOR FILE SELECTION (DEFAULT IS -.-)

! You may wish to select files only if their names have some
! characters in common; for example, all files with the extension
"SB" or all Type-Rite files (whose names begin with "TY") or
all files whose names contain precisely three characters in the
main part. For this purpose, in addition to the "." and the
alphanumeric characters, two special characters are defined:

-	Any number of arbitrary characters, including none at all.
*	Exactly one arbitrary character. You may use a sequence of asterisks to represent a specific number of arbitrary characters.

You may use both special characters in defining a given pattern.
Some examples of patterns, and the groups of filenames selected:

-.SB	Selects files whose main part is anything, but with "SB" as the only desired extension.
TY-.-	Selects files whose first two characters are "TY" and whose remainder is anything, including both those with extensions and those without.
*B-.-	Selects all files which have "B" as the second character of the main part of their names.
-.-B-	Selects all files whose names have an extension with a "B" in it.
***.-	Selects file names whose main part has precisely three characters, both those with extensions and those without.
***	Selects file names whose main part has precisely three characters, with no extension.
-***-	Selects file names whose main part has at least three characters, with no extensions.
-***-.-	Selects file names whose main part has at least three characters, both those with extensions and those without.

! Regardless of the pattern used, names with these patterns are
! always excluded from selection:

```
!       SYS
!       SYS.0-
!       SYS.1-
```

When you respond to the last question, BLDCPY begins generating
its output, which rolls up the screen a line at a time and is
stored in the file you named above. You may then execute it
with EXEC, or modify it as needed with EDIT or Type-Rite.

Noninteractive Command Line Format

```
BLDCPY[/I][/A][/D][/C][/N][/Z]  I=fn
      [O=fn] [A=fn]
      [K=f:t] [Z=dsk]
      [F=dsk] [T=dsk]
      [N=pat]
```

For every question asked by BLDCPY in the interactive mode, there is a switch or a field in the above command line. "I=fn" is always required. Either "O=" or or "A=" must be included, but not both. Some switches, such as /C and /I, are also mutually exclusive.

/I produces an indirect file to run under COPY rather than a file of individual commands to run under EXEC. In this case, only sequential files are selected.

/A produces a CHATR command for every file in the input SPRCAT listing that is not attribute protected (AP)

/D produces a DELETE command for every file in the input SPRCAT listing.

/C produces a COPY command for every sequential file in the input SPRCAT listing and an HCREAT and HKOPY command for every hash file.

/N overrides the /R switch on the outfil member of COPY commands.

/Z prevents zero-length files from being copied.

I=fn is a required field naming the input file -- the result of a previous SPRCAT/D operation.

O=fn is an optional field naming an output file. If this file exists, BLDCPY will automatically delete and recreate it.

A=fn is an optional field naming an output file. If this file already exists, BLDCPY will append output to it.

K=f:t is an optional field giving the ratio of sector size on the source disk to sector size on the target disk. It adjusts the index size of hash files when the copy is going to occur between devices of different types. The default is a 1:1 ratio.

Z=dsk is an optional field naming the disk device from which the utilities (CHATR, DELETE, etc.) are to be loaded. The default is the primary disk.

F=dsk is an optional field naming the disk device from which files will be copied. Default: the primary disk.

T=dsk is an optional field naming the disk device to which files will be copied, and on which a CHATR or DELETE will be performed. Default: the nominal disk.

N=pat is an optional field specifying a pattern by which BLDCPY will select names for inclusion in the output file. The default pattern (-.-) will select all filenames.

Error Processing

ERROR IN COMMAND LINE

BLDCPY displays a screen of command-line information that precedes the generation of the output command file. If your command line cannot be executed, BLDCPY displays the message above, repeats your command line, and leaves the informational screen in place for your examination.

Examples

```
BLDCPY/A/D I=FP0CAT O=JUNKEM
```

This command invokes BLDCPY, which then reads FP0CAT on the nominal device and, for every file name it contains, generates a $CHATR command and a $DELETE command with respect to a file expected to be on the nominal device.

```
BLDCPY/A/D/C/N/Z I=BD0CAT A=CPYCAT K=8:1 T=FP00 N=*HA-.-
```

This command invokes BLDCPY, which then reads BD0CAT on the nominal device and appends to CPYCAT on the nominal device. BLDCPY looks for names in BD0CAT that have "HA" as the second and third characters. All files that meet this criterion are selected unless they are of zero length. The command file includes a CHATR, DELETE, and COPY for each selected sequential file. There is also an HDELT and HKOPY for each selected hash hash file, adjusting sector size in an 8:1 ration for FP00.

BOOT -- Execute J100 System Bootstrap

Executing BOOT is equivalent to pressing the APL button on a J100 Videocomputer, or pressing the INITIALIZE switch on a J100 console. It is not relevant to a J500.

WARNING: BOOT should be used only in an emergency, or on an idle system after REMOVE commands have been performed on all disk devices.

Command Line Format

```
BOOT    <RETURN>
```

When the command is entered, the screen clears and the word BOOT appears on the bottom line. The APL ROM looks for the system on whatever disk is specified by its device switch.

CHATR -- Change File Attributes

CHATR modifies a file's entry in the disk directory, declaring that file to have certain characteristics (attributes) that affect its use.

Command Line Format

```
CHATR fn[/sw] [fn[/sw]]...
```

fn is a required field specifying the name of a disk file. Several disk files, each with its own series of attribute switches may be specified.

/sw is an optional series of one or more two-character switches, each preceded by a slash. The list of all possible attributes is given in the description of the utility SPRCAT, along with more details about the attribute's meaning. Some attributes only the system can set; others are under your control. The controllable attributes are:

AP	Attribute protected
PF	Permanent file
RO	Read only
WO	Write only
ES	Execution sharable
RS	Read sharable
WS	Write sharable
SF	Secure file
TL	Top loading

A hash file is always made read and write sharable even when these attributes are not requested. A secure file is always made secure even when this attribute is not requested. The attributes cannot be changed while a file is in use. The attributes are not changed if any of the following conditions are true: (1) the file is attribute protected, (2) /ES is specified and the file is a hash file, (3) /TL is specified and the file is a hash file, (4) the file is secure and the job is not logged in as the file's owner.

The attribute /TL causes a file so CHATRed to be loaded at the top of memory. Its use is inappropriate, however, for files other than the special system files delivered from Jacquard whose extension is .SB.

Error Processing

If you have keyed in a command line containing several file names, and if CHATR is unable to locate or change the attributes of one of them, CHATR will terminate and display a system error message. A carat mark will point to the file name on which the error occurred. No file named in the series after the carat mark will have been CHATRed; you will have to key them in again.

Examples

```
CHATR PRGRM/TL/ES/RS
```

This command invokes CHATR, which flags the directory entry for the file PRGRM, declaring it to be top loading, execution sharable, and read sharable (PRGRM is presumably an executable module).

```
CHATR  RULES/RO/RS/PF  THING
```

This command invokes CHATR, which flags the directory entry for a file called RULES, making it read-only and permanent, but allowing several users to read it simultaneously, and a file called THING, declaring it to have no attributes at all.

CHBUFS -- Change Buffer Pool

CHBUFS can change two items in a system on a disk: the default date and the system buffer pool size. These changes do not take effect until the operating system is rebooted. (Do not reboot a system in which you have changed buffer size until all jobs have terminated and you have executed REMOVE commands for all disks.)

Command Line Format

```
CHBUFS sysfil Y=yy M=mm D=dd [B=xxxx]
```

sysfil — is a required field designating the name of the bootable system -- the name written in a special disk location by the program INSTAL (see the SYSGEN manual). For systems on AM Jacquard software diskettes, this name is always SYS.

Y=, M=, D= — are required decimal number fields designating the new system date: Y=year, M=month, D=day. This date becomes the default used to initialize and display the real-time clock at boot time.

B= — is an optional field specifying the new system buffer pool size in hexadecimal. This must be a multiple of 16. System initialization rounds the buffer pool size up to a multiple of the maximum buffer size.

Error Processing

BAD DATE

The new system date specified is not legal.

NOT A MULTIPLE OF 16

The B= parameter must be a multiple of 16, or it is rejected.

NO EXTENSION ALLOWED

The name of the bootable system must not have an extension.

FILE IS A DEVICE

The name given must not be identical to a device name.

Examples

```
$CHBUFS $SYS Y=78 M=3 D=31 B=1B00
```

This command invokes CHBUFS from the primary disk, and modifies the bootable system file SYS so that, when rebooted, the date 78/03/31 will be displayed. The system buffer pool size becomes X'1B00 words.

```
CHBUFS SYS Y=78 M=12 D=25
```

This command invokes CHBUFS, and modifies only the default date.

CHNOM -- Change Nominal Disk

CHNOM displays or changes the specification of the nominal disk on which files with unqualified (unprefixed) names are assumed to reside. Initially, the nominal disk is floppy drive FP00 for each terminal (if the default in SYSGEN has been taken). After the initial system load, CHNOM may be used to change the nominal disk associated with a given terminal.

Command Line Format

```
CHNOM [dsk]
```

dsk is a disk device (FP01, DP00, etc.). This device immediately becomes the new nominal disk for the terminal.

If the "dsk" field is omitted, CHNOM simply displays the identity of the current nominal disk for the terminal.

If you refer to a file on a disk other than the nominal disk, you must always prefix the filename with a device identifier -- for example, DP01:XYZ.

If the file is on the primary disk (usually also the location of the system), you can use a dollar sign as a prefix, rather than an actual device identifier -- for example, $ABC.

Note that if the CHNOM command is entered at a time when the current nominal disk is not also a disk containing the CHNOM program itself, some prefix will be needed for the command -- typically a dollar sign. Otherwise, a "PROGRAM NOT FOUND" error message will appear.

CHPRI -- Change Primary Disk

CHPRI displays or changes the specification of the primary disk, on which files with "$" prefixes, and files with special system software extensions (such as "SB" and "CT") must reside.

When System II is loaded into memory, the initial assignment of the primary disk is as specified during system generation. This need not be the disk from which the system was bootstrapped.

CHPRI may be executed from any terminal, at any time, to reassign the primary disk. This assignment then applies to all terminals until CHPRI is executed again, or until the system is reloaded.

Command Line Format

```
CHPRI [dsk]
```

dsk is a disk device (FP00, DP01, etc.). This device immediately becomes the new primary disk for the terminal.

If the "dsk" field is omitted, CHPRI simply displays the identity of the current primary disk.

If you refer to a file on the primary disk, you can always prefix the filename with an device identifier -- for example, FP00:XYZ -- but it's usually more convenient to use the special dollar sign prefix instead, as in $XYZ.

It is possible -- and for systems with rigid disks, quite common -- for the primary and nominal disks to be the same disk.

Note that a command involving the specification "$CHPRI" requires that the CHPRI program itself be available on the current primary disk, not on the new disk to which assignment is being made.

CAUTION: When using an overlaid system (such configurations as 2, 3, 4, 5, 7, B, and C), the primary disk designation must never be changed -- without reloading System II -- unless the new primary disk contains exactly the same version of the operating system. Likewise, any special system files required by the application must be on the new primary disk.

CHRGEN -- J500 Character Generator

On a J500 Videocomputer, software control is provided for the dot patterns displayed by characters in the range X'80 through X'FF. The CHRGEN utility allows the loading or the editing of a disk file containing these dot patterns. Such a file is called a "font."

Command Format for Font Loading

```
CHRGEN/L [font]
```

/L indicates a "font load." CHRGEN immediately reads the specified font, transfers it into the J500's character generator, and terminates execution. No screen messages are involved.

font is the name of a disk file containing a set of 128 dot patterns. If the "font" field is omitted from the command, a file distributed with other System II software -- $CHRGEN.SB -- is assumed.

The dot patterns thus established remain in effect until the next execution of CHRGEN, or until a power loss occurs.

It may be appropriate to include a CHRGEN/L command in the System Initial Command File ($SYSICF), as described in the Introduction to this manual.

Command Format for Editing

```
CHRGEN [font]
```

font — is a disk filename. If CHRGEN is later commanded to store the newly edited font, this is the file which will be replaced (or created); it must not have attributes which will interfere with replacement. If the "font" field is omitted from the command, the name $CHRGEN.SB is assumed.

Two types of J500 CRT are available. One displays each character as a matrix of green dots, 8 columns by 20 rows. The other uses an 8-by-10 matrix of white dots. When a CHRGEN editing session begins, the program therefore asks which type of screen is expected to be used with the font being processed. A "green font" can be loaded into a "white" J500, or vice versa, but the appearance of the displayed characters may not be satisfactory, because alternate rows of dots will be dropped or repeated to fit the matrix.

CHRGEN then provides a grid with either 10 or 20 rows for designing a character's dot pattern. A hyphen or a blank represents the absence of a dot; any other character -- typically an "X" -- represents the presence of a dot.

The cursor control keys, including <TAB> and <BACKTAB>, provide access to any grid position, and to the line indicating the hex value of the next character to be designed. The standard editing keys are also available; for example, <ERASE> or <RULER/PRINT> will clear the whole grid at once.

Initially, CHRGEN charts the entire font currently in the J500's character generator, which is not necessarily the font named in the command line, and which may be completely blank or non-existent. When a hex value from 80 to FF is entered on the top line, pressing <F1> displays the corresponding grid pattern, which may then be modified as desired.

When <F2> is pressed, the grid pattern takes effect in the J500's own character generator hardware -- on the top line, in the font chart, and in all subsequent displays, even after CHRGEN is terminated. However, <F2> does not record the pattern on disk.

If -- and only if -- <F5> is pressed, every character established with <F2> in this editing session is recorded in the specified disk file. Other characters in the file are, in effect, not changed; they are made blank on a newly created file. CHRGEN then terminates.

The <CANCEL> key may be used at any time to abandon the current execution of CHRGEN.

COPY -- Copy a File

COPY performs data transfers from one sequential file or device to another. COPY cannot be used on hash files; see HKOPY. COPY operations can take three different forms: copy and terminate; copy and return to user; indirect execution with a command file.

Copy-and-Terminate Command Line Format

```
COPY  infil[/O][/E][/T][/C]
      outfil[/O][/E][/N][/X][/D][/R]
```

infil is the name of a sequential file or device that is the "source" element in an infil/outfil pair.

outfil is the name of a sequential file or device that is the "target" element in an infil/outfil pair. That is, the copy occurs from infil to oufil.

/O causes odd parity check (if appended to infil) or odd parity generation (if appended to outfil).

/E causes even parity check (if appended to infil) or even parity generation (if appended to outfil).

/T causes tabs in the input file to be expanded to blanks in the output file.

/C causes blanks in the input file to be compressed to tabs in the output file.

/N suppresses parity generation on the output file.

/X extends the output file by appending the input.

/D deletes (clears) the attributes of the output file.

/R replaces the output file completely with the input. If outfil does not exist, it is created. If outfil exists and has the same attributes as infil, then the existing outfil is deleted and copy occurs from infil to a new outfil; this form fails if infil and outfil have the same attributes but those attributes include both AP and PF.

Tab compression/expansion uses predefined tab stops every 8 columns (i.e., tabs at 1, 9, 17, etc.) Tab compression is performed in a manner which allows exact recreation of the original text when tabs are expanded.

COPY automatically transfers the attributes of the input file to the output file unless the D or X option is included in the command line. The D option drops attributes on outfil; the X option retains the attributes of outfil even though they may be different from the attributes of infil.

If the E or O options are selected on infil, X'5C is substituted for each byte of the input file that is the wrong parity.

Copy-and-Return-to-User Command Line Format

```
COPY    <RETURN>
```

After COPY loads, the cursor awaits your input. Key in an infil/outfil pair (all switches are permitted) and press <RETURN>. COPY will perform the data transfer and return the cursor to prompt you for another infil/outfil pair. When the prompt is on display, <CANCEL> or <RETURN> will terminate COPY.

Indirect COPY Command Line Format

```
COPY cmdfil/I
```

cmdfil is the name of a sequential file, each line of which contains nothing but an input name and an output name, with whatever appended switches (as described on the previous page) are needed. The BLDCPY utility may be used to generate such a file; so may EDIT or Type-Rite.

/I is a required switch directing COPY to use cmdfil. COPY will then read and execute that file, one line at a time.

Error Processing

When COPY is linked to by a background job, none of these error messages is output. When COPY is linked to by a foreground job, it opens the job message terminal, writes the error message, and waits for operator input. If the input and output files are included in the command line invoking COPY and no error occurs, COPY does not open the job message terminal.

COMMAND ERROR

The command line is unacceptable.

OUTFILE NOT SPECIFIED

Command line omitted an output specification.

HASH INFILE (or OUTFILE) NOT ALLOWED

HKOPY should be used to copy a hash file to a hash file.

OPTION CONFLICT

(1) Both /E and /O, or both /T and /C, on same file.
(2) /N, but also either /E or /O, on same file.

ILLEGAL INDIRECT FILE REFERENCE

A file named in a "COPY/I" command was not sequential.

INFILE NOT FOUND

The specified input file does not exist.

EXTEND OPTION NOT SPECIFIED FOR OUTFILE

The output file already exists; neither /X nor /R was specified.

ATTRIBUTES DO NOT MATCH

/R was specified, but the input and output attributes conflict.

PARITY ERROR

/E (even parity) or /O (odd parity) was specified on the input file and a byte with the wrong parity was encountered. This byte is replaced by X'5C and copying continues.

Examples

```
COPY SMALL.SR/T DD00:BIG.SR/X
```

This command invokes COPY in the copy-and-terminate mode. It appends data from SMALL.SR (on the nominal disk) to an existing file called BIG.SR (on DD00) expanding tabs in SMALL.SR to the right number of blanks in BIG.SR.

```
COPY DD00:CLIST/I
```

This command invokes COPY in the indirect mode, using a file named LIST on DD00. To this command to execute correctly, CLIST must contain lines of the following kind --

```
$PROG    DD00:PROG/R
SMALL    $SMALL/X
$BIG     DD01:BIGGER
  .          .
  .          .
  .          .
```

-- all of which are legal infil/outfil pairs for COPY.

CREATE -- Create a File

This program creates a new directory entry for a previously non-existent sequential file, and allocates exactly one sector to that file.

This utility can be used to reserve a file name for future use, or to define a file so that its attributes may be assigned (by CHATR) before the data is written. For creation of a new hash file, see HCREAT.

Command Line Format

```
CREATE fn [fn]...
```

Each "fn" is a filename which does not already exist on the relevant disk.

Error Processing

If you have keyed in a command line containing several file names, and if CREATE is unable to make a disk directory entry for one of them, the program will terminate and display a system error message. A caret (^) will point to the file name on which the error occurred.

No file named in the series after the caret will be created; you will have to key them in again.

Example

```
CREATE AFILE $BFILE FP03:CFILE
```

This command invokes CREATE. It creates 3 new filename entries -- for AFILE on the nominal disk; for BFILE on the primary disk; for CFILE on FP03.

DELETE -- Delete a File

This program deletes a specified filename from a disk directory, and releases all disk space occupied by the file. The DELETE command may be used only for sequential files; for hash files, see HDELT. To delete all the files on a disk in a single operation, see the /C option of FORMAT.

Command Line Format

```
DELETE fn [fn]...
```

Each "fn" is a filename which actually exists. When several "fn" entries are used, each one is processed separately, so different device prefixes may appear as needed. On a floppy disk, several seconds may elapse while a file's sectors are freed and the bit map is updated.

Error Processing

If you have keyed in a command line containing several file names, and if DELETE is unable to locate or delete one of them, the program will terminate and display a system error message. A caret (^) will point to the name on which the error occurred.

No file named in the series after the caret will be deleted; you will have to key them in again.

Note that some file attributes, as described under SPRCAT, prevent the deletion of a file.

Example

```
DELETE AFILE $BFILE FP03:CFILE
```

This command invokes DELETE. It deletes 3 files-- AFILE on the nominal disk; BFILE on the primary disk; CFILE on FP03.

DISCON -- Disconnect Communications Device

The DISCON utility is useful for controlling access to a device with the following description:

> Asynchronous, on a J500 system; or BSC point-to-point, on any Videocomputer system.
>
> Connected to a dial-up line.
>
> Not given the characteristic "DC" -- Automatic Disconnect -- during system generation.

Command Line Format

```
DISCON[/B] [device]
```

If the "device" field is omitted, then BPE1 is assumed.

For an asynch device, DISCON simply turns off the modem control signal Data Terminal Ready (DTR), causing the data set to break a dial-up telephone connection.

For a BSC device, DISCON sends the standard BSC disconnect sequence -- DLE followed by EOT -- before turning off DTR.

If the optional "/B" switch is used, the specified device is made "busy" with regard to incoming calls. This guarantees that line is available for outgoing calls. To clear the "busy" status and allow incoming calls, DISCON must be executed again for the device, without the "/B" switch.

DOCCRT -- Document a Screen

This utility records the complete contents of a CRT screen on a sequential file or device -- typically a printer -- and may be useful in the documentation of programs. Screen characters not in the range X'21 to X'7F, such as protection markers, are
! recorded as spaces. The cursor's position is not indicated.

DOCCRT is a BASIC program. It requires access to one CRT (or asynch terminal) for execution, to another CRT for the data to be recorded, and to the file SRTLIB.SB on the primary disk.

Command Line Format

```
DOCCRT      <RETURN>
```

Once started, the program continues until terminated with the <CANCEL> key. Each of the following questions requires an answer which ends with the <RETURN> key.

TYPE IN NO IF YOU DON'T WANT EXTRA CHARACTERS

The "extra characters" make up a frame of screen line and column
! numbers, as illustrated on the next page. The frame may be useful for some purposes, but it must be printed on wide paper, then reduced or turned sideways to fit a normal page. To obtain
! the frame, simply press <RETURN>.

ENTER LISTING FILENAME AND PRESS RETURN

Enter the device or file to be used -- for example, FP01:WX.YZ or DPR1. Naming a printer will tie it up until DOCCRT is terminated.

CRT TO PRINT?

Specify the screen to be recorded -- for example, CRT2. The message "CRT NOT PRESENT" indicates an unacceptable response. A response which includes either of the characters "!" or ";" in any position will be recognized as a comment. Such a response -- a whole line of text if desired -- will be copied onto the listing device, after which the same question will appear again.

Once a CRT has been specified, and its screen has been recorded, the last question will appear again. DOCCRT is then ready to record another screen on the same output file.

! EXAMPLE OF OUTPUT

This "frame" and screen data were recorded while the specified CRT was being used by Type-Rite.

```
    1         10        20        30        40        50        60        70        80
    +---------+---------+---------+---------+---------+---------+---------+---------+
  1 +                              PRINT INITIALIZATION                               + 1
  2 +                                                                                 + 2
  3 +                                 Make any changes desired                        + 3
  4 +                                                                                 + 4
  5 +                        tenths of an inch shift to right     = 0                 + 5
  6 +                        tenths of an inch shift to left      = 0                 + 6
  7 +                        the name of the device to print on   = DH04              + 7
  8 +                        the first page to be printed will be = 1                 + 8
  9 +                        the last page to be printed will be  = 9999              + 9
 10 +                        the number of copies printed will be = 1                 + 10
 11 +                        pages to be printed (ALL,EVEN,ODD)   = ALL               + 11
 12 +                        the initial value for %PAGE will be  = 1                 + 12
 13 +                        print lines that start with a "\"    = N                 + 13
 14 +                                                                                 + 14
 15 +                        THEN PRESS ONE OF THE FOLLOWING KEYS:                    + 15
 16 +                                                                                 + 16
 17 +     F1   PAGE AT A TIME mode.  Pauses and blinks before each page to allow      + 17
 18 +           paper to be inserted; then press PRINT/ERASE to continue printing.    + 18
 19 +     F2   Pauses before printing FIRST PAGE ONLY, then press PRINT/ERASE         + 19
 20 +           to print entire document.                                             + 20
 21 +     F3   Prints the document with NO PAUSES.                                    + 21
 22 +                                                                                 + 22
 23 +                                                                                 + 23
 24 +Command?  P DD03;UTLUP.RP                          A          P=1   L=1      61+ 24
    +---------+---------+---------+---------+---------+---------+---------+---------+
    1         10        20        30        40        50        60        70        80
```

DSKCHK -- Disk Structure Check

DSKCHK analyzes the catalog and file structure on a disk, then reports such problems as unreadable sectors, bad catalog entries, and cross-linked files. These problems may arise from a hardware malfunction, from a system crash during disk output, or from a failure to execute the REMOVE utility before inserting a new disk. One symptom, readily apparent to users of EDIT or Type-Rite, is the intrusion of data from one file into another.

DSKCHK checks for cross-linked files by passing sector-use information records to SORTM.SB, the system's generalized sort module. This is called the "input phase." The records are sorted into ascending logical disk address order. Cross-linked files will result in successive records which indicate the same logical disk address. This is detected during the "output phase."

When processing of a disk starts, DSKCHK does the system call which returns the number of sectors marked "used" in the bit map. This number is adjusted and displayed as "n SECTORS USED" with "n" in decimal.

Unless DSKCHK is linked to, it uses the bottom line of the job terminal to report its progress. During the input phase, each time it begins a new primary catalog sector it updates the message "# OF SORT RECORDS: n" where "n" is a decimal count of the records which have been passed to SORTM.SB.

At the end of the input phase, this message is rolled up, leaving the bottom line blank. This allows the user to estimate what proportion of the input phase has been completed -- at the end of the input phase, the number of sort records will equal the number of sectors used, if there are no errors.

The information generated by DSKCHK may be used with the DSPLAY utility to further identify, and perhaps to repair, any problems.

Note that DSKCHK performs a sorting operation. SORTM.SB must therefore be available on the current primary disk, and there must be no active sort I/O translation table -- no $SORT.CT file.

Command Line Format

```
DSKCHK[/A][/U][/F]  [L=fn[/A]] [P=w] [S=dn]
                    [dsk1] [dsk2] [dsk3] [dsk4] [dsk5]
```

/A specifies that every used sector is to be reported.

/U allows adjustment of the bit map. When a used sector or fake sector is found to be marked "available" in the map, the corresponding bit is reset to indicate that the sector is, in fact, unavailable.

/F specifies that every lost sector is to be freed if it is usable. Before freeing a lost sector, it is zeroed and read. If either of these disk operations results in an error, the sector is not freed; it is reported as unusable. When /F appears, /U is assumed as well.

L=fn specifies the name of a listing file or device. The default is the job terminal. If DSKCHK is linked to, this field is required. The "/A" switch appends the listing to "fn" if it already exists; without the switch, a new file is created.

P=w specifies the number of words in SORTM.SB's work partition. The default is the minimum needed for a merge order of 3 on the scratch disk.

S=dn specifies a disk device name for sort scratch space. The default is the nominal disk. Approximately eight words of space are needed for each used sector on the disk being checked.

dsk1 to dsk5 specify up to five disks to be checked successively. Any mixture of rigid and floppy disks may be included. The default is the nominal disk. While each disk is being checked, DSKCHK must have exclusive access to it.

Initial Messages

CHECKED DISK IS SCRATCH DISK (or) LISTING DISK
BIT MAP WILL NOT BE CHECKED

This is a reminder that the disk to be checked is also the disk specified for the sort space or for the listing file. The bit map of the disk is not checked, because it will change while DSKCHK is running. Other types of checks are still performed.

Initial Errors

ONLY /A, /F, AND /U ALLOWED

Invalid switch appended to utility name in command line.

SCRATCH NOT A DISK

The S= field was used to specify the name of something other than a disk device.

NOT AVAILABLE xxxx

The disk "xxxx" is currently in use by some other job, and is not available for checking.

Input Phase Errors

**** aaaa CATALOG O=oooo A=pppp BAD ENTRY

The catalog entry beginning at offset "oooo" in sector "aaaa" is not a valid catalog entry. That is, its four-word filename field is not valid, or it is a sequential file entry specifying an impossible number of bytes in the last sector. Sector "aaaa" is in the chain anchored on primary catalog sector "pppp." After four bad entries in the same sector, DSKCHK proceeds to the next primary catalog sector.

**** CATALOG A=pppp LOOP

The chain of catalog sectors beginning at primary catalog sector "pppp" loops on itself.

**** filename O=oooo A=iiii LOOP IN HASH DATA

The chain of data sectors belonging to a hash file, and anchored in word "oooo" of index sector "iiii," loops on itself.

**** aaaa filename DISK ERROR

Immediately preceding this message, a standard system error message reported the disk error to the job terminal. This error occurred on the sector with logical address "aaaa," in the file with the name shown. If the sector is a bit map sector, the name "BIT MAP" appears instead. If the sector is a lost sector, the name is blank.

Output Phase Messages

If "****" precedes a message, an error occurred. These formats are used for one group of messages:

```
aaaa  name                          text
aaaa  name       O=oooo  A=iiii     HASH DATA
aaaa  *SYS*              A=pppp     CATALOG SECONDARY
```

The message fields have these meanings:

aaaa — Logical disk address in hex. Omitted if preceding address is identical.

name — Name of the file to which the sector has been allocated. *SYS* indicates a system allocation.

O=, A= — Sector aaaa is a data sector in a hash file, and it is in the chain of data sectors anchored at word oooo in index sector iiii.

A= — Sector aaaa is a secondary catalog sector in the chain anchored at primary catalog sector pppp.

text — One of these comments --

```
SYSTEM BOOT
BIT MAP
OVERLAY DIRECTORY
CATALOG PRIMARY
HASH INDEX (or) HASH INDEX 1ST
SEQUENTIAL DATA (or) SEQUENTIAL DATA 1ST
```

-- or one of the following messages.

FILE ENDS WITH A 0 LINK

Sector aaaa contains a zero link word. DSKCHK proceeds to the next file.

DISK ERROR ON LINK

The link word of sector aaaa contains the logical address of the sector on which the immediately preceding DISK ERROR occurred. DSKCHK proceeds to the next file.

SECTOR COUNT ENDS FILE (or)
LAST SECTOR ADDRESS ENDS FILE

The catalog entry for each sequential file should contain the number of sectors in the file, the first sector in the file, and the last sector in the file. However, starting at the first sector in the file, the number of sectors in the catalog entry was reached before the last sector in the file, or the converse. DSKCHK proceeds to the next file.

NON-ZERO LINK IN LAST HASH INDEX SECTOR

The link word in the last index sector of a hash file should be zero.

BEYOND END OF BIT MAP

There is no bit map bit for logical address aaaa; the bit corresponding to aaaa would lie beyond the actual end of the bit map. On a DDxx type device, there are usable physical sectors -- 6% of the disk -- for which this is true.

Following are other formats used for output phase errors.

**** aaaa M=mmmm O=ooo A=bbbb text

M= Mask. A hex number containing exactly one non-zero bit. This non-zero bit is the bit map bit corresponding to logical address aaaa.

O=, A= The bit map bit corresponding to logical address aaaa is mmmm, in the word at offset oooo, in the bit map sector with logical address bbbb.

text As detailed below.

LOST SECTOR

Sector aaaa is marked "used" in the bit map, but it is not a part of any file. The /F optional switch may be used to free all lost sectors. Before freeing a lost sector, it is zeroed and read. If either of these disk operations results in an error, the sector is not freed, and the sector is reported as unusable.

UNUSABLE SECTOR

The /F switch was used, but the attempt to write or read sector aaaa caused a disk error.

USED SECTOR FREED

Sector aaaa is marked free in the bit map, but it is part of a file. If a /F or /U switch was specified in the command line, DSKCHK corrects this error by marking aaaa as used in the bit map.

FAKE SECTOR FREED

Sector aaaa is marked free in the bit map, but it does not correspond to any physical sector -- the logical sector field or the logical track field is too large. If a /F switch was specified in the command line, DSKCHK corrects this error by marking aaaa as used in the bit map.

**** BIT MAP WILL NOT BE CHECKED FOR LOGICAL ADDRESSES xxxx to yyyy

A disk error occurred while reading the bit map sector containing the bits for logical addresses X'xxxx to X'yyyy. Consequently, the bit map can't be checked for these logical addresses.

DSKOPY -- Copy an Entire Disk

DSKOPY makes a physically identical copy of a rigid or floppy disk, sector by sector, without regard to file or free space boundaries. During this process, both the "source" and the "target" disk are made unavailable to other jobs.

Command Line Format

DSKOPY[/N] source target

/N specifies no generation or verification of a software checksum for each sector. This allows the processing of a disk which is not in System II format, although soft sectors are still required.

source is the disk device copied from -- FPxx, DDxx, etc.

target is the disk device copied to -- a device of the same type (FPyy, DDyy, etc.) as the source device. All of its previous contents will be destroyed.

A screen is required for a DSKOPY job; detached execution is not allowed. The job starts by displaying the source and target device names for checking. Press <CANCEL> to abort the job; press <RETURN> to perform the copy.

When copying and verification are complete, the message "DSKOPY COMPLETED - KEY <RETURN> TO EXIT" appears. DSKOPY does not terminate until this operator response is entered.

Even if the target disk has never been formatted, it is not necessary to run the FORMAT utility. The target disk will, in any case, become a duplicate of the source disk. However, if the target is a factory-fresh floppy with no "3740-compatible" initialization, it will be necessary to precede DSKOPY with FORMAT/I (or FLOPIZ).

DSKOPY allows the current primary disk to be named as its target. This possibility requires extreme caution, and usually implies that the system will be reloaded immediately after DSKOPY ends.

Error Messages

DISKERR C:n S:n H:n SW STATUS:h HW STATUS:h

A read or write error has occurred on the disk named above the message. The cylinder, sector, and head are decimal numbers. The status words are in hex; for details, see the System II Error Messages manual.

CMPR ERR C:n S:n H:n source-word target-word

During verification, a comparison error occurred at the indicated disk address. The words shown in hex do not match on the source disk and the target disk.

Error Responses

After an error message appears, these responses are accepted:

BE	Begin DSKOPY again.
RE	Retry the sector.
PR	Proceed, ignoring the error.
EX	Exit -- terminate DSKOPY immediately.

DSPLAY -- Display and Edit Disk Data

This program can display, in hex and ASCII, any area of a disk, or any file, even while another job is writing on the same sectors. DSPLAY can also be used to make spot changes to disk data or to software control words.

Whenever the message "DSPLAY" appears, followed by the cursor, a complete command line, of any type described below, may be entered. It is processed when <RETURN> is pressed. In this situation, <CANCEL> will terminate the program.

Normally, data is displayed in screen "pages." When the message "CONTINUE?" appears, the <RETURN> key brings up another page. The <CANCEL> key in this situation will abandon the current operation, allowing another command to be entered.

A listing file may be specified as needed. If a disk file is named, it must not already exist; otherwise a system error message will appear. The file is closed when the processing of the one current command is completed or abandoned. When a listing file is active, the screen is not paged; instead, the lines going to the listing file roll up continuously.

Interactive Operation

DSPLAY

This System II command simply brings up the prompting message "DSPLAY" -- allowing the entry of any command format described below. After each command has been processed, another command may be entered, or <CANCEL> may be used to terminate the program.

Sequential File

DSPLAY fn[/T][/E][/C] [L=fn]

fn specifies a sequential filename. If no switches are used, the entire file is displayed.

/T restricts the display to logical sector addresses, without any data.

/E restricts the display to the file's last sector.

/C restricts the display to a sector count only. This information is also available from SPRCAT.

L=fn specifies a listing file.

Hash Files

DSPLAY fn[/T][/C] [L=fn]

fn specifies a hash filename. If no switches are used, the index sectors and then the node sectors of the file will be displayed. First the message "HASH FILE INDEX SECTORS ARE:" appears, then the first index sector of the file. The next-to-last word, if nonzero, gives the address of the sector containing a continuation of the index sector. The last word is an internal check and can be ignored. Each of the other words anchors one chain of data sectors.

A chain of data sectors is referred to as a "node." If the node anchor contains zero, then the node is empty. If the node is not empty, then the node anchor contains the address of the first sector in the chain of data sectors comprising the node. The next-to-last word of an index or node sector (the link word), if nonzero, is the address of the continuation of the index or node sector. The sectors can thus be described as "forward linked."

After all the index sectors have been displayed, the message "HASH FILE NODE SECTORS ARE:" appears, and DSPLAY processes each node.

/T restricts the display to logical sector addresses, without any data. First the index sectors are listed, then the data sectors. Asterisks delimit each node.

/C restricts the display to a sector count only. This is the only convenient way to obtain this information for hash files.

L=fn specifies a listing file.

Logical Addresses

DSPLAY [dsk] A=i:j [/T][/L]] [L=fn]

dsk — is a disk device name. The default initially is trhe current nominal disk. After that, the default is the disk most recently accessed.

A=i:j — specifies, in hex, a a range of logical disk addresses to be displayed. The form A=i -- implying A=i:i -- is also allowed.

/T — displays the chain of disk addresses -- but not the data -- beginning at A=i, and linked as a sequential file.

/L — same as /T, but the data is also displayed.

L=fn — specifies a listing file.

Physical Locations

DSPLAY [dsk] [C=k:l] [S=m:n] [H=o:p] [L=fn]

dskdev — is a disk device name, as above.

C=k:l — is a decimal range of physical cylinders. Default: all cylinders.

S=m:n — is a decimal range of physical sectors per track. Default: all sectors.

H=o:p — is a decimal range of physical disk heads. Default: all heads.

L=fn — specifies a listing file.

Editing or Searching

Whenever the message DSPLAY and the cursor are on the bottom line, you may enter the "edit or search mode" by erasing the word "DSPLAY" -- typically by pressing <HOME> and <ERASE LINE>. In this mode, the following commands are available.

CS — Recompute sector checksums (the final word of each sector). Use the results for output and for verifying input. This rule is the default when DSPLAY begins execution, and remains in effect for all disk I/O until reversed with a command of NC.

NC — Do not generate or verify sector checksums. This rule remains in effect for all disk I/O until reversed with a command of CS. Note that NC allows access to a disk which is not in System II format, although soft sectors are still required.

DB — Display the current contents of the sector buffer, as determined by the most recent disk input command.

WS — Write the current contents of the sector buffer to the current disk sector, as shown in the header of the DB display.

AL offset value [value]...

Store the hex values into the sector buffer, starting at the specified hex offset. The first word in the buffer has an offset of 0.

UP fn [F=i] [E=j] [B=k]

Update the catalog entry for a sequential file, by forcing new values as specified. The file is rolled up the screen, and the sector count is updated in the catalog. This involves a check that the chain of sectors, beginning at the first sector in the file, reaches the last sector in the file. If DSPLAY detects any errors, no changes are made.

fn — is the name of a sequential disk file.

F= — is a new first sector value in hex.

E= — is a new end (last) sector value in hex.

B= — is a new decimal value for the number of characters in the file's last sector. This updates not only the catalog, but also the link word in the file's last sector.

FL word

Defines a link word, in hex, for the search mode. The link word occupies the next-to-last position of a sector. It is the address of the next sector in the file. Once FL is defined, subsequent commands to display a given disk file or region of the device will stop at the specified link word value. Entering "FL" without a parameter exits search mode.

FS 'string' or FS X'hex

Defines an ASCII string for the search mode, or a hex sequence which represents an even number of characters. Entering FS <RETURN> causes the message "TYPE IN STRING." You may then type in an ASCII string without quotes to define a search parameter, or simply press <RETURN> again to exit from the search mode. Once FS has been defined, subsequent commands to display a given file or region of the disk will search for the string.

FW word [word]...

Defines a word sequence for the search mode. Or, key in FW and press <RETURN>. The message "TYPE IN WORDS" then asks for a sequence. Once FW has been defined, subsequent commands to display a given file or region of the disk will search for the word sequence. FW followed by <RETURN> twice exits search mode.

FD Displays a search parameter (FL or FW word; FS string.)

For both FS and FW, sectors are displayed in 8-word lines, with the first occurrence of the search argument in each line underscored. A pattern of more than one character or word may cross a sector boundary. DSPLAY can not detect such patterns.

Error Messages

Ordinarily, if you key in something that DSPLAY does not recognize or cannot handle, the erroneous input will be ignored and one of the following messages will be displayed. DSPLAY follows the error message with a prompt for another command.

BAD INPUT!

The last command input contained an error.

ILLEGAL OUTFILE

You called for a listing file, and a file of that name already exists.

DISK RANGE SPECIFICATION ERROR

Your command specified a physical cylinder, sector, or head value that was too large for the device to be displayed.

FILE NOT FOUND

The disk device you specified is not a device in the system generated, or the file name you specified does not exist.

ILLEGAL DISK ADDRESS LINK

In passing down a chain of disk sectors, DSPLAY found a link to a nonexistent sector.

BYTE COUNT TOO LARGE

The B= parameter on an UP command specified that more than the maximum number of bytes per sector were used in the last sector.

ONLY THE CATALOG ENTRY OF A SEQUENTIAL FILE MAY BE UPDATED

On an UP command, a device or a nonsequential disk file was specified.

NOT A PHYSICAL SECTOR

In an "A=i" field (i.e., a range of one value), "i" is a logical disk address that does not correspond to a physical disk sector.

Examples

```
DSPLAY SFIL
```

This displays the sequential file SFIL, on the nominal disk.

```
DSPLAY SFIL L=LPT1
```

This displays and prints the sequential file SFIL.

```
DSPLAY FP01:SFIL/E
```

This displays the last sector of SFIL on FP01.

```
DSPLAY FP01:SFIL/T
```

This displays the logical disk addresses of all sectors in the file SFIL on FP01.

```
DSPLAY HFIL/C
```

This displays the total number of sectors allocated to the hash file HFIL on the nominal disk.

```
DSPLAY FP00 A=0:100
```

This displays all of the physical disk sectors with logical disk addresses between zero and X'100. (Note that some logical disk addresses do not correspond to physical disk sectors.)

```
DSPLAY FP01 A=5B9/L
```

This displays the chain of sectors on FP01 beginning at logical sector address 5B9.

```
DSPLAY A=5B9/T
```

This displays the logical disk addresses of the chain of sectors beginning at logical sector address 5B9, on the disk last accessed by DSPLAY.

```
DSPLAY DP00 C=187:190 S=5:9
```

This displays sectors 5 through 9 on all tracks (head 0 and head 1) of cylinders 187 through 190 on DP00.

Examples of Search and Edit Commands

AL 3E 4A7

If the current sector is a floppy sector, this sets the link word to logical disk address 4A7.

UP ABCD E=4A7 B=126

This updates the catalog and sector 4A7; it says that 4A7 has 126 bytes used in it and that it is the last sector in the disk file ABCD on the nominal disk.

FW 0000 1234

This places the words 0000 1234 in the search buffer. Until the buffer is cleared or written over, disk files or regions of disk devices that are displayed will be shown in rollup fashion, stopping only at sectors containing the word sequence 0000 1234. Those sectors will have the sequence underscored.

FS 'SSSSS'

This places the string 'SSSSS' in the search buffer. Until the buffer is cleared or written over, disk files or disk devices will be displayed in rollup fashion, stopping only at sectors containing the specified string, and underscoring the string.

FL 1942

This puts the link word X'1942 in the search buffer. Subsequent display of a disk file or a region of a device will occur in rollup fashion, stopping at the sector containing the link word specified. The message "THIS SECTOR HAS THE DESIRED LINK WORD" will precede the sector display.

EDIT -- File Editor

The EDIT utility program -- often referred to as "the Editor" -- is used to create or modify sequential disk files containing lines of printable data. Examples are source programs for ASM, DATRIT, SGBASC, or SBASIC; System II command files; input for system generation; key tables for SORT; application test data.

Execution is started with this System II command:

```
EDIT  <RETURN>
```

There are no parameters on the EDIT command line; all operations are handled through keyboard requests, as summarized here.

<CANCEL>	Switch from Text Mode to Command Mode.
<RETURN>	Switch from Command Mode to Text Mode.
<F1> & <F2>	Line up; scroll up.
<F3> & <F4>	Scroll down; line down.
<F5>	Screen up.
<F6>	Line insert and line up.
<F7>	Line delete and line up.
<F8>	String processing.
<F9>	Line split and line down.
<F10>	Line delete and line up.
R file	Read.
W file	Write.
A file	Append.
M file	Modify.
KM	Kill modifications.
CI & CO	Close input; close output.
FI file	File insertion.
DE file	Delete file.
RE file,new	Rename file.
Q	Quit using current files after scrolling through.
EX	Exit from Editor after scrolling through.
AB	Abort execution immediately.
T list	Tab stops.
TR n,list	Tab stops relative to column "n" stop.
MR left,right	Margins.
N start,step	Number lines.
RP n	Repeat current line "n" times.
LO & UP	Lowercase accepted; uppercase forced.
BO & NB	Blank lines output; no blank lines output.
FS	Find string.
IS & AI	Insert string on request; automatic insertion.

WINDOWS

The Editor divides its screen into a Text Window -- lines 1 through 22 -- and a Command Window -- lines 23 and 24. Two blinking underlines, called the Text Cursor and the Command Cursor, are maintained separately; each is visible only when keyboard input is allowed for its window.

The line number of the Text Cursor's position, relative to line 1 of all file and keyboard input, is continuously displayed at the end of line 24. The middle of line 24 is reserved for the names of the files being processed.

COMMAND MODE

When the Editor is in Command Mode, as it is initially, the cursor on the lefthand end of line 24 invites the entry of a command -- one or two letters, perhaps followed by a space and some parameters, then a <RETURN>. Line 23 displays either the command most recently executed, or the latest error message.

Typically, a series of commands will be entered to specify file names; margins and tabs; line numbering pattern; options to allow blank lines or lowercase text. After that, Command Mode will usually be needed only to set up string searches, or to terminate the processing of the current files.

The <RETURN> key without a preceding command, or the use of certain Function Keys, will switch the Editor into Text Mode.

TEXT MODE

When Text Mode is entered, the Command Cursor disappears; the Text Cursor reappears at the first tab stop of its current line. Line 23 displays a "ruler" with "+" signs at the tab stops and "-" signs at all the other positions within the current margins.

Unless scrolling or some other continuous process is underway, pressing <CANCEL> switches the Editor back to Command Mode.

Any line or character in the Text Window may be altered by positioning the Text Cursor, then using appropriate keys for data entry, editing, or function requests. Lines which are scrolled out of the Text Window may be brought back for further alteration or deletion as needed.

SPECIAL KEYS

In Text Mode, some keys have meanings unique to the Editor; others have the standard meanings described in the System II Reference Manual. For convenience, all of them are covered here.

The phrases "current line" and "current character" refer to the line or character which is currently over the Text Cursor, even if that cursor is not visible because the Editor is in Command Mode.

Any of the keys from <F1> through <F6>, if pressed in Command Mode, will automatically switch the Editor into Text Mode, then perform the operation defined for that mode.

Functions

<F1>	The text is shifted up one line.
<F2>	The text is shifted up until it's exhausted, or until <CANCEL> is pressed.
<F3>	The text is shifted down until it's exhausted, or until <CANCEL> is pressed.
<F4>	The text is shifted down one line.
<F5>	The text is shifted up one screenful -- 22 lines.
<F6>	The current line, and those above it, are shifted up one line, resulting in a new blank line.
<F7>	The current line is deleted; those above it are shifted down to close the gap.
<F8>	String processing is performed, as described in a later section.
<F9> or <LINE INSERT>	All text below the current line is shifted down one line. The current character, and all those to its right, are then shifted down too. This leaves the original line either split or blank.
<F10> or <LINE DELETE>	The current line is deleted; those below it are shifted up to close the gap.
<F11> to <F20> and <ALT>	Should not be used; results vary with keyboard and controller installed.

Other Text Operations

<RETURN>	Normally, a blank line is inserted. If an N command is active, a line number is generated.
<ERASE> or <RULER/PRINT>	A warning message appears; if the next key pressed is <ERASE> again, all of the lines in the Text Window are deleted. If a Function Key, or <RETURN>, or <CANCEL> is pressed, the message disappears and the previous <ERASE> is withdrawn.
<CHAR INSERT>	The current character, and all those to its right on the same line, are shifted right, creating a blank with the cursor under it.
<CHAR DELETE>	The current character is deleted; all those to its right on the same line are shifted left to close the gap.
<ERASE LINE>	The current character, and all those to its right on the same line, are deleted.

Cursor Movement

<Up Arrow>	The cursor moves up, wrapping around from line 1 to line 22.
<Down Arrow>	The cursor moves down, wrapping around from line 22 to line 1.
<Left Arrow>	The cursor moves left, wrapping back to the end of the previous line.
<Right Arrow>	The cursor moves right, wrapping forward to the start of the next line.
<HOME>	The cursor jumps to the first tab stop on the current line.
<TAB>	The cursor jumps to the next tab stop, wrapping forward to the first stop on the next line.
<BACKTAB>	The cursor jumps to the previous tab stop, but no further back than the first stop on the line.

COMMANDS

The valid Command Mode entries will now be described in functional groups. No command is executed until <RETURN> is pressed; until then, corrections may be made with such keys as <CHAR INSERT> and the cursor arrows. Note that all letters in commands are forced into uppercase; no <SHIFT> is needed.

FILE HANDLING

Where a file name is expected, it must be in one of the standard System II formats, as described in Chapter 1 of this manual. The processing of some commands, and the scrolling of lines off the screen, will involve temporary files named "xxxx.TP" and "xxxx.TM" with "xxxx" taken from the first four positions of the terminal's device name -- CRT1.TP, for example. File names with these patterns must therefore be avoided. Temporary files are allocated on the same disk as the input file, or on the current nominal disk if there is no input file.

Separate Input and Output Files

R file	(Read)
W file	(Write)

The R command specifies an existing file to be read when input is required for a scrolling or searching operation. The file is opened with a status of "Read Share" but no records are obtained until they're needed later. The file's name appears on line 24 -- in the form "R=file" -- and any previous "R=" file is closed.

The W command specifies a new file to be created; it will contain whatever lines scroll up and off the screen. When this file is eventually closed, its only attribute will be "RS" ("Read Sharable"), so it will be available for further reading or modification as needed. The file's name appears on line 24 -- in the form "W=file" -- and any previous "W=" (or "A=") file is closed.

If an input file is only to be scanned, with no modification intended, then no output file is needed. However, the warning message "NO OUTPUT FILE" will appear the first time that a line would have scrolled up and off, because such lines can't be scrolled back down. The command or special key which caused the warning may then be repeated.

If an output file is to be created solely from keyboard entries, then no input file is needed.

Appending Output

A file	(Append)

This command is similar to the W command, except that the specified file must already exist and must be available for output. Lines scrolled up and off the screen are added to the previous end of the file. Only these added lines can be scrolled back down onto the screen, or tracked by the Command Window line counter. The form "A=file" appears on line 24 when an A command is accepted.

Modifying a File

M file	(Modify)
KM	(Kill Modify)

The M command specifies an existing file to be modified -- that is, to serve as both input and output. The file's name appears on line 24 -- in the form "M=file" -- and any previously active files are closed. No R, W, A, or M command is accepted while an "M=" file is open.

Lines scrolled off the top of the screen go onto a temporary file, and the "M=" file is not actually modified until it is closed by a Q or EX command. Until then, entering a KM command will "kill" -- that is, nullify -- all changes made since the last M command, and close the file.

Explicit Closing

CI	(Close Input)
CO	(Close Output)

The CI command immediately closes the current "R=" file. This is typically followed by a request to modify or delete the file, neither of which is possible while the file is open for reading.

The CO command immediately closes the current "W=" or "A=" file. This prevents the output of any further lines which scroll off the top of the screen.

Inserting a Secondary File

FI file	(File Insertion)

The specified file is opened; all of it is immediately inserted under the Text Cursor's line; the file is closed. One blank line is then created after the inserted text, with the Text Cursor on that line.

The FI process may be terminated prematurely by pressing <CANCEL>.

Renaming or Deleting

```
RE file,new          (Rename)
DE file              (Delete)
```

The RE command immediately changes a file's old name to the specified new name, which must not already exist in the directory of the relevant disk.

The DE command immediately deletes the specified file, which frees all the space it occupied, and its name as well. Further access to the data is impossible.

Neither of these commands accepts a file which is currently in use by any job in the system, including the Editor itself. Furthermore, the directory attributes of the file must allow the requested operation, else an error message will appear.

Avoiding Tab Compression

```
WF                   (Write Full)
AF                   (Append Full)
MF                   (Modify Full)
```

These commands are used when output lines must be "full length" rather than "tab compressed" as the Editor normally makes them. In all other respects, these commands are equivalent to the W, A, and M commands.

Tab compression means that consecutive spaces (X'20) preceding the system's standard tab stops (columns 9, 17, and every eighth column following) are not written to an output file; instead, Horizontal Tab characters (X'09) are inserted. The resulting lines are not acceptable in these contexts:

> Input data files for applications written in Data-Talk, or for the DATXFR utility. Tab compression in a source program file is recognized, however, by the DATRIT compiler.
>
> Transparent input operations, such as an INPUT COUNT in BASIC or an RDS call in assembly language -- unless the program's own logic is prepared to decompress the data.
>
> Binary input operations which expect fixed field positions, such as SORT with comparison codes other than CA or CD.

To generate an output file intended for these purposes, a WF, AF, or MF command must be used.

The Editor drops trailing spaces from output lines, whether tab compression is in effect or not. Each line contains an even number of characters, and is terminated with an ASCII Carriage Return (X'0D), preceded by one space if necessary.

TERMINATION

Q	(Quit)
EX	(Exit)
AB	(Abort)

These three commands all terminate the processing of the current input and output files, but the results are quite different.

When a Q command is entered, the Editor scrolls everything in the Text Window, and all the lines below that, onto the output file. Then all the open files are closed. The Editor is now ready to work on a completely unrelated set of files.

An EX command also causes "scrolling through" the text to complete the output file. But once that has been accomplished, execution of the Editor ends, and System II itself regains control of the terminal.

An AB command requests the immediate end of Editor execution. All open files are closed, of course, but no "scrolling through" is performed. Any "W=" or "A=" file will include only the text which is currently above the top of the screen. An "M=" file will be unmodified, just as if a KM command had been entered.

FORMAT CONTROL

The commands in this group may be entered as often as needed; they are provided as conveniences for the keyboard operator.

Tab Stops

```
T list                  (Tab Stops, Absolute)
TR n,list               (Tab Stops, Relative)
```

The Editor's tab stops, to which the Text Cursor will skip during various operations, are initially set to the system's standard values -- columns 1, 9, and every eighth column folowing. The entry of a T or TR command clears all existing stops and sets new ones. A list of up to nine new stops, in left-to-right order, is acceptable. The "ruler" which appears on line 23 in Text Mode indicates the current settings with "+" signs.

The T command puts the stops at exactly the columns specified. For example, the following command restores the initial settings:

```
T 1,9,17,25,33,41,49,57,65,73
```

The TR command makes column "n" the first stop, then uses the list for the relative spacing of the remaining stops. The initial settings could be restored by this command:

```
TR 1,8,8,8,8,8,8,8,8,8
```

Margins

```
MR left,right           (Margins)
```

The Editor's initial margin settings are at columns 1 and 80. When an MR command is entered, the "left" and "right" values specified become the new margins. All lines in the Text Window are trimmed to these margins, and so is the "ruler" which shows tab stops in Text Mode.

The MR command also clears all of the current tab stops, and automatically sets new stops as if this command had been entered:

```
TR left,8,8,8,8,8,8,8,8,8
```

If these tab stops are not acceptable, a T or TR command may, of course, be entered after an MR.

Note that the home position for the Text Cursor is at the first tab stop, and not at the left margin, if these positions are different.

Numbering Lines

N start,step (Numbering)

This command generates line numbers, and is thus useful for BASIC source programs. The first number will be the specified "start" value, and the "step" value will be added for each subsequent line. A typical command would be --

N 100,10

-- resulting in line numbers of 100, 110, 120, and so on. The comma and the "step" value may be omitted; 10 is assumed. In any case, the highest number provided is 32767, after which only the character "?" appears.

Normally, pressing <RETURN> in Text Mode simply inserts a blank line, with the cursor at the first tab stop. However, once an N command has been entered, pressing <RETURN> in Text Mode inserts a line with the next available number left-justified at the first tab stop, and with the cursor at the second stop.

To insert a completely blank line even when an N comand is active, <F6> may be used.

An N command with no values will halt the generation of line numbers.

Repeating a Line

RP n (Repeat Line)

This command generates a set of identical lines. It is particularly useful for assembly language source programs.

When an RP command is entered, the Editor locates the line currently indicated by the Text Cursor, even though that cursor is not visible in Command Mode. Everything below that line is scrolled down "n" lines. Exactly "n" repetitions of the entire line are then created, filling the space just opened up.

The Editor remains in Command Mode, and the Text Cursor remains on its current line.

Lowercase Input

LO	(Allow lowercase input)
UP	(Force uppercase input)

When the Editor begins execution, all keyboard input is forced into uppercase -- that is, every small letter from "a" to "z" is automatically shifted into a capital as it goes onto the screen.

After an LO command, the Editor accepts Text Mode keyboard input exactly as entered, and a <SHIFT> key must be used to generate capital letters. The LO command is thus useful when assembly language, Data-Talk, or BASIC comments and string literals in lowercase are desired.

The UP command restores the automatic shifting of letters in Text Mode.

Blank Lines

BO	(Allow blank lines in output)
NB	(No blank lines in output)

If a completely blank line scrolls up and off the screen, the Editor normally ignores that line, and does not transfer it to the output file. This feature is a convenience for assembly language programmers; ASM considers a blank line to be an error.

After a BO command is entered, a blank line is handled like any other line. After NB, the normal suppression of all blank lines is restored.

Blank lines above the first non-blank line are never transferred to the output file. When a Q or EX command is used to complete the output by "scrolling through" the input, blank lines below the last non-blank line are always dropped too.

The Editor automatically imposes BO status if it encounters at least one completely blank line in the current input file; an assumption is made that if the input was created with BO in effect (or by a program other than the Editor), then the output should match.

STRING PROCESSING

FS	(Find String)
IS	(Insert String)
AI	(Automatic Insertion)

These commands are used to "activate" or set up strings -- that is, character sequences -- to be found, inserted, or replaced by the subsequent use of certain Function Keys.

The message "TYPE IN STRING" appears after an FS command is entered. The message "TYPE IN INSERT STRING" appears for an IS or an AI; these two commands share the same string.

At this point, the <CANCEL> key, or any Function Key, means "Withdraw this command." An immediate <RETURN> means "Deactivate any string set up by a previous use of this same command."

If neither of these is appropriate, then the Editor expects from 1 to 30 characters -- which may include lowercase letters if an LO command is currently in effect -- followed by a <RETURN>.

Normally, quotation marks are not required around the string, and trailing blanks are ignored. However, a matched set of double quotes or apostrophes will be needed if some trailing blanks must be part of the string. For example, if this string were used for a search --

```
the
```

-- then a match would occur starting at the letter "t" in each of these lines --

```
so the choice
when there is
another way
```

-- but this string --

```
" the "
```

-- would match only the five characters between the quotes, as in the first of the three lines above.

Apostophes or double quotes may appear as actual characters within an unquoted string. A string which starts and ends with double quotes may have one or more apostrophes within it, or vice versa. Examples:

```
isn't wrong
"don't reject"
' section "A" '
```

Searching

Once an FS string has been defined, every forward scroll started with <F2> or <F8> -- and every <F3> backward scroll too -- will stop when the Text Cursor reaches the closest occurrence of the string, or when all available lines have gone by. Meanwhile, the FS string is displayed as a reminder on line 24. As usual, <CANCEL> may be used to halt scrolling at any time.

Spot Replacement

Pressing <F8> in Text Mode means "Delete some characters starting at the current position of the Text Cursor; insert the IS string; start a forward scroll." If an FS string is currently active, the deletion covers the same length as that string; otherwise, no deletion is performed. Similarly, if no IS string is active, then nothing new is inserted before the scrolling starts.

Typically, both an FS string and an IS string are set up first. Then entering <F2> and <F8> will find and replace one occurrence of the FS string, but entering <F2> and <F2> again will simply skip over that occurrence.

Automatic Replacement

If an AI string, rather than an IS string, is currently active, then <F8> still requests deletion, insertion, and scrolling as just described. However, the Editor processes every occurrence of the FS string until there are no more input lines, or until <CANCEL> is pressed.

TEXT BLOCK MOVEMENT

A group of lines can be duplicated or moved by a procedure which scrolls them onto a secondary file, then inserts that file into the main text.

First, enter an R (not a M) command for the text file. Then scroll until the first line of the desired group is at the top of the screen. Enter a W command, giving some arbitrary name for the secondary file. Use <F1> to scroll just the needed lines off the screen and onto the secondary file. Finally, enter a CO command.

If another group is to be added to the secondary file, position it at the top of the screen, enter an A command for the same secondary file, scroll off the lines, and enter a CO. Repeat this process as needed.

When the secondary file is complete, enter a CI. After an M command, superfluous main text lines can be deleted, and the secondary file can be inserted wherever it's needed with FI. Later, the secondary file can be deleted with a DE command.

MESSAGES

All error and status messages are collected here in alphabetical order. The value shown as "nn" is a decimal number, as listed in the Error Messages Manual.

ATTRIBUTES PROHIBIT REQUEST	A file could not be opened, shared, modified, deleted, or renamed as requested.
BAD FUNCTION KEY	General purpose message.
COMMAND ERROR	Superfluous field in System II command requesting EDIT.
COMMAND LINE ERROR	General purpose message.
DISK ERROR, nn	General purpose message.
DISK OVERFLOW	No disk space left for lines scrolling off the screen. Space must be freed by deleting some obsolete files.
EDITOR (Rn.n) yy/mm/dd	EDIT execution has just started.
EDITOR TERMINATED	EX or AB command just entered.
ERASE KEY, HIT AGAIN TO ERASE	If the next key pressed is another <ERASE>, all of the Text Window will be deleted.
ERROR ON INPUT FILE, nn	General purpose message.
FILE ALREADY EXISTS	Command must name a new file.
FILE DOESN'T EXIST	Command must name an old file.
FILE ERROR, nn	General purpose message.
FILE IN USE, CAN'T MODIFY	M command is not acceptable now.
ILLEGAL COMMAND IN MODIFY MODE	M command is in effect now.
INSUFFICIENT CORE	Work area (around 200 words) could not be obtained.
LINE TOO LONG	Input file line was longer than 80 characters, and was trimmed.
NO MORE INPUT	There are no lines left to scroll up.

NO OUTPUT FILE	Warning. Lines scrolled up and off will be lost. The command which caused this message may be repeated.
NO TEMP FILE	A temporary file (xxxx.TP or xxxx.TM) could not be opened.
TEMP FILE ERROR, nn	General purpose message.
TYPE IN INSERT STRING	IS or AI command just entered.
TYPE IN STRING	FS command just entered.

EXEC -- Execute a Command File

EXEC reads and executes a previously created file of System II command lines. Each command is displayed as it is retrieved from the file. If a program executed in this way detaches itself from the terminal, the next command is executed immediately.

Command Line Format

```
EXEC cmdfn
```

cmdfn is the command file's name. A sequential file, with lines no longer than 78 characters each, is required.

Control returns to System II when the end of the command file is reached, or when a program terminates abnormally -- <CANCEL> key, KILL command from another terminal, or fatal error. However, such trivial errors as a unacceptable CHATR (system error code 34) or DELETE (code 36) simply go on to the next command line in the file.

The command file may include PAUSE commands, which display messages and wait the <RETURN> key. Command lines starting with "*" or "!" are recognized as comments; they are displayed without pausing.

An EXEC command may itself appear within a command file, thus switching control to some other file. Command processing will never return to the first file; on the contrary, it is closed once an EXEC command is encountered.

At bootstrap time, CRT1 automatically behaves as if this fixed command had been entered:

```
$EXEC $SYSICF
```

For details, see the section headed "Initial Command File" in Chapter 1.

FCMP -- File Compare

FCMP compares the contents of two sequential files -- presumably two files which are expected to be absolutely identical. The results of the comparison are displayed and optionally listed.

FCMP terminates when both files are exhausted, or after twenty mismatching two-character words are found. A summary then appears -- either "FILES EQUAL" or "FILES UNEQUAL."

Command Line Format

```
FCMP file1 file2 [L=fn[/A]]
```

file1	is an existing sequential file.
file2	is another existing sequential file.
L=fn	specifies a file or device for the FCMP listing.
/A	appends output if the listing file already exists.

Example

```
FCMP ALPHA BETA L=DPR1
```

This command compares a file named ALPHA against one named BETA, both on the nominal disk. The results of the comparison are directed to a character printer called DPR1, as reproduced here. Note that unprintable characters in the ASCII column are represented by periods.

```
FCMP ALPHA BETA L=DPR1
WORD  0012 IN SECTOR  115F OF FILE   ALPHA .
WORD  0012 IN SECTOR  17ED OF FILE   BETA  .

HX I   DC I        HEX             DEC          ASCII

0012      18    2030  2031      8240    8241     0   1
0013      19    393A  303A     14650   12346    9:  0:
0014      20    3437  3031     13367   12337    47  01
0030      48    2035  2036      8245    8246    .5  .6
0068     104    2031  2032      8241    8242    .1  .2
0175     373    2022  2027      8226    8231    ."  .'
017D     381    4C22  4C27     19490   19495    L"  L'

* FILES UNEQUAL *
```

Messages

WORD xxxx IN SECTOR yyyy OF FILE ffffff.ee

For only the first mismatch found, FCMP displays a pair of lines in this form. The first line refers to file1; the second to file2. The message means that word 'xxxx' (hex offset from the beginning of the sector) of logical disk sector (logical address) 'yyyy' of file 'ffffff.ee' does not match the corresponding word in the other file. This data facilitates the use of DSPLAY for further investigation.

```
   HX I  DC I       HEX             DEC          ASCII
!  hhhh  dddd   hhhh   hhhh    dddd   dddd       cc cc
```

For the first 20 mismatches found, FCMP displays lines of this form. The headings "HX I" and "DC I" give the "index" of each
! mismatched pair of words, in hex and in decimal, from the start
! of each file. The next three headings give the mismatched words
! as hex, decimal, and ASCII values -- from file1 first, then from
! file2, for each pair. Asterisks indicate that one file still had
! data remaining when the other file was exhausted.

! DEVICE INPUT NOT ALLOWED

! Each input file must be a specific disk file, not a whole disk
! or some other device type.

! ILLEGAL HASH FILE USAGE

! Only sequential disk files may be specified.

! HIGH-TO-LOW STACK OVERFLOW

! A software failure occurred within FCMP. The integrity of other
! programs and data is now doubtful; the system should be reloaded
! as soon as possible.

! INDEX ROLLOVER

! 65535 words have been compared; FCMP's counter has been reset
! to zero; normal processing continues. Subsequent entries in
! the "HX I" and "DE I" columns will reflect this adjustment.

FILES -- File Status Inquiry

FILES displays one line of information for each active file or device on the system, excluding terminals. The entire list is updated approximately once per second until FILES is deliberately terminated with the <CANCEL> key.

Command Line Format

```
FILES
```

The display has the following headings; all counts are in hex.

FILENAME	The directory name of an open disk file.
DEVICE	The name of a device currently in use.
TYPE	S (sequential file); H (hash file); D (device).
USE	R (read); W (write).
USERS	Count of current users.
STATUS	Hardware status bits (see Error Messages manual).
SECTORS	Count of disk file sectors already allocated.
CHRS/SECT	Count of characters in each full sector.
CHRS/LAST	Count of characters in last, possibly short, sector.

FLOPIZ -- Initialize Floppy Disk

FLOPIZ is a special-purpose utility, available only on the J500, for initializing a floppy disk without formatting it for System II usage. Normally, the FORMAT utility, described elsewhere in this manual, is used to prepare a new or recycled floppy disk for such usage; FLOPIZ processing alone is not sufficient.

FLOPIZ writes track headers and trailers, sector headers and trailers, and the Cyclic Redundancy Check (CRC) fields required by the disk hardware. It is thus possible for a factory-fresh floppy which lacks "soft sectors" to be made "3740-compatible." A run-time option of the FORMAT utility can also accomplish this.

Command Line Format

FLOPIZ[/R] device

/R causes FLOPIZ to repeatedly execute the command line, after pausing for a change of floppies, until canceled.

device is the name of the floppy drive which contains the disk to be processed -- FP01 for example.

Normal Messages

KEY "RETURN" TO PROCEED.

As a safety feature, this message demands an operator response before proceeding. The <CANCEL> key will abort FLOPIZ, either here or at any later point in processing.

CYLINDER: nnnn

Here, "nnnn" is a decimal number. This message is updated continually to indicate the cylinder most recently completed.

OF BAD TRACKS: nnnn

This message reports the number of tracks on which initialization failed.

Error Messages

J500 REQUIRED

FLOPIZ can't be executed on a J100, because that computer's hardware requires that a floppy already have soft sectors.

STATUS CYLINDER HEAD: x s c h

On completion of the attempt to initialize Cylinder "c" at Head "h" (both decimal), the disk hardware status was hex "x" and the software status was hex "s" -- indicating that an error occurred during the operation.

FORMAT -- Format a Disk

FORMAT prepares a rigid or floppy disk for System II usage. This operation must be performed before a factory-fresh disk can be accessed by any other program (except for the utilities DSKOPY and MTUTIL). FORMAT can also verify a disk's readability, or replace a system bootstrap, or delete all existing files.

Command Line Format

```
FORMAT[switches]  device:volid  [L=fn] [J=n] [C=addr]
```

FORMAT normally creates bootstrap sectors, empty catalog sectors, a "bit map" of free sectors, and a dummy filename for the disk's "Volume Identification" (or "Volid"). Then it writes zeroes and checksums into each remaining sector, and verifies -- reads back -- the entire disk. Optional command switches allow many of these operations to be controlled separately:

/B Bootstrap only. An appropriate bootstrap program file is copied into the bootstrap sectors of the disk being processed. No other writing or verification is done.

/C Clear only. New bootstrap, catalog, bit map, and Volid sectors are written. As a result, all previous files are deleted, regardless of attributes. No other writing or verification is performed.

/I Initialize; can be used only on a J500. The FLOPIZ utility is executed before any other processing requested. This switch is necessary if the disk is a factory-fresh floppy which does not already have "soft sectors" -- that is, if it has never been made "3740-compatible."

/N No verification pass. This switch saves time, but lessens the reliability of the result.

/V Verification only; no writing. This switch may be used to check a disk which contains active files, but which is thought to be unreliable.

/R Repeat. FORMAT normally processes just one disk, then terminates execution. The /R switch causes repeated execution of the command line, with pauses for changing disks. (No REMOVE command is needed.) This feature is useful for software librarians who must format or clear many disks successively.

Of the following command line fields, the first two are always required; the rest are optional.

device Name of the drive containing the disk to be formatted -- for example, DP01, DD01, or FP01. Must be separated from "volid" by a colon or semicolon, not by a space.

volid Name for the dummy (zero-length) file which becomes the disk's Volume Identification. This name will appear first in any unsorted catalog listing. Like any filename, a Volid must start with a letter, followed by 0 to 5 letters and numerals. An example, used for AM Jacquard software distribution, is R70DAT ("Release 7.0 Data-Rite").

L=fn Filename for FORMAT error listings. This field allows screen messages to be copied to any file or device, except the disk being formatted. If an existing disk file is specified, the listing is appended to it.

J=n The type of bootstrap to be recorded. "J=1" indicates a J100; "J=5" indicates a J500. This field allows a J500 to format a system disk which will actually be loaded on a J100, or vice versa. If the "J=n" field is omitted, processing automatically matches the type of Videocomputer being used to run FORMAT.

C=addr The refresh address (in hexadecimal) for the Master CRT (device X'28). This address is recorded as part of the bootstrap. The default is DC00. For a CRT with a universal keyboard (Model 4800), the true refresh address is always used at bootstrap time, and the recorded address is ignored. If the Master CRT has a Model 3000 keyboard, and if its refresh address is not DC00, then "C=addr" will be needed for bootstrap message displays.

FLOPIZ Messages

If the /I switch was used, any of the messages described for the FLOPIZ utility may appear during an execution of FORMAT.

Normal FORMAT Messages

KEY "RETURN" TO PROCEED.

As a safety feature, this message demands an operator response before proceeeding. The <CANCEL> key will abort processing at this point.

WRITE FORMAT PASS (or) READ TO VERIFY PASS

Displayed just before the indicated pass.

CYLINDER: n

This message is updated continually to indicate the decimal number of the cylinder most recently processed.

BOOT WRITTEN

A bootstrap program file was obtained from the current primary disk, and written into the bootstrap area of the disk being processed. This message appears only for the /B switch, although a new bootstrap is also written if the /C switch, or no switch at all, is used on the command line. The specific program file is selected according to the disk and computer types involved, subject to the "J=n" command field.

Program:	Computer; device:
FOBD1.SB	J100; BDxx
FOD31.SB	J100; DAxx, DDxx, DPxx
FOFDU.SB	J500; DAxx, DDxx, DPxx, FAxx, FDxx
FOFP1.SB	J100; FPxx
FOFPU.SB	J500; FPxx

Unusable Sector Messages

OF UNUSABLE SECTORS: nnn

Displayed when all operations on one disk have been completed. Here, "nnn" is a decimal number. Each unusable sector is marked as "used" in the bit map, and becomes unavailable for future allocation to a file.

If the /V switch (verification only) was used, "nnn" may not agree with the total number of appearances of the next message.

UNUSABLE SECTOR: xxxx

Logical address "xxxx" is unusable. It is marked as allocated in the bit map.

UNUSABLE MEDIUM, CRITICAL SECTOR UNUSABLE: xxxx
UNUSABLE MEDIUM, BIT MAP SECTOR UNUSABLE: xxxx

Logical address "xxxx" is unusable, and because it must be used for the bootstrap, bit map, or catalog, the disk itself is unusable.

The system error message immediately preceding any of the last three messages will appear only on FORMAT's own terminal (not on any "L=" file), and will include the software and hardware status associated with the attempted access.

Fatal Error Messages

ERROR - SPECIFY DEVICE:NAME

The command line omitted the required field consisting of a device name and a Volume Identification.

J= 1 OR 5

The command line field "J=n" is not acceptable.

CANCEL - MEDIUM UNUSABLE

FORMAT was canceled before it could write the bootstrap, the bit map, and the directory. The disk is left in an unusable state, and must be formatted again.

CANCEL - MEDIUM USABLE

FORMAT was canceled after it wrote the bootstrap, the bit map, and the empty catalog. If the rest of the disk was formatted previously, it is still usable.

PROGRAM NOT FOUND: $FLOPIZ

The /I switch was used, but the FLOPIZ utility was not available on the current primary disk.

PROGRAM NOT FOUND: $name.SB

The required bootstrap file was not available on the current primary disk.

GTAD -- Time and Date Inquiry

GTAD displays the current System Date (in the format YY/MM/DD) and the System Time (HH:MM:SS, on a 24-hour clock). The utility STAD may then be used to force new values as needed.

Command Line Format

```
GTAD
```

When System II is loaded, the date is set to that specified by SYSGEN, and the time is set to midnight (00:00:00). The date is advanced at each subsequent midnight, but the system assumes that every month has 31 days.

Note that every Videocomputer programming language has some means of obtaining date and time information internally.

HCREAT -- Hash File Creation

HCREAT creates a directory entry, and allocates index space, for a hash (keyed) file. The permanent attributes HF, RS, and WS are given to the entry. Optionally, the file may also permit duplicate keys, and it may be a specially-organized "Batch File" (typically, for a Data-Rite application).

Command Line Format

```
HCREAT fn[/D][/B] K=k [S=s] [F=f]
```

fn	is the name of the file to be created.
/D	allows the later creation of records with duplicate keys.
/B	indicates a Batch File.
K=k	is the key length, in two-character words.
S=s	is the number of sectors to be allocated for the index.
F=f	is a decimal number -- the maximum number of records expected in the file. Either "S=" or "F=" must appear.

In most cases, the user knows the approximate number of records a hash file must hold. The "F=" field, rather than the "S=" field, is therefore more practical.

When "F=" is chosen, HCREAT computes the number of index sectors such that an even distribution of record keys will result in approximately two records per node.

Examples

```
HCREAT HFIL K=10 S=2
```

This command writes a directory entry on the nominal disk for a file named HFIL. The key length for each record is 10 words (20 characters), and each key must be unique. Two disk sectors are allocated for the file index.

```
HCREAT FP00:HFIL/D K=15 F=10000
```

This command writes a directory entry on FP00 for a file named HFIL. Duplicate keys are permitted. Each key is 15 words (30 characters) in length. The file is estimated to require 10000 records. HCREAT will allocate the index sectors, based on the device's sector size, the "F=" value, and the "K=" value.

HDELT -- Hash File Deletion

HDELT deletes a hash file. All data records are deleted, and the disk space is freed. The directory entry itself may be deleted or retained as needed. To delete a sequential file, the DELETE utility must be executed. To delete all the files on a disk, use the /C option of FORMAT.

Command Line Format

```
HDELT   fn[/C] [fn[/C]]...
```

fn — is a hash file to be deleted. A series of names may be specified.

/C — indicates that all data records are to be deleted, but that the file's directory entry and index space are to retained for future use.

Error Processing

If a filename is unacceptable, HDELT displays the command line with a caret under the first name rejected. Only the files before that point have been deleted.

Example

```
HDELT   ALPHA/C $BETA
```

This command clears the data records from hash file ALPHA on the nominal disk, but retains the directory entry and the cleared index sectors. BETA, on the primary disk, is completely erased.

HKOPY -- Hash File Copy

HKOPY can transfer data between an existing hash file and a specially organized sequential backup file. Transfer from one hash file to another is also possible, if the two files have the same key length, and if they agree with regard to the presence or absence of "duplicate key" and "batch file" attributes.

THIS PROGRAM CAN CAUSE A SYSTEM FAILURE; see "P=w" below.

Command Format

```
HKOPY in[/R][/B] out[/X][/B] [R=x:y] [E=fn] [P=w]
```

in — Input file, always required.

out — Output file, always required unless /R is used.

/R — "Read Only." No output file is specified. HKOPY reads through the input file, which may be either hash or backup, for the sole purpose of obtaining file statistics, as described under "Normal Messages" below.

/X — "Extend Output." Allows data to be appended to an existing hash output file.

/B — "Backup." For input, /B indicates a sequential backup file, created by a previous HKOPY. For output, /B indicates that the output file should be a sequential backup, not another hash file. /B can't be used for both input and output.

R=x:y — "Range." This optional field specifies the minimum (x) and maximum (y) acceptable length (in decimal words) for an input record. Any record not fitting the range is reported; HKOPY proceeds to the next hash file node.

E=fn — "Error File." A device, or a disk file not already in existence, for recording all error displays.

P=w — "Partition size," in decimal words. Used with /B. HKOPY allocates this partition for I/O of backup file records. If /B is present, but P=w is absent, P=1008 is assumed. P must be at least 2 words longer than the hash file's longest record. Otherwise, HASH INPUT RANDOMLY ALTERS MEMORY, TYPICALLY CAUSING A SYSTEM FAILURE MESSAGE OF "PCER" after HKOPY terminates with an error.

Like COPY, the HKOPY utility has three modes of operation -- process and terminate; process and return to user; indirect command file. Refer to the description of COPY for details on the command line for each mode.

Examples

```
HKOPY HFIL SFIL/B
```

The hash file HFIL is backed up on the sequential file SFIL. A header record is written first, preserving the input file's name, attributes, and backup date. DSPLAY or PCRT can later to be used to examine this information.

```
HKOPY SFIL/B HFIL/X
```

The backup file SFIL is used to extend the existing hash file HFIL. The attributes preserved in SFIL's header must match those for HFIL.

```
HKOPY HFIL/R
```

HFIL is read through for statistics; no output file is involved.

Space Recovery

When an application program deletes records from a hash file, the disk space involved remains allocated to the file. However, this space is automatically recovered by HKOPY, which transfers only non-deleted records. The following sequence of commands will purge HFIL of deleted record allocations.

```
HKOPY HFIL SFIL/B
HDELT/C HFIL
HKOPY SFIL/B HFIL/X
```

Normal Messages

r LOGICAL RECORDS b BYTES. MAX RECORD SIZE w WORDS.

This message is continually updated as HKOPY reads the input file. The running totals -- all decimal numbers -- include each record's one-word length field and all of its key. There are two characters ("bytes") in each word.

OPERATOR CANCELLED

The <CANCEL> key, or the KILL utility, forced termination.

HASH FILE COPY COMPLETED

Processing terminated normally.

Error Messages

Except for the first message below, HKOPY will proceed to its next command, if any, after the error is reported.

ERROR IN NODE I=sector O=offset

An input hash file had an error in the node anchored as shown. HKOPY proceeds to the next node. If the immediately preceding message indicates a System Error Code 8, a disk hardware error occurred. A data sector is usually involved, but if the offset is for the last node in an index sector, the error may be in the index sector itself.

RECORD SIZE OUT OF BOUNDS length

The "R=x:y" command field made this record unacceptable.

I/O PARTITION TOO SMALL

During sequential input, the "P=w" command field was too small. A "/R" execution may be used to determine the required value.

I/O PARTITION EXCEEDED

During hash input, the "P=w" command field was too small. System II's memory has been altered unpredictably. Cancel HKOPY, terminate all other jobs currently active, and reload System II. Use the DELETE utility for the backup file which was being created. Run HKOPY/R to determine the partition size required.

ERROR: LINKED HASH BUFFERS

A system software failure has occurred. Reload the system, and delete any output files generated by the job.

ATTRIBUTE MISMATCH: reason

The input and output files do not have matching attributes or key lengths. For example, duplicate keys must be either allowed or prohibited on both files.

ILLEGAL FILE TYPE
INFILE (or OUTFILE) NOT HASH
OUTFILE NOT SPECIFIED
OUTFILE ALREADY EXISTS
INFILE ISN'T HASH FILE BACKUP
BACKUP FILE CAN'T BE HASH

These self-explanatory messages usually indicate a trivial error in the command line, such as a missing switch or field.

OUTFILE HAS DATA. EXTEND OPTION MUST BE USED.

At least one non-deleted record already exists on the output file. To append the input file, use the "/X" switch. To delete all existing records, run HDELT/C before HKOPY/X.

CAN'T NEST INDIRECT FILES

As explained under COPY, an indirect command file can't specify another indirect file.

INDIRECT FILE IS HASH

Only a sequential file may be used for indirect commands.

J5LDAB -- J500 Absolute Loader

Except for the program name itself, the procedures and options for J5LDAB are the same as those described under LDABS.

JOBS -- Job Status Inquiry

JOBS displays one line of information for each current job -- that is, for each program being executed. The entire list is updated approximately once per second until JOBS is deliberately terminated with the <CANCEL> key.

Command Line Format

```
JOBS    <RETURN>
```

The display includes these headings:

JN — Job number, assigned arbitrarily. In this column, slots are available for the maximum number of jobs specified at SYSGEN time. At least two entries will always appear -- System II itself, and the JOBS utility's own job.

DEV — Device from which a program was loaded. For System II itself, the configuration type.

PROGRAM -- filename used to execute a program.

STIME — Start time for a job.

TERM — Terminal from which a program was executed.

NOMDSK — Nominal disk associated with "TERM" entry.

BS — BSECT start address. |
BL — BSECT length. |
| --- Program memory locations.
TS — TSECT start address. |
TL — TSECT length. |

CPU — CPU seconds -- not elapsed time -- consumed since a job started.

TC — Task count. Only assembly language programs can have more than one task.

FC — File count. This includes every open device or disk file, and a job's terminal if it has one.

BUFS — Buffer words allocated, in hex.

KILL -- Terminate All Copies of a Program

KILL forces the immediate abnormal termination of a specified program. All copies of the program, -- those attached to other terminals and those running in a detached mode -- are affected. Furthermore, all copies of the program in memory are marked as unavailable for further use, so that a subsequent execution will load a fresh copy into memory.

Command Line Formats

```
KILL [device:]pgmfn
KILL $pgmfn
```

device is the specific device from which the program was loaded, and is required unless the device corresponds to the current nominal disk.

pgmfn is the filename used when the program was executed.

$pgmfn is appropriate when the program to be killed was loaded from the current primary disk.

Note that the KILL command always refers to a program in terms of the specific file from which the program was loaded for execution -- just as shown when the JOBS utility is used.

Failure to specify the correct name -- typically because the identity of the nominal disk has been changed since execution began -- will result in the error message "PROGRAM NOT FOUND" -- that is, not found to be active in memory.

It is possible for programs written in assembly language to "hold off" the effect of a KILL command, or to handle it as if the <CANCEL> key had been used. The documentation for such programs will, of course, cover this point.

LDABS -- J100 Absolute Loader

LDABS performs the loading and execution of a standalone program -- that is, a program which does not run under the control of System II. Such programs, typically, are field service aids.

LDABS is relevant only to a J100. On a J500, the utility program J5LDAB is used, with identical procedures and options.

Command Line Formats

```
LDABS[/D]  pgmfn [S=addr]
J5LDAB[/D] pgmfn [S=addr]
```

pgmfn — is the filename of the program to be loaded and (if "/D" does not appear) executed immediately. Only a file on a floppy disk is accepted.

/D — loads the System II Debugger (DEBUGX.SB) into high memory after "pgmfn" is loaded, and gives control to the Debugger. The DEBUGX.SB file must be on a floppy disk which is currently the primary disk.

S=addr — overrides any start address in the program file. A hex number is expected; a decimal prefix -- as in D'256 -- is allowed.

If there are any other jobs, or any disk activity, an appropriate error message is displayed; the command is rejected. Otherwise, a REMOVE command is implicitly issued for every disk device (that is, the SMT is flushed). This message appears:

```
SYSTEM II DISABLED
```

An I/O reset is forced on a J100. In any case, subsequent use of System II will require reloading it from disk.

The specified program is loaded with BSECT reference address 0 and TSECT reference address X'100. ASECT code is loaded as appropriate.

Abort Messages

If one of the following messages appears, System II is not disabled, and the utility is abnormally terminated.

OTHER JOBS EXECUTING

Jobs other than the utility are still executing.

DISK ACTIVE

Some disk device still has a non-zero user count.

DISK MUST BE FLOPPY

The program to be loaded must reside on a floppy drive. If the "/D" switch appears, the primary device must be a floppy drive.

ONLY /D ALLOWED

An unacceptable switch has been used in the command line.

TSECT OVERLAP

There is insufficient memory for the program being loaded.

BSECT OVERLAP

More than 256 BSECT words were required.

MUST BE INVOKED FROM A CRT

The utility can not be invoked from a remote asynch terminal; only a standard Videocomputer CRT is acceptable.

Total Abort Messages

If one of the following messages appears, the utility failed after System II was disabled. The system must be reloaded.

DISK ERROR

A disk error occurred; 5 retries could not overcome it.

WRONG BUFFER

An invalid buffer was released. This message is never seen in normal operation. It may indicate a malfunctioning memory.

WRONG SIZE

A buffer of invalid size was requested. This message is never seen in normal operation. It may indicate a malfunctioning memory.

NO START ADDRESS

The utility command had neither a "/D" switch nor an "S=" field. The specified program had no start address in its end block.

LSPY -- Line Driver Monitor

The LSPY utility monitors the information sent and received by a communications driver. Each LSPY job is tied to one specific driver from the set listed here with typical names and SYSGEN characteristics for the related devices:

BPEDRV.SB	BPE1	BSC Point-to-point, EBCDIC
BPADRV.SB	BPA1	BSC Point-to-point, ASCII
BMEDRV.SB	BME1	BSC Multipoint, EBCDIC
BMADRV.SB	BMA1	BSC Multipoint, ASCII
SILDRV.SB	COM1	General SILA Communications
UJEDRV.SB	UJE1	Univac Remote Job Entry

LSPY results may be displayed, or written to a file, or both. Each new transmission, reception, or error code is indicated with one of these lines:

```
** T **
** R **
** ERR nn **
```

A right bracket ("]") is used to mark the actual end of a line of text, so that trailing blanks may be counted if necessary.

If LSPY output in ASCII is appropriate for a driver, the utility uses whatever code translation table was associated with the device most recently opened by that driver.

After translation, a control character -- defined as any 8-bit character with a value less than hex 20 -- is shown as a space, a standard designation, and another space. For example, if the character is hex 03, then " ETX " will appear.

However, if the translated value is hex 00, then the designation " NUL " appears only if the character was also hex 00 before translation. Otherwise, the untranslated value is shown instead as a space, two hex digits, and another space.

A hex FF on the line always appears as " PAD " without regard to translation.

Command Line Format

```
LSPY[switches] [device] [L=listfile[/control]]
```

LSPY always runs in background mode, so the keyboard of the initiating terminal may be used to start another job, and the screen may be shared between that job and LSPY's rolled output. The KILL utility must be used to terminate LSPY.

A series of optional switches may be appended to the utility name, to control output formatting and the selection of a driver:

/L Listfile is only LSPY output. If this switch is omitted, the screen and any "listfile" specified are both used.

/H LSPY output in 8-bit hex only.

/M BMEDRV.SB monitored. EBCDIC output unless /H used.

/S SILDRV.SB monitored. /H always assumed.

/U UJEDRV.SB monitored. Univac XS-3 output unless /H used.

Only one driver switch -- /M, /S, or /U -- may appear. If neither a switch nor a "device" entry is given, driver BPEDRV.SB is assumed. In any case, if the relevant driver is not already in memory, LSPY will load it from the current primary disk.

If the optional "device" entry appears, any driver switch is ignored. The driver is taken to be the one associated at SYSGEN time with the specified device. However, monitoring is not restricted to that device.

The optional "L=listfile" entry names a sequential file or device for LSPY output. A printer is usually too slow to be practical; a disk file or a tape would be appropriate.

"L=listfile/A" appends output to a disk "listfile" if it already exists, and creates a new file otherwise.

"L=listfile/R" deletes and replaces a disk "listfile" if it already exists, and creates a new file otherwise.

For the following command example, it's assumed that device BPA1 has its standard SYSGEN definition. LSPY would then monitor all line activity going through driver BPADRV.SB, with output in hex to file ALPHA, but not to the screen.

```
LSPY/L/H BPA1 L=ALPHA/R
```

MONITR -- Remote Terminal Monitor

MONITR allows the initiation of a job on a remote terminal -- that is, on a device other than a J100/J105/J500. A remote terminal -- an RS-232, or Teletype-compatible, device on a single asynchronous communications line or a multiplexer line -- is considered a peripheral device by System II. As such, it cannot directly initiate a job.

MONITR, however, can be initiated at a Videocomputer CRT. This program then detaches itself, and runs in the background mode. MONITR will read System II job initiation commands entered by a user at a remote terminal. Upon receipt of a command, it initiates a separate job to execute the command. MONITR and the secondary job both use the remote terminal as the default I/O device.

Command Line Format

For each remote terminal to be monitored, the following command is entered at a Videocomputer CRT:

```
<terminal>MONITR
```

The "terminal" field -- which must be enclosed in angle brackets, as shown -- is the device name established for the terminal during system generation. Examples:

```
<ACY1>MONITR

<MCA0>MONITR
```

The MONITR program will display the following message at the corresponding terminal:

```
MONITR
ENTER CMD
```

The operator of the remote terminal may then key in a System II command, with a maximum length of 80 characters. When the <RETURN> key is pressed, MONITR passes the command to the system with a "link" call, thus initiating a new job. When the job ends, the MONITR message "ENTER CMD" appears again.

MONITR Termination

To end the execution of MONITR, the following System II command must be entered; any terminal may be used.

```
KILL MONITR
```

Note that this command will end all MONITR jobs. It is not possible to stop monitoring one remote device while continuing with another.

If a program is currently running on a remote terminal, MONITR will not end on that terminal until the program itself ends execution.

When MONITR does end execution for a given remote terminal, the following message appears at that terminal:

```
DE-ACTIVATING THIS LINE
```

Input Editing

If, at system generation, the IE (input editing) option was selected for the remote device, then various control keys may be used -- before <RETURN> is pressed -- to edit commands and program input:

!	CTRL H (BS)	Delete previous character; echoes BS
!		if EM (Echo Mode) characteristic active.
!	CTRL G (BEL)	Delete previous character; echoes "#"
!		(or BEL in column 1) if EM active.
	CTRL X (CAN)	Delete entire line.
	CTRL D (EOT)	Send end-of-file.
!	CTRL [(ESC)	Abort the last job initiated.

If an error condition occurs in the job that is executing, a standard system error message will appear at the remote terminal.

Program Restrictions

An application program must use some kind of message to prompt for operator input at a remote terminal. This is required even on a remote CRT, because the cursor is visible at all times, not just when input is expected.

Full screen I/O is not supported at a remote terminal. Neither application programs, nor standard utility programs, which
! require such I/O can be executed successfully through MONITR.

MTUTIL -- Disk/Tape Dump and Load

MTUTIL can dump, onto a magnetic tape, the complete contents of any floppy or rigid disk (except a CDC 9762, device BDnn). The utility can also load the tape onto a disk of the same type.

A single tape may contain several dump, not necessarily of the same type. Each dump is followed by one filemark (also called a tapemark, or simply an "EOF"). A double filemark indicates the end of all data. Various MTUTIL commands commands to MTUTIL allow a tape to be cataloged and searched.

Command Line Format

```
MTUTIL[/N] tape [I=fn]
```

tape — specifies a magnetic tape device, such as MT00. This device is opened immediately; if it's not not on-line and ready, MTUTIL terminates abnormally.

/N — is an optional switch, indicating that a new tape volume is mounted. MTUTIL writes two EOFs at the load point to indicate that there is, as yet, no data on the tape. The /N switch is used to initialize a factory-fresh tape or to recycle a previously used tape.

I= — is an optional filename -- a command file for MTUTIL. If this field appears, commands are read and executed until the file ends, or until an EX command (see below) is encountered in the file.

You enter keyboard commands for MTUTIL when the message "KEY CMD" is on display and the cursor is awaiting your input. An operation commences when you press <RETURN> at the end of a command line. Pressing <CANCEL> stops any current operation, terminates the program, and returns control to System II.

MTUTIL commands are described on the following pages.

DU disk [F=number]

Dump a disk onto a tape. When this command is accepted, a comment line -- from 0 to 80 characters, followed by <RETURN> -- must be supplied next from the keyboard (or from the command file). The specified disk must not be in use by any other job; once dumping starts, MTUTIL will lock out other jobs. The optional field "F=" specifies the tape file to be dumped to. Tape files are numbered from the load point -- 1, 2, 3, and so on. If the file number is not specified, the dump is made beginning at the previous end of data (a double EOF); a new double EOF EOF is written at the end of the dump file. Hence, there is never any recoverable data beyond the most recently written dump file. A disk error during dumping is reported on the screen, but does not stop the dump.

PO [F=number]

Positions the tape. The F= field is an option allowing you to specify a tape file number. The default is file number 1. When MTUTIL has reached the position specified, the identification record of the file is displayed.

LO disk1 [disk2] [disk3] [disk4]

Loading a tape file onto as many as four disks. The tape must be at the beginning of a file when load commences; hence, a PO command must precede each LO command. A disk must not be in use and, must not be the current primary disk. During the load, each disk is locked to prevent use by any other job. MTUTIL recalculates and verifies the checksum word in each tape block being loaded; no other data checks are made.

VE disk

Verifies a disk to ensure that it can be read after a load from a tape. Each sector of the disk is read, and its checksum word checked.

LI [L=fn]

Lists, on file or device "fn" -- typically a printer -- the identification record of each file on the tape, as described in the next section. If the "L=" field is omitted, then "L=LPT1" is assumed. If a disk file is specified for the listing, any previous version of the file is deleted automatically.

EX

Exits (terminates) MTUTIL and returns control System II.

Tape Format

Each disk image is one file on the magnetic tape. Each tape file begins with an identification record -- system time and date of tape file creation; disk type; DU command comment line; control information used by MTUTIL.

The rest of the file contains the physical disk image, blocked one logical disk track per tape record. Each tape record contains a checksum word that is verified during tape input. The logical sector address precedes each sector.

The sequence of disk sectors accessed has been calculated to minimize latency delays. This provides the fastest possible dumping and loading. If there are other jobs competing for CPU time, this strategy will be defeated, and dramatic increases in execution time will result.

Sufficient partition space must be available for MTUTIL and for the tape driver MT9DRV.SB -- about 2000 decimal words each. An I/O buffer is also required -- 1691 words for a floppy disk; 6169 words for a cartridge.

Nominal dump and load times in minutes and seconds, assuming that no other jobs are running:

0:75	Floppy
4:55	Pertec D3341
9:40	Pertec D3441, 256-word sectors
7:20	Pertec D3441, 512-word sectors

Length of tape consumed: floppy, 35 feet; Pertec D3341, 275 feet; Pertec D3441, 550 feet. A seven-inch tape reel conveniently holds one Pertec D3441 (DDnn or DAnn) disk image, two D3341 (DPnn) images, or up to 17 floppy images. However, search times for floppy images near the hub of a tape reel may be inconveniently long.

Fatal Errors

CRT NOT OPEN

Open, using task page zero as the FNB, failed. This would happen if a background job linked to MTUTIL.

TAPE OPEN ERROR

The tape device specified could not be opened.

UNIT IS NOT A TAPE

Command line did not correctly specify a magnetic tape drive.

Non-Fatal Errors

SKIP FILE ERROR

System call MTSFF took its error return. Position the tape to the load point and try again.

FILE DOES NOT EXIST

The double EOF marking the end of data on the tape was encountered before the specified file number. Does the file really exist? Is it on another tape? Did you make a mistake with the file number in the command line?

TAPE READ ERROR

System call MTRD took its error return. This message is followed by the "STATUS= xxxx" message giving the magnetic tape status. This message may indicate hardware trouble or a bad tape.

DISK UNAVAILABLE or ACTIVE FILES ON DISK

The disk specified is locked and can't be accessed by MTUTIL at the moment. Investigate other system activity (perhaps with the FILES and JOBS utilities, then try again.

DISK NOT SPECIFIED or UNIT IS NOT A DISK

A command of LO, DU, or VE failed to specify a disk.

NEED POSITION BEFORE LOAD

A PO command is required before each LO command.

DISK TYPES DON'T MATCH

LO command specified a disk of a different type from that recorded on the tape. For example, you can't load a floppy image onto a cartridge disk.

CAN'T LOAD PRIMARY DISK

MTUTIL will not allow you to name the current primary disk in a command of LO. If you really meant to write to that disk, and if you are running a version of System II without overlays, then exit from MTUTIL, use the CHPRI utility, and run MTUTIL again.

NO PARTITION SPACE

Each time LO or DU is entered, a partition is requested, of a size appropriate to the disk specified. Check other system activity, and try again.

SYNTAX ERROR

The command given to MTUTIL could not be recognized.

TAPE REWIND ERROR

System call MTRWD took its error return. Use the tape drive's own controls to rewind to load point; try the command again.

TAPE WRITE [EOF] ERROR

System call MTWD (or MTWEF) took its error return. A DU command did not terminate properly. Reposition the tape and try again.

BAD FILE #

A command of PO specified a number less than 1.

DISK TYPE NOT ALLOWED

MTUTIL only handles disk types whose standard names are of the forms FPnn, DPnn, DDnn, and DAnn.

TOO MANY DEVICES SPECIFIED

Only four disks may be specified for LO.

CREATE ERROR (or OPEN, CLOSE, DELETE ERROR)

Processing of the "L=" field, in a command of LI, was not completed properly. Did you name a permanent disk file?

TAPE VOLUME WRITE PROTECTED

As always, "No Ring, No Write." Is this the correct reel?

TAPE CHECKSUM ERROR

A block of data was read from the tape, a checksum was calculated for it, and this checksum was compared with the checksum included in the block of data. These two checksums differed. This is a software checksum, not a hardware checksum. There is one checksum per block. A block contains a whole logical disk track. This message may indicate a serious problem.

DSK dddd LA: aaaa SW STATUS: ssss HW STATUS: hhhh

A read or write of logical sector "aaaa" (hex) encountered a disk error. The software status was "ssss" and the hardware status was "hhhh" (hex). This message may indicate a serious problem.

STATUS: ssss

A magnetic tape operation encountered a tape error. The software status was "ssss." This message may indicate a serious problem.

PARTS -- Partition Inquiry

PARTS can display (or print) data about each partition currently allocated in memory. The display is continually updated, approximately once per second, until PARTS is deliberately terminated with the <CANCEL> key.

Command Line Format

```
PARTS [L=fn[/A]] [S=z] [J=x:[y]]
```

L=fn — specifies a disk file or a printer. PARTS opens "fn" to record the list just once; the file is then closed.

/A — appends the listing to an existing "fn" disk file, rather than deleting and replacing it.

S=z — specifies the partition number at which the list starts. Default: S=1.

J=x:y
J=x — specifies a range of job numbers from "x" to "y" (or just "x" alone). Partitions are then sorted out and listed by related jobs, as indicated on a line above the column headings. Can't be used if "S=z" appears. A printed listing may be necessary; on the screen, job "x" is replaced by job "y" too quickly to be visible.

Columns on Listing

NBR	Partition number, assigned arbitrarily.
PROGRAM FILENAME	Two columns. Program name appears second. First entry is device loaded from, or indication of special usage (such as Type-Rite's shared messages and printer queues), or System II configuration (Cn), or job which requested this partition (Jn).
TSECT ADDR TSECT LENTH	Start and length of Top Sector, in hex words.
BSECT ADDR BSECT LENTH	Start and length of Bottom Sector, in hex words.
START ADDR	Execution start address, or 0000.
NBR USERS	Number of users sharing this partition, in hex.
PROG FLAGS	Program flags (4000 = shared; 2000 = being loaded).

If there are too many partitions to fit on the screen, the word "MORE" will blink in the upper lefthand corner. Another execution of PARTS will be needed, with "S=z" covering the rest of the partitions.

Example

```
PARTS J=1:3 L=DPR1
```

This command prints separate lists of the partitions allocated to jobs 1, 2, and 3. A form feed is issued after each list, as represented here by a line of hyphens. The date in each heading is the creation date of PARTS, not the current date.

```
                P A R T I T I O N S  -  7 8 / 0 6 / 1 2
                     J O B   N U M B E R      1
        PROGRAM                TSECT TSECT BSECT BSECT START NBR   PROG
NBR     FILENAME               ADDR  LENTH ADDR  LENTH ADDR  USERS FLAGS
 1    (C3) SYS R70             0100  30E0  0000  0010  0100  0001  0000

-------------------------------------------------------------------------

                P A R T I T I O N S  -  7 8 / 0 6 / 1 2
                     J O B   N U M B E R      2
        PROGRAM                TSECT TSECT BSECT BSECT START NBR   PROG
NBR     FILENAME               ADDR  LENTH ADDR  LENTH ADDR  USERS FLAGS
 2    BD00  TYPIV              31F0  21A0  0010  0009  4790  0006  4000
 3    (J02) TYPIV              53A0  01D0  0000  0000  0000  0001  0000
 7    ******TYPIVMES           5B60  01D0  0000  0000  0000  0006  4000

-------------------------------------------------------------------------

                P A R T I T I O N S  -  7 8 / 0 6 / 1 2
                     J O B   N U M B E R      3
        PROGRAM                TSECT TSECT BSECT BSECT START NBR   PROG
NBR     FILENAME               ADDR  LENTH ADDR  LENTH ADDR  USERS FLAGS
 2    BD00  TYPIV              31F0  21A0  0010  0009  4790  0006  4000
 5    (J03) TYPIV              5580  01D0  0000  0000  0000  0001  0000
 7    ******TYPIVMES           5B60  01D0  0000  0000  0000  0006  4000
-------------------------------------------------------------------------
```

PAUSE -- Pause in Command File

A PAUSE request is intended for use in an EXEC utility command file. It suspends execution of the command file, displays this message --

DEPRESS THE RETURN KEY TO CONTINUE

-- and waits for the operator's response. Pressing the <CANCEL> key in this situation will abort further processing of the current command file, and return control to the keyboard.

Any fields on the command line are ignored, and may be used to put comments or operator instructions on the screen.

Command Line Format

```
PAUSE [comments, through column 78]
```

Example

The following sequence of lines might appear at the end of an EXEC file:

```
!
!  Operator --
!                To delete the old files, press <RETURN>.
!                To save them, press <CANCEL>.
!
PAUSE
DELETE ALPHA BETA GAMMA
!
!  End of command file
```

PCRT -- "Print" on Screen

PCRT displays a sequential file of printable data, always starting at the first file record. PCRT is thus a convenient way to inspect a file without going through EDIT or Type-Rite.

Pressing <CANCEL> once will temporarily halt the rolling of the display. Pressing <RETURN> starts the roll again.

Pressing <CANCEL> twice -- or reaching end-of-file -- will terminate the display and return control to System II.

Command Line Format

```
PCRT[/L] fn
```

fn — is the name of the sequential file to be displayed.

/L — controls the display of long file records. With /L, the entire record is displayed in a series of 80-character lines. Without /L, only the first 80 characters are displayed.

If you accidentally specify a file name that is not, in fact, a sequential file of printable data, one of two things will happen.

For a hash file, system error message 84 will appear. The DSPLAY utility, not PCRT, must be used for hash files.

For a sequential file containing binary (non-printable) data, meaningless characters will fill the screen until you press <CANCEL> twice, or reach the end of the file.

Some of these characters may start or stop such screen features as blinking, underlining, intensity changes, and reverse video. When the cursor reappears, press the <ERASE> or <RULER/PRINT> key to clear the screen. Again, the DSPLAY utility, not PCRT, must be used.

PRINT -- Print a File

PRINT is used to transfer the contents of a sequential file to an on-line printer. The file must contain only print-image information. If a hex dump is needed, the DSPLAY utility must be used. If the file contains Type-Rite print controls (such as underscore markers, or lines starting with "\"), then only Type-Rite itself can be used, unless the intention is to print the controls verbatim.

PRINT runs as a detached job, and therefore does not tie up the initiating terminal while the file is being printed (or is waiting for the printer to become available).

If the specified printer is already being used by a previous execution of the PRINT utility (not by some other program), then further printing is queued -- that is, put on a waiting list -- automatically, and proceeds as soon as possible. A command line switch (/K), described on the next page, can remove a file from a PRINT queue, or cancel an entire queue.

Command Format for Printing

```
PRINT[/R] [fn] [out]
```

fn — is the name of the disk file to be printed. It is immediately opened for non-sharable reading, and remains in that status until printing has been completed. System error message 39 appears if this is impossible because the file is already in use.

out — is the printer to be used. Another device, or even a disk file, may be used to meet some special requirement. In any case, the PRINT command is rejected (with an error code of 39) if "out" is already in use by any program except a previous execution of PRINT.

If "out" is an existing disk file, PRINT output is appended to it.

If this field is omitted, LPT1 is assumed. Note that if LPT1 was not defined during system generation, a file named LPT1, on the nominal disk, is implied.

/R — means "remove" -- that is, delete -- "fn" once it has been printed completely.

Command Format for Queue Control

```
PRINT/K [fn] [out]
```

/K indicates that the command is intended to "kill" -- that is, to cancel -- a previously queued PRINT request.

fn restricts the "kill" to one specific file. If this field is omitted, the entire queue for the "out" device is cancelled immediately.

out identifies the queue to be affected. If this field is omitted, then LPT1 is assumed.

Examples

Assume that a system has two character printers, DPR1 and DH04. The operator puts three requests on the PRINT queue for DPR1:

```
PRINT ALPHA DPR1
PRINT BETA DPR1
PRINT GAMMA DPR1
```

Before file BETA is actually printed, it is determined that DH04 should have been used instead. These commands would be used:

```
PRINT/K BETA DPR1
PRINT BETA DH04
```

REMOVE -- Remove a Disk

REMOVE is used to warn the system that a floppy disk, or a removable rigid disk, is about to be taken out of its drive. It is strongly recommended that every Videocomputer operator acquire the habit of entering this command every time that a disk is physically removed from its drive.

If an operator "deceives" the system by switching disks without a REMOVE command, the integrity of both the old disk and the new one will be doubtful.

Command Line Format

```
REMOVE device
```

device is the drive from which a disk is about to be removed, such as FP01 or DP00.

When a REMOVE command is entered, the system checks these points:

Is the disk still in use by some other job or task?

Have any sectors from the disk been brought into system memory, in anticipation of an input request?

Are any sectors intended for the disk still in system memory, waiting for output?

Note that these questions are relevant even to a single-user J500, because PRINT, Type-Rite and other programs may have several tasks which access the specified disk.

If the system concludes that removal is safe, this message appears -- often instantaneously:

```
REMOVE DISK NOW
```

On the other hand, the following message indicates that the disk can not be removed safely, and that REMOVE must be executed again after an appropriate delay:

```
REMOVE FAILED
```

Similarly, if nothing happens for 60 seconds, because certain system processes are causing a delay, this appears:

DISK ACTIVE

Again, the operator must try another REMOVE. Meanwhile, an execution of the FILES utility may be helpful in determining what job is still using the disk.

If REMOVE displays a disk I/O error message (code 8), then a sector in memory could not be written out correctly.

If System II is a configuration with overlays, then a REMOVE for the primary disk may appear to work, with a "REMOVE DISK NOW" message. However, the next keyboard command will cause the blinking message SOVL. The entire system will continue to malfunction until one of these actions is taken:

The same system disk is replaced in the drive.

Another disk with an identical system is inserted.

The system is reloaded from some disk.

RENAME -- Rename a Disk File

RENAME changes the name of a disk file, but has no effect on the file's attributes or contents. A permanent file (with a "PF" attribute) cannot be renamed without a preceding CHATR.

Command Line Format

RENAME old new

old is the current name of the file to be renamed. As always, a device or a "$" prefix will be needed if the file is not on the current nominal disk.

new is the proposed new name for the file. A device prefix will be ignored here. If the name already exists on the disk containing "old," then RENAME will fail.

Examples

```
RENAME ALPHA BETA
RENAME $GAMMA DELTA
RENAME FP01:ABC.01 ABC.02
```

RLDR -- Relocating Loader

RLDR combines separate binary files (output from an assembler or compiler) into a single executable binary file, resolving external references among the input modules.

Command Line Format

```
RLDR[/A][/L][/D] B=out [L=list] [P=w] [G=glbl]
                       [ind/I]...  [fn]...
```

B=out is the name of the output file to be created; it must not already exist. The format of the file is described in the System II Assembler manual.

/A is an optional switch for generating absolute, rather than relocatable, output. The initial origin of the BSECT will be zero, and the initial origin of the TSECT will be X'100. ASECTs will be copied as supplied. RLDR makes no attempt to prevent conflict between ASECTs and relocatable code.

/A is appropriate for linking a newly generated version of System II itself. Because an absolute module has no start address, /A should not be used for programs to be loaded with LDABS.

L=list is a device or file for the load map, which is always displayed as well. The load map shows entry points for all files included in the link operation. It also includes diagnostic and error messages.

/L specifies a "long" version of the load map -- all global symbols, not just linked entry points.

P=w is a decimal number of words for RLDR's workspace and symbol table partition. The default is 1500 words.

/D forces the global symbols into the output file. The output file may then be included in a subsequent RLDR operation and combined with more binary modules. If /D is chosen, unresolved externals will be included without error in the output file. A subsequent RLDR operation may resolve them. For final modules, intended for use with System II, do not choose /D.

G=glbl is a filename used in conjunction with the /D switch. With /D in effect, G= restricts the global symbols in the output to the set named in the "glbl" file. That file consists only of symbol names, one per line.

ind/I is an "indirect" filename. The /I distinguishes it from ordinary "direct" filenames. Any number of indirect names may appear in the command. Each file contains a series of binary module filenames, one per line, to be included in RLDR processing. This allows the use of more binary module names than the command line can hold in its 80 characters.

An indirect file may also contain comments, which will be copied to the load map. A comment is indicated with ";" or "*" in column 1. Names with /I switches are not allowed within an indirect file.

fn is any field on the command line with no "=" in it. Such fields name individual binary modules to be included in RLDR processing; any number of files may be specified in this way.

Modules are processed in the order specified; "indirect" and "direct" filenames may appear in any sequence needed.

RLDR uses the last non-zero end statement (.END xx) as the start address for the linked module. Only a TSECT address may be used.

The symbols XXBEND and XXTEND are automatically stored in RLDR's symbol table. These symbols give, respectively, the end address of the B sector and the T sector. If these symbols also appear in a binary input module, a "DUPLICATE SYMBOL" message will appear in the list output of RLDR. This will always happen if the BASIC/Assembler interface module BASM is linked, but in this case, the validity of the result is not affected.

Non-Fatal Errors

nnnnnn vvvv r DUPLICATE SYMBOL

This message flags each definition of an already defined global symbol. That is, the first definition is not flagged, but every subsequent definition is flagged. In a listing, the message immediately precedes the usual line for a global symbol, and the first three fields are precisely the same:

nnnn	Symbol name
vvvv	Value in hex
r	Relocation mode --

A	Absolute
B	BSECT relocatable
T	TSECT relocatable

This error may be ignored only if the symbol is defined to have the same value each time.

DISPLACEMENT EXCEEDED

An instruction references a base page location larger than X'FF. Should not appear in normal operation.

dddd r X'7F EXCEEDED WARNING

An instruction not in the following list references a base page location larger than X'7F --

JMP, JSR, LD, ST
ADD, SUB, AND, OR
DSZ, ISZ, SKAZ, SKNE, SKG

The instruction's displacement within its module is hex "dddd." The relocation mode "r" applies to the displacement, not to the symbol in error.

dddd r CAN NOT RESOLVE DISPLACEMENT

An instruction attempts to reference an absolute symbol not between +127 and -128 decimal. Or, the displacement for a PC (or index register) relative address reference is not in this range. The "dddd" and "r" fields are as above.

nnnnnn SYMBOL DOESN'T EXIST

The global symbol "nnnnnn" was referenced but not defined. System II will not execute a module containing unresolved global references.

Fatal Errors

If one of these messages appears, RLDR deletes the "B=" file before terminating.

INPUT FORMAT ERROR

A module contains a "%OVL" line specifying an overlay level greater than 9. Only the 6th character position in the line defines the overlay number.

SYMBOL TABLE FULL

Part of RLDR's work partition is a table, used to match global symbol definitions with global symbol references. If this message appears, increase the "P=" field in the command line and execute RLDR again.

BINARY FILE FORMAT ERROR

Either a record could not be read, or a file expected to be a Relocatable Load Module had an unacceptable format.

CHECK SUM ERROR

Each RLM record contains its own checksum word.

NO OUTPUT FILE

The "B=" field, required in every RLDR command, was missing.

TOO MANY EXTERNALS

A maximum of 255 global symbols are allowed for RLDR output.

BSECTOR EXCEEDED

More than 255 words of BSECT were used. A TSECT directive was probably missing from an assembly language source.

Examples

```
RLDR B=LDMODL ABC XYZ
```

This command produces the output file LDMODL on the nominal disk. LDMODL will consist of ABC and XYZ, in that order.

```
RLDR/D B=BIN.RB $SYSSYM.RB $ERCODE.RB ABC DP01:XYZ L=LPT1
```

This command produces BIN.RB on the nominal disk; global symbols will be included. The short form of the load map is printed. BIN.RB will consist of the standard system symbol and error code modules (from the primary disk), ABC from the nominal disk, and XYZ from DP01, in that order. It may also contain unresolved externals.

```
RLDR B=PROG BIN.RB ALPHA/I BETA/I L=MAP P=5000
```

This command produces PROG on the nominal disk. PROG consists of BIN.RB (perhaps from the previous example) and whatever binary modules are listed in the files ALPHA and BETA. A short-form load map is stored in file MAP. A partition of 5000 words is reserved for the symbol table during RLDR execution. If unresolved externals were present in BIN.RB, they will be resolved in this RLDR operation, or an error message will appear in MAP.

```
RLDR INDA/I MOD1.RB INDB/I INDC/I B=DD00:BIGBIN
```

This command BIGBIN on BD00 -- all the modules listed in the indirect file INDA, followed by MOD1.RB, followed by all the modules listed in INDB, followed by all the binary modules listed in INDC. The load map is not preserved -- possibly a mistake.

```
RLDR/D/L G=GAMMA RBLIST/I B=PRGRM.RB L=LODMAP
```

This command PRGRM.RB, which contains global symbols, but only those listed in the file GAMMA. PRGRM.RB includes all the binary modules listed in the file RBLIST. The long version of the load map is stored in LODMAP.

RTDUMP -- Real-Time Memory Dump/Edit

RTDUMP generates a hex and ASCII dump of specified memory ranges. It can also alter the contents of memory.

Command Line Format

```
RTDUMP[/R] [lo:hi]... [L=fn[/A]]
```

/R repeats the RTDUMP display, with continuous updating, until <CANCEL> is pressed.

lo:hi specify low and high address limits for the dump. Up to 10 sets of limits may appear.

L=fn names a device or file for the dump. If this field appears, dump lines roll up the screen as they are recorded, and the /R switch is ignored.

/A appends output to "fn" if it already exists. If omitted, each dump deletes and replaces the file, preserving only the one most recent dump.

RTDUMP can be executed either interactively or noninteractively.

In the noninteractive mode, you key in a command line, as shown above, followed by a <RETURN>. If your command line did not include a /R switch, one dump is performed, and the message RTDUMP appears on the bottom line, followed by the cursor. Control does not return to System II; RTDUMP is still running, in the interactive mode.

If your command line included a /R switch, the display will be updated until you press <CANCEL>. Then RTDUMP will enter the the interactive mode.

When the message RTDUMP is on display and the cursor is on the bottom line, the <CANCEL> key will return control to System II.

Interactive Mode

In RTDUMP's interactive mode, you enter command line fields, as described above, on Line 24 of the screen. Then one of the following function keys must be pressed. Commands with syntax errors are simply ignored. <CANCEL> terminates a dump in progress and restores the cursor.

<F1> Adds all ranges ("lo:hi" pairs) on Line 24 to the range list. The previous entries in the range list are pushed down, so new entries will be displayed first.

<F2> Sets the range list to just the pairs on Line 24.

<F6> Sets up a memory alteration for <F7>. First, pairs in this format --

```
address:content
```

-- are entered on Line 24; then <F6> is pressed. Both elements are hex numbers; "content" is the proposed alteration. RTDUMP responds by displaying the address, the current content, and the proposed alteration, so they may be checked before using <F7>.

<F7> Applies the alteration displayed by <F6>. Anything on Line 24 is ignored. If any other function key is used after <F6>, the next <F7> is ignored; memory is not altered. If the <F7> is accepted, the last <F6> display is updated.

Examples

RTDUMP/R 100:11F 5200:523F <RETURN>

This command invokes RTDUMP noninteractively. It displays the memory ranges specified until you press the <CANCEL> key. The righthand range appears above the lefthand range.

RTDUMP <RETURN>

This command invokes RTDUMP interactively. Commands like this one can be entered:

```
C500:C5FF A500:A5FF <F2>
```

This command sets the list of range pairs to the values shown, and displays a few ranges beginning at A500. Because the range is too large to fit on the screen, you may then enter --

```
L=LF <F1>
```

-- which will dump on a file called LF all the addresses in the ranges specified in the previous command -- the "current list" until added to with an <F1> reset with an <F2>.

```
A500:2020 A501:2020 <F6>
```

This command, used when RTDUMP is in the interactive mode, produces the following display:

```
ADDR   A500   A501
OLD    xxxx   yyyy
NEW    2020   2020
```

If your next key is <F7>, memory will be altered and the display will be change to:

```
ADDR   A500   A501
OLD    2020   2020
NEW    2020   2020
```

SDUMP -- Stand-Alone Memory Dump

SDUMP is a debugging tool which loads under System II, but executes without further System II interaction. This utility prints the hex and ASCII representations of the contents of memory in specified address ranges.

SDUMP requires that the hardware system include a line printer -- which must be named LPT1 -- and a console. It handles the CRT and the line printer with self-contained code.

Command Line Format

```
SDUMP    <RETURN>
```

This command loads SDUMP and displays the message "DUMP START ADDRESS hhhh." Record this address on paper and proceed.

You may now enter pairs of memory ranges -- hex numbers in this form:

```
xxxx:yyyy
```

Each <RETURN> sends the contents of the specified range to LPT1. <CANCEL> halts any printing in progress, and returns the cursor for further input.

To resume dumping after a system failure, press the RESET switch on the console, set the console register switches to SDUMP's start address, and press the START switch.

To perform any other task, you must reload System II from disk.

Dump Format

First line: contents of the registers.

Second line: contents of the 16 stack locations.

Subsequent lines: memory contents, 8 words per line.

SI5DEF -- J500 System Definition Inquiry

For details on this program, refer to the SIIDEF write-up.

SIIDEF -- J100 System Definition Inquiry

SIIDEF is a J100 program; SI5DEF is a J500 program. Either utility displays or lists the characteristics of the System II configuration currently running, including all the questions and answers involved when the system was generated.

SYSGEN accepts as input a disk file which was created by SIIDEF. Similarly, J500GN accepts a file from SI5DEF. See the relevant system generation manual for details.

Command Line Formats

```
SIIDEF [L=fn]
SI5DEF [L=fn]
```

L=fn is an optional field, specifying a disk file or printer for the output of the utility. If this field does not appear, the output is displayed one full screen at a time; <RETURN> will then bring up another page, and <CANCEL> will terminate the display.

The first portion of a typical SIIDEF listing is reproduced on the next page.

The nominal disk and the primary disk shown are the current settings, which may be different from those specified at system generation time.

For each J100 universal CRT controller, system initialization overrides the SYSGEN refresh address with the actual hardware setting, as shown in SIIDEF output.

Any device characteristics altered in memory by ACUP, BCUP, or similar programs will also affect SIIDEF/SI5DEF output.

When the system is loaded, every disk's total retry count and error count are zeroed. When an operation is retried, the disk's retry count is increased by 1. If the maximum number of retries does not overcome the failure, the disk's total error count is increased by 1. An information line for each disk device, showing the current number of retries and errors, appears in SIIDEF/SI5DEF output if the values are not zeros.

```
  ;SIIDEF
  ;OPERATING SYSTEM DESCRIPTION
  ;CONFIGURATION  6
  ;END OF SYSTEM  3280

SYSTEM DATE ?
79 4 4

BUFFER POOL SIZE (HEX) ?
1400
  ;BUFFERS   AC00:BFFF   BUFFER CONTROL TABLE  AAC0:ABFF

MAX BUFFER SIZE (HEX) ?
0200

DISK SECTOR MANAGEMENT TABLE SIZE ?
3

MAX NUMBER OF PROGRAM PARTITIONS ?
28
  ;PARTITION SPACE  3280:AAAF
  ;END OF PARTITION SPACE   AAC0

MAX NUMBER OF JOBS ?
14

FLOPPY DISKS?
FP00 3B 00
FP01 3B 01

PERTEC D3341 DISKS?(3+3 MEGABYTES)?
DP00 1A 0 0
DP01 1A 0 1

PERTEC D3441 DOUBLE-DENSITY DISKS (6+6MEGABYTES)?

PERTEC D3441 DBL-DENSITY (512 WORD SECTOR) DISKS?

CDC 9762 DISKS?

PRIMARY DISK ?
FP00

DEFAULT NOMINAL DISK ?

J100/J105/J50 CRT TERMINALS ?
CRT1 28 DC00 FP00
CRT2 21 D800 FP00
CRT3 22 D400 FP00
CRT4 23 D000 FP00

CHARACTER OUTPUT DEVICES/CENTRONICS/PRINTRONIX PRINTERS?
LPT1 0F 0080 132 66  FF TS LC LU FB

DIABLO/QUME PRINTERS ?
DPR1 0E 0080 132 66  8 12  FF FS TS LC FB
```

SORT -- File Sort

SORT arranges records in ascending or descending order, based on the content of sort fields that you choose. SORT handles both sequential and hash files; records may be of either fixed or variable length.

You may define the sort parameters either interactively or in a single System II comamnd line. The parameters supply SORT with information about the type of records in the file ("alpha" or "binary"), the maximum record size, the fields to be sorted, and their order of processing.

The special binary file SORTM.SB must be on the current primary disk when SORT is invoked. If an input/output code translation table is to be used (SORT.CT, as described in the System II Reference manual), it must also be on the primary disk.

Interactive Use of SORT

Key in this System II command:

```
SORT   <RETURN>
```

The following display appears. The cursor is positioned for the MODE parameter. After you key in each parameter, move to the next one with the <TAB> key. The parameters are described on the next few pages.

```
MODE: _
RECORD SIZE:
INPUT FILE NAME:                    OUTPUT FILE NAME:
                                    APPEND OUTPUT TO EXISTING FILE? N

                        SORT KEY TABLE
      KEY #     SORT CODE     FIELD ORIGIN     FIELD SIZE
       1
       2
       3
       4
       5
       6
       7
       8
       9
       10

PARTITION SIZE:                     SCRATCH DEVICE:
```

MODE

Your choices are ALPHA or BINARY. If the file you want to sort was created by Type-Rite or EDIT, choose ALPHA. Users of BASIC and hash files generally know the characteristics of their files, and can apply the information in the following pages to their particular needs. Type-Rite and EDIT, however, let you make files without worrying about record format, etc. For that reason, specific guidance is offered about the parameter choices to make for Type-Rite files. Then general file information is given so that other users can figure out what they need to do.

In general, ALPHA mode defines the files to be line-oriented; that is, the end of each record is marked by a line terminator such as a carriage return. BINARY mode defines the records as of "transparent" type; that is, line terminators are ignored. The two modes have slightly different effects with respect to sequential and hash files.

For sequential files, ALPHA simply means that input reads are stopped when a line terminator is encountered and tabs are expanded. The input records thus may be lines of variable length, such as those in BASIC symbolic files or those created by Type-Rite or EDIT. Input and output records have the same format (except for the tab expansion). BINARY mode, when applied to sequential files, means that input records are of fixed length and line terminators are ignored. Sequential binary files output by BASIC may be this kind if strings are forced to fixed length.

For hash files, both ALPHA and BINARY use the same method of reading the files; each record is read without regard for terminators. Records may be of either variable or fixed length, since hash file records contain a length count. In BINARY mode, SORT will include the length count as the first word of the output record. In ALPHA mode, SORT will add two terminators (two X'00 characters) to the end of each record.

RECORD SIZE

Key in the maximum word count of a logical record in the input file. (For Type-Rite files, one-half the value of the largest righthand margin setting in the file, plus one.)

In ALPHA mode for sequential file input, record size is one-half the maximum number of characters allowed in a line, plus one. In BINARY mode for sequential file input, the record size is the number of words in a logical record. For hash file input in either mode, the record size is the maximum number of words in a logical record in the file (the key is considered part of the logical record).

INPUT FILE NAME and OUTPUT FILE NAME

Enter the name of the file containing the data to be sorted. Then enter the name of the file in which you want to store the sorted result. The output file will have the format you selected under MODE. If the input is a hash file and the mode is BINARY, SORT will add to each output logical record a word containing a count of the words in the record. The word count is the number of data words plus one (because the count is itself a word, and is included). Do not include this word added by SORT when calculating a sort-key field origin. If the input is a hash file and the mode is ALPHA, each output record will consist of the hash file record (whose size including the hash file key may not exceed RECORD SIZE) and a word of binary zeros.

APPEND OUTPUT TO EXISTING FILE?

The default response for this parameter is "no," indicated by the N already on the screen. If you key a Y over the N, you are telling SORT that your output file already exists, and that you want the result of the sort to be added to the end of it. If you accept the N and the output file already exists, SORT will display an error message.

SORT KEY TABLE

The sort key table defines the field (or fields) in each record which will determine the order of records in the output file. From 1 to 10 keys may be specified. Key 1 is the "major" key, "within" which key 2 determines further sequence, then key 3, and so on.

It takes three parameters to define a sort key: sort code, field origin, and field size. Field origin is the offset in words from the beginning of the record (which is word 0, so a field starting there has an offset of 0) ; field size is the size in words of the sort key field.

For Type-Rite or EDIT files, field origin can be determined directly from the tab settings in the file. If the tab setting is odd-numbered, subtract 1 and divide by 2 to get field origin. If the tab setting is even, subtract 2 and divide by 2. (The separate calculations are required because you have to set field origin at a computer-word boundary, and a column can begin in the middle of a computer word. The character position just before such a column must be blank.)

For Type-Rite or EDIT files, field size is also calculated separately for odd- and even-numbered character lengths. If odd, round to an even number and divide by 2. If even, just divide by 2. Be careful not to wind up with a field length that crosses over the boundary of the adjacent field origin. You may have to subtract 1 after you have divided.

The sort code is a two-letter combination, in which the second letter indicates that output is to be "ascending" or "descending" with regard to this field.

Code:	Comparison:
SA or SD	Signed binary, over the entire field. Useful for BASIC's 4-word representation of a decimal value in a "binary" file.
AA or AD	Signed binary, on each word in the field. Useful for BASIC's 1-word representation of an integer value in a "binary" output file.
LA or LD	Unsigned logical, on each word in the field. Useful for binary data sequences from assembly programs.
CA or CA	Character by character, over entire field. The Videocomputer's standard collating sequence for printable characters is shown in the ASCII chart at the end of this chapter. Useful for a file generated by Type-Rite, by Data-Rite, or by BASIC fixed-field "symbolic" output.

Automatic data code translation of SORT input and output -- for example, between ASCII and EBCDIC -- is possible, regardless of the comparison code specified. For details on this facility, refer to the write-up for the SORTM.SB program, and the SORTM.CT file, in the System II Reference Manual.

PARTITION SIZE

Key in the decimal number of words in memory that you want to allocate as scratch space during sorting. If you key in nothing, a default partition will be allocated according to the rules described in the System II Reference Manual. Type-Rite and EDIT files never require more than the default partition size.

SCRATCH DEVICE

Key in the name of a disk device, or key in nothing to accept your CRT's nominal device. You may also key in MEMORY; the sort will be performed within the memory assigned under PARTITION SIZE, with a default of 2048 words. The partition must be large enough to contain all of the records in the file simultaneously.

Interactive Functions

When SORT's full-screen template is on display and the cursor is in the upper part of the screen, <F1> moves the cursor to the bottom line, where you may key in commands from the set described below.

Do not press <CANCEL> instead of <F1> in this situation. SORT will terminate prematurely, close and delete any scratch files, display an error message, and return control to System II.

The commands listed below are followed not by <RETURN>, but by any Function Key other than <F1> -- shown here as <Fn>.

DELETE file <Fn>	Delete the file specified.
KB <Fn>	Position the cursor to the start of the sort key table and allow keyboard entry.
NEW <Fn>	Erase the display, redisplay the template, add the default values, position the cursor adjacent to MODE, and allow keyboard entry.
QUIT <Fn>	Return to the operating system.
SAVE file <Fn>	Save the mode, record size, and sort key descriptions from the screen on the disk file specified, which must not already exist.
RESAVE file <Fn>	Delete the specified file; then process as a SAVE command.
READ file <Fn>	Set up the screen according to a disk file created by a previous SAVE or RESAVE.
SORT <Fn>	Initiate the sort process. After you key in this command, there will be a pause while SORT does its work. If the sort completes successfully, some statistics will appear on line 22, screen, and the cursor will be ready to accept further sort commands. (The statistics are detailed in the System II Reference Manual.)

Sort Key Table File Structure

SORT's interactive mode can generate a special "sort key file" with the SAVE and RESAVE commands just described. Such a file may be used for another interactive SORT (READ command), or with a non-interactive SORT, as described later.

You may also use EDIT or Type-Rite to create a sort key file. The file must contain at least three records: one for mode, one for maximum record size, and at least one for sort field definition. It may contain as many as 31 sort field definition records. The records must have these formats:

Record:	Column Start:	Length:	Columns contain:
1: Mode	1	6	Either ALPHA or BINARY
2: Record Size	1	5	A decimal number of words
3: Field	1	2	A sort code (SA, SD, etc.)
	5	5	Field origin (offset in decimal words from beginning of record, which is word 0)
	13	5	Field length (in decimal words)

For example, suppose you want to sort a file of addresses alphabetically by state, and within each state by city. The maximum record size is 100 characters (50 computer words).

The city is at the start of a record (offset 0); it is limited to 30 characters (15 computer words). The state starts in column 31 of a record (offset 15 words), and is a 2-character (1-word) abbreviation. The sort key file would look like this:

```
(columns in        (0   0       1)
 each record)      (1...5.......3)

(record 1)          ALPHA
(record 2)          50
(record 3)          CA  15       1
(record 4)          CA  0        15
(end of file)
```

NON-INTERACTIVE SORT

All the necessary parameters for a SORT may be supplied in a single System II command line. A sort key file of the proper format must already exist (see below). The command format is:

```
SORT I=in O=out[/A] S=skf [P=w] [D=scr]
```

I=in is a required field naming the file to be sorted.

O=out is a required field specifying the output file.

/A is an optional switch causing SORT to append its output to an existing disk file.

S=skf is a required field naming a previously created sort key file.

P=w is an optional field to override the default work space partition size of 2048 decimal words.

D=scr is an optional field specifying a scratch device to be used for the sort. If this field is not included, scratch sorting will occur on the nominal device. If D=MEMORY is specified, scratch sorting will occur in the allocated partition space in memory.

ASCII CHART

When an ALPHA file is sorted with a code of CA or CD (character-by-character over an entire field), some questions may arise. What, exactly, do "ascending" and "descending" mean with regard to printable characters? Do numerals come before letters? What is the effect of capitalization and punctuation?

If no SORT I/O translation table (a SORTM.CT file, as described in the System II Reference Manual) is active, then the following chart is relevant. It lists the 8-bit data values for which the American Standard Code for Information Interchange (ASCII) defines a set of printable characters. The "Seq" column shows each character's position in the "collating sequence" or sorting order; the "Hex" column is the hexadecimal equivalent. The "ASCII" column corresponds to a CRT display, or to a character printer with an "ASCII-compatible" printwheel; other printwheels will produce different characters, or spaces, for some 8-bit values. 32 in sequence (hex 20) is the character always displayed and printed as a space.

Seq	Hex	ASCII	Seq	Hex	ASCII	Seq	Hex	ASCII
32	20		64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	;	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C	|
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	~
63	3F	?	95	5F	_			

Error Messages

A SORT error message has this format:

```
*******SYSTEM ERROR nn    ****SORT UTILITY ERROR xx
```

Here, "nn" is 0 (no system error), or one of the standard codes listed in the System II Error Messages manual.

The "xx" is one of these hex codes for an interactive SORT:

4A	Illegal sort key table; key field not all in one record, or no key specified.
76	Illegal hash file usage.
7A	Field containing cursor caused the system error.
7D	Extension was specified for the scratch device.
7E	Both a device and a file name were specified for the scratch device.
7F	Error in the field containing the cursor.

The "xx" is one of these hex codes for a non-interactive SORT, indicating a specific command line error:

01	Mode
02	Record size
03	Input file name
04	Output file name
05	Append
06	Partition size
07	Scratch device
08	Field not recognized
09	Key 1 sort code
0A	Key 1 field origin
0B	Key 1 field size
0C	Key 2 sort code
0D	Key 2 field origin
0E	Key 2 field size
	etc.

SPRCAT -- Disk Catalog Inquiry

SPRCAT performs three related tasks: (1) listing all the files on a disk in alphabetical or random order; (2) presenting the directory information for a given file on a disk; (3) presenting disk space usage information.

For any operation of SPRCAT which implies alphabetical ordering of file names, the current primary disk must include the SORTM.SB program, and must not include an active sort I/O translatiuon file (SORTM.CT).

Command Line Format

SPRCAT[switches] [disks] [files] [L=fn] [S=scr] [P=w]

The optional switches appended to the utility's name are:

/D Details. Each file generates one line of information, sorted alphabetically. This is the form of input required by the BLDCPY utility. From left to right, each line contains: (a) the name of the file; (b) a decimal number -- for a sequential file, its length in characters; for a hash file, its key length in words; (c) the file's attributes (described below); (d) a decimal number -- for a sequential file, the number of sectors; for a hash file, the number of index sectors; (e) the date the file was last opened for writing.

If no switches are used, file names are sorted and listed two per line, with only items (a) and (b).

/X Hex details. All of the features of /D, plus each file's complete catalog entry in hex.

/U Unsorted entries, two per line, items (a) and (b) only. This saves time; it also eliminates the requirements for $SORTM.SB and for sort workspace.

/S Space usage only. Statistics about the disk -- percentage of disk space remaining; number of disk sectors available; number used. This information is automatically included in listings generated by other switches. For DDnn devices, see the note later in this write-up.

The optional command fields are:

disks — Any number of disk device names; mixed types are allowed. The default is the nominal disk.

files — Any number of disk file names, with device prefixes as needed. If no "disks" parameters are used, this will restrict SPRCAT's output to information about just these specific files. Any mixture of "disks" and "files" parameters are acceptable.

L=fn — File or device for the output of SPRCAT. The default is the job's own screen. If "fn" is an existing disk file, the output is appended to it.

S=scr — Scratch device required for sorted output. The default is the nominal disk. "S=MEMORY" may be used; no scratch device will then be required.

P=w — Partition size for sort workspace, overriding SORTM.SB's computed default.

Error Messages

NOT AVAILABLE xxxxxx

The named device or disk file is currently in use by another program, and can't be accessed by SPRCAT.

FILE NOT FOUND xxxxxx

The name "xxxxxx" is neither a device nor a disk file.

ONLY /D /S /U /X ALLOWED

Your command included an undefined switch.

DISK ERROR - LOGICAL ADDRESS: xxxx

A disk error occurred. SPRCAT proceeds to the next primary catalog sector.

BAD ENTRY - LOGICAL ADDRESS: xxxx

An incorrect catalog entry was found. Bad entries are not included in the output. If 3 bad entries occur in the same disk sector, SPRCAT proceeds to the next primary catalog sector.

DDnn Space

On a Pertec double-density drive -- a device name in the form DDnn -- 6% of the disk space is not available for user files. As a consequence, when SPRCAT indicates that 94% of such a disk has been filled, further creation or expansion of user files is not possible.

Examples

```
SPRCAT
```

This command produces an alphabetized listing (two files per line) of all the files on the nominal disk. Output appears on the screen.

```
SPRCAT/X/U FP00 FP01 DP00 DP01 DP01:VEND L=CATF
```

This command produces unsorted, detailed lists, including hex catalog entries, for every file on two floppies and one cartridge. As a last item, it lists again one particular file -- VEND on DP01. It sends all this information to a file on the nominal disk called CATF, appending to that file if it already exists.

```
SPRCAT/D/S DD01:CUST L=DPR1
```

This command prints details about one file -- CUST on DD01. It also prints space usage statistics for that disk.

FILE ATTRIBUTES

AP Attribute protected -- Files with this attribute cannot have their attributes changed.

PF Permanent -- Files with this attribute cannot be deleted without a CHATR to nullify the attribute.

RO Read only -- Files with this attribute cannot be written to while the attribute is in effect.

WO Write only -- Files with this attribute cannot be read while the attribute is in effect.

ES Execution sharable -- Files with this attribute can be used by more than one job at a time. Applies only to executable files written in reentrant code.

RS Read sharable -- Files with this attribute can be read by more than one job at a time.

WS Write sharable -- Files with this attribute can be modified simultaneously by more than one user.

HF Hash file -- Declares that records within the file are located on the disk according to a hashing algorithm, rather than sequentially. (See the System II Reference manual.)

DK Duplicate keys allowed on a hash file.

BT Batch File structure on a hash file. (For Data-Rite.)

SF Secure file -- protected from unauthorized access.

SP Secure file with permits.

TL Top loading -- Files with this attribute are loaded for execution at the top of memory. Appropriate only for AM Jacquard software files -- ".SB" filename extension.

STAD -- Set Time and Date

STAD sets values for the system date and time. It may be executed from any terminal at any time. The related utility GTAD may be used to check the current settings.

When System II is loaded, the date is initialized to the value specifed by SYSGEN, and the time is set to midnight (00:00:00). The date will be advanced automatically at each subsequent midnight, but the system assumes that every month has 31 days, so STAD should be executed when this assumption is incorrect.

It is strongly recommended that you keep the time and date set correctly. SPRCAT and Type-Rite can supply you with useful information about the creation or modification dates of disk files, but only if the system time and date are properly maintained. Application programs may also be affected.

It is a good idea to include a STAD command in the SYSICF file (see Section 1).

Command Line Format

```
STAD
```

The following message appears:

```
ENTER DATE (YY/MM/DD)
```

Key in the date in the order shown -- first the year, then the month, then the day. When the date information is accepted, the following message appears:

```
ENTER TIME (HH/MM/SS)
```

Key in the hours and minutes. When the time information is accepted, control returns to System II.

STAD expects the time according to a 24-hour clock:

```
00:00          midnight
00:59          59 minutes after midnight
 1:00          1:00 AM
12:00          noon
13:00          1:00 PM
23:59          11:59 PM
```

Delimiters between date and time fields may be indicated in one of three ways -- slash, colon, or space -- and leading zeros may be omitted. That is, keying in 78/12/1, or 78:12:1, or 78 12 01 will all set the date to December 1, 1978.

TDSPLY -- Tape Display

TDSPLY displays or lists, in hex and ASCII, the data on a magnetic tape.

Command Line Format

```
TDSPLY tapev[/E] [n:m] [F=i:j] [B=k] [L=fn]
```

tape is a required parameter -- the device name of a magnetic tape drive, such as MT00.

/E is an optional switch, indicating that the tape was written in EBCDIC. With /E set, TDSPLY will display hex with EBCDIC equivalents, rather than ASCII.

n:m is an optional field designating a range of tape blocks, in decimal. If both "n:m" and "F=i:j" are omitted, the entire tape is displayed.

F=i:j is an optional field designating a range of tape files. (Each file on a tape ends with one filemark.) There is no default for this field, but F=i is equivalent to F=i:i -- that is, one file only.

B=k is an optional field specifying the maximum tape block size as a decimal number of characters. The default is 2000. If the block size displayed matches k, the block may have been truncated on input. To avoid truncation, set k larger than the maximum block size to be read.

L=fn is an name for a listing file or device.

Error Processing

COMMAND ERROR

The TDSPLY command is unacceptable as entered.

```
*** TAPE ERROR ***
STATUS: ssss
ACTUAL BYTE CNT: bbbb
# OF RETRIES: rrrr
```

Reading the tape caused an error. The associated software status was ssss (hex). The number of bytes (characters) actually read was bbbb (decimal). The number of retries is meaningless in this context, and should be ignored.

Examples

```
TDSPLY MT00
```

This command displays the contents of the tape mounted on unit MT00. If the tape blocks are longer than 2000 bytes, they are truncated in the display.

```
TDSPLY MT00 F=1 B=5000 L=LPT1
```

This command displays and also prints, on LPT1, the contents of the tape up to the first single filemark. Only if blocks are longer than 5000 bytes are they truncated.

TRANSL -- ASCII/EBCDIC Translation

TRANSL converts ASCII input into EBCDIC output. Optionally, it can do the reverse. Either mode of operation normally requires the presence on the primary disk of the standard code translation table, EBCDIC.CT. Conversion according to some other table is also possible.

TRANSL detaches itself from the terminal and runs in background mode, so that other jobs may be initiated while the code translation is in progress.

Command Line Format

```
TRANSL[/E] in out [T=ct]
```

/E is an optional switch that reverses the default direction of conversion. Default operation uses the first 128 words of a translate table; /E uses the second 128 words. If no T= field appears in the command, this means that the default is ASCII input to EBCDIC output, and that /E indicates EBCDIC input to ASCII output.

in names the file to be translated.

out names the new file to be created. If it already exists, an error message will appear.

T=ct is an optional field specifying your own custom code translation table. If this field is omitted, the disk file $EBCDIC.CT must be available. For the structure of a translation table, and for a listing of EBCDIC.CT, see the System II Reference Manual.

UPDATE -- Source File Update

UPDATE is used to to merge a file of insertion, deletion, and replacement commands with an existing source code file; to implicitly resequence the lines; to produce a revised source code file.

UPDATE processing is appropriate for source programs written in any Videocomputer language. The utility does not change explicit statement numbers in BASIC; for this purpose, see the description of the RENUM utility in the BASIC manual.

Command Line Format

```
UPDATE [U=filnam] [I=infil] [O=outfil] [C=filnam]
```

U=filnam — is an optional field, specifying a line-oriented symbolic file containing UPDATE commands and lines to be merged with an input file.

I=infil — is an optional field naming an existing line-oriented symbolic file. If I= appears, O= must appear too.

O=outfil — is an optional field naming an output file. If a file of this name exists, it will be deleted and recreated. If O= appears, I= must appear too.

C=filnam — has exactly the same meaning as the U= field; it is provided for compatibility with earlier releases. The command should not have both U= and C= fields.

! If your command line includes a U= (or C=) field, an I= field, and an O= field, the input file will be updated according to the instructions given in the command file, and an output file
! will be created.

! If your command line includes I= and O=, but no U= (or C=), the
! input file will be copied to the output file with some changes.
! Sequences of blanks are converted to Horizontal Tab characters;
! trailing blanks are deleted; each line is forced to contain an even number of characters by appending a blank where necessary.

! Input (and command) lines longer than 82 characters are truncated
! before processing them.

Rules for UPDATE Command Files

References to implicit line numbers in the input file must appear in ascending order. The lines of the command file are read and processed one at a time, according to the following rules.

1. File Positioning. Any command file line in which the first nonblank character is a positive decimal number causes lines of the input file to be appended to the output file up to and including the specified line number. Lines of this type may contain comments following the specified line number.

2. Line Insertion. Any command file line whose first nonblank character is nonnumeric and not the minus sign will be appended to the output file at its current position.

3. Single Line Deletion. Any command file line whose first nonblank characters are a minus sign followed by a line number will cause lines of the input file to be appended to the output file up to but not including the specified line. The specified line will not be appended to the output file. Lines of this type may contain comments following the specified line number.

4. Multiple Line Deletion. Any command line in one of the following forms --

```
 m n
 m,n
-m n
-m,n
```

-- in which m and n are line numbers, with m less than or equal to n -- will cause lines of the input to be appended to the output file up to, but not including, the first specified line number (m). Lines m through n will not be appended to the output file.

5. If the input file is written in BASIC, the first nonblank character in command file lines of types 1, 3, and 4 must be a B.

Rule 5 is imposed by the fact that BASIC source files contain lines beginning with statement numbers. In a command file applicable to BASIC source, UPDATE processes lines beginning with Bs just as in a regular command file. All other lines it appends to the output file at its current position, regardless of their first characters.

Implicit line numbers are generated by the assembler and appear on listings output by ASM. Implicit line numbers are not generated at all by SBASIC, but appear on listings output from SGBASC (preceding the statement number).

Error and Status Messages

END OF INPUT

A normal status message indicating that the input file has been read completely.

END OF UPDATE

A normal status message indicating that the command file has been read completely.

REQUIRED: U= OR C=/ I= & O=/ U= OR C= & I= & O=

You started UPDATE with no parameters, which is not a legal command. UPDATE displays the message above for your information and returns control to System II.

ERROR: U= & C= BOTH USED
 or O= BUT NO I=
 or I= BUT NO O=

You entered a command line with illegal parameters. UPDATE displays one of the messages above for your information and returns control to System II.

PARTIAL LINE

This error message is immediately followed by END OF INPUT or END OF UPDATE. It says that either the I= file or the C= file did not end with a line terminator.

SYNTAX ERROR or SEQUENCE ERROR

The last line displayed from the UPDATE command file contains either a syntactical or sequence error. UPDATE returns control to System II.

Examples

```
UPDATE  C=SOURCE.UP  I=SOURCE.SR  O=FP00:SOURCE.SR
```

This command applies a set of update commands in the file SOURCE.UP to a source file SOURCE.SR, all on the nominal disk. The output file has the same name and appears on the floppy in FP00. This method preserves revision levels of source code.

```
UPDATE  C=SOURCE.UP  I=SOURCE.SR  O=SOURCE.NU
```

This command applies a set of update commands in the file SOURCE.UP to a source file SOURCE.SR, and creates as output the file SOURCE.NU, all on the nominal disk. This method preserves revision levels of source code.

Example of Update Command Set

Shown below are: (1) a BASIC program with line numbers output from SGBASC; (2) an UPDATE command file which could be applied to it; (3) the result, also with line numbers output from SGBASC.

```
1     20        INTEGER I
2     40        FOR I = 1 TO 10
3     60        PRINT I
4     80        NEXT I
5    100        PRINT &
6                'PROGRAM COMPLETE'
7    120        STOP
```

```
B 0
 10 ! INTEGER PRINTING PROGRAM
B 3
 65       PRINT I*2
B -6
          'THE END'
```

```
1     10 ! INTEGER PRINTING PROGRAM
2     20        INTEGER I
3     40        FOR I = 1 TO 10
4     60        PRINT I
5     65        PRINT I*2
6     80        NEXT I
7    100        PRINT &
8                'THE END'
9    120        STOP
```

COMMENT SHEET

V2-005-4

Jacquard Systems

FROM

Name _____

Business Address _____

_____

_____

Does this publication meet your requirements? yes ☐ no ☐

If not, please explain. _____

_____

_____

_____

Do you wish a reply? yes ☐ no ☐

COMMENTS

Describe any errors and/or suggested changes. Please include page number.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

(FOLD)

FIRST
CLASS
PERMIT NO.
2609
SANTA MONICA
CA.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN UNITED STATES

POSTAGE WILL BE PAID BY:

Jacquard Systems

1639 ELEVENTH STREET
SANTA MONICA, CALIFORNIA 90404

Customer Service Department

(FOLD)

CUT ALONG DOTTED LINE

1639 Eleventh Street
Santa Monica, California 90404

Jacquard Systems

3340 Ocean Park Boulevard
Santa Monica, California 90405

Jacquard Systems

3340 Ocean Park Boulevard
Santa Monica, California 90405

Jacquard Systems