Data General

Customer Documentation

RDOS/DOS User's Handbook

RDOS/DOS
User's Handbook

093-000105-04

For the latest enhancements, cautions, documentation changes, and other information on this product, please see the Release Notice (085-series) supplied with the software.

Ordering No. 093-000105

Printed in the United States of America
Revision 04, July 1987

NOTICE

DGC reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult DGC to determine whether any such changes have been made.

THE TERMS AND CONDITIONS GOVERNING THE SALE OF DGC HARDWARE PRODUCTS AND THE LICENSING OF DGC SOFTWARE CONSIST SOLELY OF THOSE SET FORTH IN THE WRITTEN CONTRACTS BETWEEN DGC AND ITS CUSTOMERS. NO REPRESENTATION OR OTHER AFFIRMATION OF FACT CONTAINED IN THIS DOCUMENT INCLUDING BUT NOT LIMITED TO STATEMENTS REGARDING CAPACITY, RESPONSE-TIME PERFORMANCE, SUITABILITY FOR USE OR PERFORMANCE OF PRODUCTS DESCRIBED HEREIN SHALL BE DEEMED TO BE A WARRANTY BY DGC FOR ANY PURPOSE, OR GIVE RISE TO ANY LIABILITY OF DGC WHATSOEVER.

RDOS/DOS
User's Handbook
093-000105

Revision History:

Original Release — March 1974 (Internal)	Second Revision — April 1975
	Third Revision — May 1978
First Revision — January 1975	Fourth Revision — July 1987

CONTENT UNCHANGED

The content and change indicators in this revision are unchanged from 093-000105-03. This revision changes only printing and binding details.

Contents

Introduction

This RDOS/DOS Handbook is a console reference to the features of RDOS and DOS and their utility programs. It describes CLI commands, many utility program commands, and the meanings of error messages returned while operating under RDOS or DOS. Unless a command is noted RDOS or DOS, it works in both systems.

This handbook does not provide detailed reference information nor does it substitute for manuals describing RDOS, DOS, the CLI, or the compilers, utilities and subsystems operating within the system framework. For that information see the *CLI User's Manual,* 093-000109, and the manuals listed in the preface to that manual.

RDOS and DOS require a Data General computer with at least 16K words of memory. These systems are available in a large number of configurations. Information in this handbook applies in general to all configurations but the restrictions of any specific version are not necessarily noted in the text.

Documentation Conventions

In the command formats, the following symbols have special meanings.

)	Carriage RETURN
↑)	Command line continuation: uparrow (↑)and RETURN
UPPERCASE	letters indicate entries which you must make literally.

lowercase letters indicate a name or specification which you must supply.

{ } Braces indicate that you must choose one of the arguments enclosed.

[ ] Italic brackets enclose an optional portion of a command line (except for RLDR). Do not enter the brackets.

... Ellipsis indicates that you can repeat the preceding argument.

dir Means any user directory.

pripart Means primary partition (RDOS).

secondpart Means secondary partition (RDOS).

subdir Means subdirectory (RDOS).

Where we must distinguish your entries from system responses, we will use

THIS TYPEFACE TO SHOW YOUR ENTRY)

THIS TYPEFACE FOR THE SYSTEM RESPONSE

Numbers

All numbers are decimal, unless specified otherwise; e.g., 11_8.

Program Controls and Interrupts

CTRL-A Terminate the current CLI command or program (except a text editor) and return to a higher-level - usually the CLI.

CTRL-C Same as CTRL-A, but wait for task I/O to finish, and create core image breakfile, BREAK.SVor FBREAK.SV.

CTRL-F From the background console (RDOS), terminate the foreground program.

CTRL-Q Restore console output suspended by CTRL-S.

CTRL-S Suspend console output; resume on CTRL-Q. Useful on CRT display terminals.

CLI Commands

Conventions

Templates (BUILD, DELETE, DUMP, LIST, LOAD, MOVE, and UNLINK Commands Only)

* Matches any single character in a filename except a period (e.g., LIST *) lists all single-character filenames without extensions).

Matches any character string of 0 characters or more, not including a period (e.g., LIST -) lists all filenames without extensions).

Other Features

%variable%

CLI replaces DATE, GCIN, GCOUT, GDIR, LDIR, MDIR, and TIME with the current value of these variables (e.g., RELEASE %MDIR%) releases the master directory).

@indirectcommand@)

Executes the contents of an indirect command string, which was built by transferring console input (XFER/A TTI filename, CLI commands, CTRL-Z) or by a text editor.

.MC

This extension defines an indirectcommand file as a macro file, which is built as an indirect file, but with .MC appended to the filename. filename) then executes the indirect command(s).

( )

Parentheses execute the commands they enclose on all filename arguments in sequence. For example, (PRINT, DELETE) LOG.CM prints and deletes the log file; (CRAND, PRINT) (MYFILE, YOURFILE) creates MYFILE and prints YOURFILE.

< >

Angle brackets expand filenames, and can be used with parentheses. For example, CRAND TEST.<1 2 3> creates TEST.1, TEST.2, TEST.3; (PRINT, DELETE) A.<1 2 3> prints and deletes A.1, A.2, A.3.

ALGOL (RDOS)

Format

```
ALGOL filename1 ...
```

Compile ALGOL source file filename.

Global Switches

/A	Suppress assembly.
/B	Brief listing; compiler source program only.
/E	Suppress compiler error messages at console.
/L	Produce listing to filename.LS.
/N	Do not produce binary file.
/S	Save temporary assembler source file in filename.SR.
/U	Append user symbols to binary output.

Local Switches

/B	Binary output file (overrides global /N).
/E	Error message file.
/L	Listing file (overrides global /L).
/S	Intermediate source output file.

Examples

```
ALGOL MAIN $LPT/L)
```

Produce MAIN.RB and a line printer listing.

```
ALGOL/A/S SUBR SOURCE)
```

Suppress assembly and output temporary assembler source file to SUBR.SR.

APPEND

Format

APPEND newfilename filename$_1$ *[...filename$_n$]*

Create newfilename and copy contents of one or more existing filenames, in the order given. The existing files are unchanged.

Example

```
APPEND DP1:BASE.SR A.SR B.SR DP0:C.SR)
```

Produce BASE.SR on DP1; BASE.SR will contain the contents of files A.SR and B.SR from the current directory and C.SR from DP0.

ASM

Format

ASM filename$_1$ *[...filename$_n$]*

Assemble one or more source files, using the extended assembler, into relocatable binary files (.RBs). (Extended assembler errors are described later in this book.)

Global Switches

/E	Suppress error messages.
/L	Produce listing to filename.LS (filename is first filename in the command).
/N	Do not produce relocatable binary file.
/S	Skip pass two and save symbol table in (F)BREAK.SV.
/T	Do not list symbols in listing.
/U	Include user symbols in relocatable binary.
/X	Produce cross reference of symbols.

Local Switches

/B	Relocatable binary name (overrides global /N).
/E	Error message filename.
/L	Listing filename (overrides global /L).
/N	Do not list this file.
/S	Skip this file on pass two.

Examples

```
ASM Z $LPT/L)
```

Produce Z.RB and a listing on the line printer.

```
ASM (A,B,DP0:E) BETA(A,B,E).RB/B)
```

Produce separate assemblies for source files A, B, and file E in the primary partition of fixed-head disk unit 0. Output relocatable binaries to disk files BETAA.RB, BETAB.RB, and BETAE.RB.

BASIC

Format

```
BASIC
```

Invoke the BASIC interpreter, which allows you to work in the BASIC language. BASIC.SV, configured via BSG, must exist for the command to be executed.

BASIC errors are described later in this book.

Example

```
BASIC)
```

When you invoke the BASIC interpreter it will initiate a short log-on sequence before starting up the BASIC system. You can leave BASIC and return to the CLI via the BYE command.

BATCH (RDOS)

Format

BATCH *[jobfile*$_1$ *...jobfile*$_n$*] [outputfile/O] [logfile/G]*

Create a job stream consisting of input job files for execution under the BATCH processor. If you omit *jobfiles,* the default input file is the card reader. The default output file is the line printer, and the default log file is the console.

BATCH commands are described later in this book.

Local Switches

/G This filename is the log file.
/O This filename is the output file.

Examples

```
BATCH)
```

Jobs input from card reader are executed and output is printed on the line printer. The log file is the console.

```
BATCH A.JB B.JB MT0:1 BATCHLOG/G)
```

Execute *jobfiles* A.JB and B.JB and the job on file 1 of MT0; print output on the line printer. The log file is a disk file named BATCHLOG.

BOOT

Format

BOOT *[specifier:] [newsystem]*

Release the current system and perform a disk bootstrap. *newsystem* may be the name of an operating system save file or the name of an executable program, or stand-alone program. *specifier* may be a partition or disk name. Consult the CLI manual for detailed rules on BOOT.

Examples

```
BOOT DP1 )
```

The disk bootstrap program, BOOT.SV resides on DP1 ; when invoked, it queries FILENAME? and you respond with the name of a system or an executable system program (e.g., DKINIT or DOSINIT). If the operating system has the default name (SYS.SV/SYS.OL), you need enter only a carriage return. The system you want to invoke must be on DP1 .

```
BOOT DP0:2NDPART:MYSYS)
```

DP0 is a primary partition, 2NDPART is a secondary partition, and MYSYS is the name of the RDOS system (RDOS example).

14

BPUNCH

Format

BPUNCH filename_1 ... *[filename_n]*

Punch filename(s) in binary on the high speed punch.

Example

```
BPUNCH A B DP1:MYFILE)
```

Punch files A and B from the current directory, and MYFILE from DP1, on the $PTP.

BUILD

Format

BUILD outputfilename *[filename$_1$ ... filename$_n$]*

Build file outputfilename, containing names of all current directory files specified. You may use template characters (*, -).

Global Switches

/A Include all permanent as well as nonpermanent *filenames.*

/K Do not include links.

/N Do not include extensions to *filenames* in outputfilename.

Local Switches

/A Include only files created this date or after, where the argument has the form: mm-dd-yy. mm and dd can be one or two digits.

/B Include only files created before this date. The argument has the same form as local switch /A.

/N Do not include files matching this name.

BUILD (continued)

Examples

```
BUILD OUT -.-)
```

Produce file OUT containing the names of all nonpermanent files in the current directory.

```
BUILD MYFILE -.RB ABC.RB/N)
```

Produce MYFILE, which consists of all nonpermanent .RBs in the current directory except ABC.RB.

```
BUILD OUTPUT -TEXT)
R
PRINT @OUTPUT@)
.
.
.
```

This sequence produces OUTPUT, which contains all nonpermanent filenames whose last 4 characters are TEXT; it then prints those files.

CCONT

Format

CCONT filename$_1$ blkct$_1$ *[...filename$_n$ blkct$_n$]*

Create one or more contiguously organized files. Each file has only the C characteristic, no attributes, and the length in blocks you specify. Each disk block is 256 words.

Global Switches

/N Do not zero each block in the new files (by default the system zeros each).

Examples

```
CCONT Z 25)
```

Create a contiguous file Z in the current directory with a length of 25 blocks.

```
CCONT TEXT 28 DP0:KG 57)
```

Create contiguous file TEXT in the current directory with a length of 28 blocks and file KG in directory DP0 with a length of 57 blocks.

CDIR

Format

```
CDIR directoryname
```

Create an RDOS subdirectory or DOS directory. The CLI assigns the .DR extension to directoryname.

Examples

```
CDIR C)
```

Create RDOS subdirectory or DOS directory C.DR in the current directory.

```
CDIR DP0:B)
```

Create directory B.DR within directory DP0.

CHAIN

Format

```
CHAIN savefilename
```

CHAIN savefilename into memory. This command overwrites the CLI, and should be used with caution. To use the CLI after a CHAIN command, savefilename must chain back to the CLI via .EXEC.

Example

```
CHAIN OSIP)
```

Overwrite the CLI with OSIP.SV.

CHATR

Format

CHATR filename$_1$ attribs$_1$ *[...filename$_n$ attribs$_n$]*

Change, add, or remove file access attributes of a file. To remove attributes, precede them with a minus sign (-); to add attributes, precede them with a plus sign (+). To retain current attributes, use an asterisk (*). You can enter one or more of the following attribs:

N	No link user can access this file.
P	Permanent file; cannot be deleted or renamed.
R	Read-protected file.
S	Save file.
W	Write-protected file.
0	Remove all attributes.
?	User-defined attribute (bit 9).
&	User-defined attribute (bit 10).

Enter two or more attributes for a file as a single argument.

Example

```
CHATR ALPHA +RW BETA 0)
```

Add read- and write-protected attributes to ALPHA's existing attributes; strip BETA of all attributes.

CHLAT

Format

CHLAT filename$_1$ attribs$_1$ *[...filename$_n$ attribs$_n$]*

Change, add, or remove the link access attributes of a file or files. The attributes control the operations which link users can perform on the file. To remove attributes, precede them with a minus sign (-); to add attributes, precede them with a plus sign (+); to retain existing attributes, use an asterisk (*). You can enter one or more of the following attribs:

N	No link use is permitted to this file.
P	Permanent file; cannot be deleted or renamed.
R	Read-protected file.
S	Save file.
W	Write-protected file.
0	Remove all link attributes.
?	User-defined attribute (bit 9).
&	User-defined attribute (bit 10).

Example

```
CHLAT A +W-P)
```

Remove the permanent attribute and add write-protection to A'scurrent attributes.

CLEAR

Format

CLEAR *[filename*$_1$ *...filename*$_n$ *]*

Clear the file use count to zero in one or more *filenames.* The command can be issued only from the background CLI when no foreground program is running.

Global Switches

/A Clear use count in all files in current directory except CLI.OL, CLI.ER, sysname.OL, sysname.TU and LOG.CM.

/D Set device use counts to zero (RDOS).

/V Verify files cleared on the console. (Used with /A .)

Examples

```
CLEAR/A)
```

Clear all files in the current directory (except those noted under /A). When use counts are zeroed, you can delete or rename the files.

```
CLEAR DP1:A:B)
```

Zero use count of file B in directory A in DP1

CLG

Format

CLG filename$_1$ *[...filename$_n$]*

Compile, assemble, load, and then execute one or more FORTRAN IV files (i.e., compile, load, and go). You can include source files in FORTRAN IV, files in assembly language (.SR extension), and binary files in the argument list of *filenames.*

FORT.LB (and, for multitasking, FMT.LB) must be available on disk. You can create overlays under CLG as in the RLDR command line by enclosing them in square brackets.

Global Switches

/B List only source program.

/E Suppress compiler error messages.

/M Suppress load map.

/T Multitasking mode (FMT.LB must be available on disk).

Local Switches

/A Assemble and load this file; do not compile.

/L Listing file.

/O Load this file; do not compile or assemble. (Other local switches of ASM and RLDR may be used as appropriate.)

CLG (continued)

Example

```
CLG/T MAIN AB/O AC/A $LPT/L)
```

Program MAIN is a file in FORTRAN IV language, AB is in binary and AC is in assembly language. The listing goes to $LPT. When compilation and assembly are completed, CLG tells RLDR to load MAIN.RB with the multitasking library FMT.LB. When loading is completed, MAIN.SV is executed.

COPY (DOS)

Format

```
COPY sdiskette ddiskette
```

Copy the entire contents of source diskette (sdiskette) to destination diskette (ddiskette). The sdiskette must be initialized, but the ddiskette must *not* be initialized.

The contents of sdiskette entirely overwrite the ddiskette. (The root portion of the bootstrap program, if any, is not copied).

Example

```
DIR DP0)
R
COPY DP0 DP1 )
R
```

Copy the contents of DP0 to DP1 .

CPART (RDOS)

Format

```
CPART partitionname blocks
```

Create an RDOS secondary partition, with a .DR extension, as a contiguous file whose length in blocks is given by blocks; blocks must be more than 48. If blocks is not an integer multiple of 16, it is truncated to the next lower multiple.

Example

```
CPART DP0:APART 50)
```

Create a secondary partition, APART, in primary partition DP0. APART is 48 blocks long.

CRAND

Format

CRAND filename$_1$ *[...filename$_n$]*

Create one or more randomly organized files in any given directory. Each file will have the D characteristic, no attributes, and zero length; it will grow as required.

Example

```
CRAND RFILE SFILE DP4:CASE1 )
```

Create random files RFILE and SFILE in the current directory and random file CASE1 in DP4.

CREATE

Format

CREATE filename_1 *[...filename$_n$]*

Create one or more sequentially organized files (RDOS) or random files (DOS) in any directory. Each filename has a zero length and no attributes.

Example

```
CREATE TEST TEST1 DP1:TEST2)
```

Create files TEST and TEST1 in the current directory and file TEST2 in DP1.

DEB

Format

DEB savefilename

Debug a save file. A symbolic debugger must have been loaded as part of the program save file. (See RLDR command.) The debugger issues a carriage return and awaits debugging commands.

After debugging, you can save the current core image of the program by returning to the CLI via the debugger command $V and then issuing the SAVE command.

Debugger commands are described later in this book.

Example

```
R
DEB A)      Debug A.SV
.
.         } Debugger commands
.
$V          Return to the CLI.
BREAK
R
```

DELETE

Format

DELETE filename$_1$ *[... filename$_n$]*

Delete one or more files from any initialized directory. To delete a directory, release the directory, then type DELETE directoryname.DR. All file use counts in directory must be zero -- see CLEAR. (To delete a link entry, use UNLINK because deleting a link deletes the resolution file.) You may use template characters (* and -) with this command.

Global Switches

/C Repeat each filename and wait for the deletion to be confirmed. A carriage return deletes the file; any other key does not.

/L List the deleted files on $LPT.

/V Verify deleted files on the console.

Local Switches

/A Delete only files created this date or after, where the argument preceding has the form: mm-dd-yy. mm and dd can be one or two digits.

/B Delete only files created before this date, where the preceding argument has the same form as local switch /A.

/N Do not delete files that match this name.

Examples

```
DELETE DP1:FILEA)
```

Delete FILEA without changing the current directory.

```
DELETE/V LIMIT.- MAP)
```

DELETED LIMIT.SR
DELETED LIMIT.RB
DELETED LIMIT.SV
DELETED MAP.

Delete and verify.

DIR

Format

DIR *[directoryspecifier:]* directory

Change the current directory to directory, initializing the new directory if necessary. In RDOS the *directoryspecifier* may be:

dir *[:subdir]*

In DOS the *directoryspecifier* may be diskette.

You may omit *directoryspecifier* if directory is a direct subordinate to the current directory.

Examples

```
DIR DP1)
```

Make DP1 the current directory.

```
DIR DP2:2NDPART)
```

Make secondary partition 2NDPART of primary partition DP2, the current directory.

DISK

Format

```
DISK
```

Display the decimal count of the number of blocks used and the number still available in the current partition or diskette. If the current directory is an RDOS subdirectory or DOS directory, the size of the parent partition is indicated.

Example

```
DISK)
LEFT:520   USED:88
```

The message indicates that 520 of 608 blocks on the current partition or diskette are still available for use.

DUMP

Format

DUMP outputfilename *[filename*$_1$ *...]*

Dump a file or files, *filenames,* from the current directory to a given file or device, outputfilename. If you omit a *filename* argument, all nonpermanent files in the current directory will be dumped. This includes subordinate directories, if any. You can use the template characters (* and -) for *filenames*. The *filename* can also be a directory name, with the .DR extension, (but not a disk name).

Global Switches

/A Dump all files, permanent and nonpermanent. (If not given, only nonpermanent files will be dumped.)

/K Do not dump links.

/L List dumped files on $LPT (overrides /V).

/S Dump file as segments of paper tape of up to 20K bytes each. (Tape segments are suitably numbered for later reloading in proper sequence.)

/V Verify dumped files on the console.

Local Switches

/A Dump only files created or modified this date or later. (Preceding argument has the form: mm-dd-yy where mm and dd may be one or two digits.)

/B Dump only files created or modified before this date. (Preceding argument has the same form as local switch /A).

/N Do not dump files matching this name.

/S Retain original oldfilename in current directory but rename it in the dump file.

Examples

```
DIR DP0)
R
DUMP/A MT0:0
```

Dump all files and directories in DP0 to MT0:0.

```
DUMP/V DP4:DUMPFI  -.SV  -.OL)
```

```
EDIT.SV
RLDR.OL
RLDR.SV
A.SV      } (Verification)
B.SV
.
.
```

Dump all nonpermanent files in the current directory with extensions .SV and .OL to outputfile DUMPFI in DP4.

```
DUMP SOURCE -.SR 22-10-77/A)
```

Dump all nonpermanent files with the .SR extension created on or after October 22, 1977 to file SOURCE in the current directory.

EDIT

Format

EDIT *[filename]*

Invoke the text editor to create or edit a text file. The editor responds with an asterisk prompt (*); you can then enter EDIT commands (summarized later in this book).

You can terminate the editor and return to the CLI by issuing the editing command UEH ESC ESC.

Examples

```
EDIT)
*
```

The editor is invoked, and awaits commands.

```
EDIT MYTEXT)
*
```

If you enter a filename, the editor will open the file for input (the editing command for this is UY filename ESC ESC). The file must already exist.

ENDLOG

Format

ENDLOG *[password]*

Close the log file opened by a previous LOG command, and terminate recording of CLI dialogue. *password* must match the password given in the LOG command, if any.

To print or delete the log file, type its full name: LOG.CM (or FLOG.CM for a foreground CLI).

Example

```
ENDLOG JSMITH)
```

The command closes the current log file, which was opened with the password JSMITH. You can then TYPE, PRINT, RENAME, or DELETE the file.

ENPAT

Format

ENPAT filename

Insert patch data in a patch file. ENPAT creates patch filename if it doesn't exist. The patch can be for a save or overlay file; you install the patch(s) in filename with the PATCH command.

When invoked, ENPAT asks questions about contents and location, as shown in the example. See the CLI manual for more detail.

Example

```
ENPAT ERSPC)

CREATING NEW PATCHFILE
SAVE FILE (0) OR OVERLAY FILE (1) 1
PATCH LOCATION: 31)
CURRENT CONTENTS: 6004)
NEW CONTENTS: 15762)
CONDITIONAL )
EXIT (0=NO 1=YES) 1
```

See PATCH for installation of the patch in file ERSPC.

EQUIV (RDOS)

Format

EQUIV newname oldname

Assign a new name to a tape or disk device before the device is initialized. The old name is a global specifier such as MT3 or DP1; the new name can be any legal filename. You must EQUIV a device before it has been initialized; when it is released, it will revert to its original name. You cannot EQUIV the master device, a secondary partition, or a subdirectory.

Example

```
EQUIV TAPE MT1 )
R
INIT TAPE)
R
```

All references to TAPE will be resolved as references to magnetic tape unit 1.

EXFG (RDOS)

Format

EXFG savefilename

Execute program savefilename in the foreground. You must type EXFG from the background console.

In a mapped system, savefilename can be any executable program, including the CLI or a system utility command line, like MAC A B C D E $LPT/L. You must have established foreground memory with SMEM. In an unmapped system, savefilename must include boundary information.

In any system, savefile will be executed only if it has enough space.

See the CLI manual for more detail.

Global Switches

/D Pass control to the debugger, not to the save file.

/E Give foreground and background equal priority (by default, FG has priority).

Examples

```
EXFG DP1:MON)
```

Find MON.SV in DP1, and execute it in the foreground.

```
EXFG CLG MAIN)
```

Execute CLG.SV in the foreground; CLG compiles, assembles, loads and executes the FORTRAN IV program MAIN.

FDUMP (RDOS)

Format

FDUMP MTn:filenumber

Fast-dump all files in the current directory to filenumber of the magnetic tape on drive n. FDUMP is a faster version of DUMP; it does not allow you to specify filenames, and dumps only to magnetic tape. FDUMP and DUMP are not compatible; you must reload FDUMPed files with FLOAD. File access is easier if you FDUMP to file 0 only. To stack dump files on one tape, you must increment dump file numbers by three: the first dump file is MTn:0, the second is MTn:3, the third is MTn:6, and so on.

Global Switches

/L	List dumped filenames on $LPT.
/V	List dumped filenames on the console.

Local Switches

/A	Continue the dump on this file when tape on the original drive is exhausted.
/L	Send dumped filenames to this file.

FDUMP (continued)

Examples

```
DIR DP0)
R
FDUMP/V)
```

Fast-dump all (except read-locked) files to file 0 of the tape on MT0. If you omit an arugment, FDUMP assumes MT0:0.

```
DIR DZ0)
R
FDUMP/L MT0:0 MT1:0/A)
```

Fast dump all readable files in DZ0 to MT0:0; when file space on this tape runs out, continue the dump on tape MT1:0 List dumped files on $LPT.

FGND

Format

FGND

Describe whether or not a foreground program is running. FGND returns one of two messages depending on whether a foreground program is running.

Example

```
FGND
NO FOREGROUND PROGRAM RUNNING
R
EXFG EDIT)
R
FGND)
FOREGROUND PROGRAM RUNNING
R
```

FILCOM

Format

FILCOM filename$_1$ filename$_2$

Compare two files word-by-word and print any dissimilar word pairs. The displacement is printed, followed by a slash, followed by the dissimilar words; for example:

```
Location:          File 1:          File 2:

15145/             000000           020044
```

Local Switches

/L Listing file (default listing is to the console).

Example

```
FILCOM TEST1 TEST2 $LPT/L)
```

Compare files TEST1 and TEST2 word by word and print dissimilar word pairs on the line printer.

FLOAD (RDOS)

Format

```
FLOAD MTn:tapefilenumber
```

Fast-load all files in tapefilenumber into the current directory. The files must have been FDUMPed (not DUMPed). To load any dumpfile other than 0, see FDUMP or the CLI manual.

If you FDUMPed to multiple tapes (FDUMP local /A), FLOAD will prompt for the next tape when it has loaded all the files on the current tape.

Global Switches

/L	Print loaded filenames on $LPT.
/N	Do not load files; display them on the console or send them to local /L file.
/V	Display loaded filenames on the console.

Local Switches

/L	Send filenames to this file.

Example

```
FLOAD/N MT0:0)
```

Display filenames in tapefilenumber 0 of tape on MT0; do not load them.

```
FLOAD/V MT0:0)
```

Fast-load all files in file 0; verify names on the console.

FORT

Format

FORT filename_1 ...

Compile and assemble FORTRAN IV source file filename(s).

FORTRAN IV compiler and runtime errors are described later in this book.

Global Switches

/A	Do not assemble the compiled file.
/B	List compiler source input only.
/E	Suppress compiler error messages.
/F	Equivalence compiler-generated FORTRAN labels and variable names to assembler-acceptable symbols.
/L	Produce listing to filename.LS.
/N	Do not produce a binary file.
/P	Process only 72 columns/record or characters/line.
/S	Save the intermediate source file.
/U	Output user symbols during assembly (must be used with /F).
/X	Compile statements with X in Column 1.

Local Switches

/B	Binary output filename (overrides /N).
/E	Error message file (overrides global /E).
/L	Listing file (overrides global /L).
/S	Intermediate source filename.

Examples

```
FORT/F/U/L DP1:TABLE)
```

Compile and assemble FORTRAN IV file TABLE (or TABLE.FR) in DP1, producing a binary version of TABLE. Include user symbols in the binary; send assembly listing to file TABLE.LS.

```
FORT MAIN $LPT/L)
```

Compile and assemble FORTRAN IV file MAIN (or MAIN.FR) and produce binary file MAIN.RB with a listing to the line printer.

FORTRAN (RDOS)

Format

FORTRAN filename$_1$ ...

Compile FORTRAN 5 source file(s), filename(s).

Global Switches

/B Brief listing (compiler source input).
/C Check syntax only.
/D Give line number and program name on all runtime error messages.
/I Do not list source lines from INCLUDE files.
/K Do not delete compiler temporary files after assembly.
/L Produce listing, filename.LS.
/P Process only 72 columns/record or characters/line.
/S Generate code for subscript checking.
/X Compile statements with X in column 1.

Local Switches

/B Relocatable binary output file.
/E Error message file.
/L Listing filename (overrides global /L).

Examples

```
FORTRAN/B MARK $LPT/L)
```

compile and assemble file MARK with output of binary MARK.RB and a brief listing on $LPT.

```
FORTRAN/D/S/L TEST)
```

Compile and assemble file TEST with output of binary TEST.LB and listing file TEST.LS. Perform subscript checking and list line number and program name on runtime errors.

FPRINT

Format

FPRINT filename *[oct/F] [oct/T]*

Print the contents of a disk file in octal, decimal, hexadecimal, or byte format. By default, the location counter is printed in octal, and output goes to the console.

Global Switches

/B Print in byte format.
/D Print in decimal.
/H Print in hexadecimal.
/L Print on line printer.
/O Print in octal (default).
/Z File starts at location zero (by default, FPRINT starts at location 16_8).

Local Switches

/F Print file starting at *oct* location preceding this switch.
/L List to given file. (Overrides global /L).
/T Last *oct* location to be printed.

Example

```
FPRINT/B/L MYFILE 2000/F 3500/T)
```

FPRINT MYFILE (from location 2000 to location 3500) in byte format on the line printer.

GDIR

Format

```
GDIR
```

Display the name of the current directory.

Examples

```
GDIR)
MYDIR2
```

The current directory is *MYDIR2.*

```
RELEASE %GDIR%)
```

Release the current directory.

GMEM (mapped RDOS)

Format

GMEM

Display the memory allocation for the foreground and background program sections in blocks of 2,048 bytes. Use SMEM to change this.

Example

```
GMEM)
BG:30   FG:34
```

61,440 bytes of memory (*30* times 2,048) are available for the background and 69,632 bytes are available for the foreground.

GSYS

Format

```
GSYS
```

Display the name of the current operating system.

Examples

```
GSYS)
SYS
```

The current system is named *SYS*.

```
R
MESSAGE THIS SYSTEM IS %GSYS%)
THIS SYSTEM IS SYS
R
```

Display the current system name.

GTOD

Format

```
GTOD
```

Display the current time of day and date.

Example

```
GTOD)
12/15/77  23:15:20
```

The time is given as 20 seconds after 11:15 p.m. on December 15, 1977.

INIT

Format

```
     ( tapedrive                          )
INIT <                                     >
     ( [directoryspecifier:] directory    )
```

Initialize a tapedrive or directory for I/O access. In RDOS, *directoryspecifier* can be:

dir *[:subdir]*

In DOS, *directoryspecifier* can be diskette.

You can omit *directoryspecifier* if directory is a direct subordinate to the current directory.

A file in a directory cannot be accessed until its directory and all superior directories are initialized.

Global Switches

/F Full initialization, clearing all information from the directory. This destroys all existing files. On a disk, INIT/F writes a new file directory and storage map. On a tape, INIT/F writes two EOFS (logical EOT) at the beginning.

Examples

```
INIT DP1:MYDIR)
```

All files in directory MYDIR and in directory DP1 are now accessible.

```
INIT/F MT1)
```

MT1 is rewound and two EOFs are written. (The system begins writing on tape at the double EOF.)

LDIR

Format

```
LDIR
```

Display the name of the last current directory (the directory last specified in a DIR command).

Example

```
GDIR)
DP0
R
DIR SUBDIR)
R
LDIR)
DP0
R
DIR %LDIR%; GDIR)
DP0
R
```

Change the current directory from *DP0* to SUBDIR; use LDIR to display the name of the last directory, then use the variable %LDIR% to return to the last current directory.

LFE

Format

LFE key libname$_1$ *[...libname$_n$] [rb$_l$ ... rb$_n$]*

Invoke the Library File Editor program to create, modify, update, or analyze library files, libnames, using existing libnames or relocatable binary files *(rbs)* within or oustside the libnames. LFE gives output libraries the names described under the keys below. You can specify a different output library name by inserting newname/O in the command line. LFE acts according to the function given by the key as follows:

Key Action

A Analyze global declarations of libname$_1$ or of binaries (*rbs*) of libname$_1$.

Analyze Switches:

Global:

/M Analyze the two or more library filenames following as one library.

Local:

rb/B rb is an .RB, not a library (.LB).

D Delete *rbs* from libname$_1$ producing library D.L1.

LFE (continued)

Key Action

I Insert *rbs* among *rbs* in libname$_1$, producing library I.L1.

Insert Switches:

Local:

rb/A Insert the rb that follows the switch immediately after this rb.

rb/B Insert the rb that follows the switch immediately before this rb.

M Merge libnames to produce output library M.L1.

N Produce new library file N.L1 from *rbs.*

R Replace binaries (rb_1, rb_3, ...) in libname$_1$ with binaries (rb_2, rb_4, ...) producing R.L1.

T List *rb* titles in libname, according to .TITL.

X Extract *rbs* from libname$_1$ as relocatable binary files *(rbs.RB)*. The *rbs* receive the original .TITL.

Local Switches

name/E Send error messages to file name.

name/F Start analysis of each *rb* on a new page (used with local /L).

name/L List to file name instead of to $LPT. (Used only in LFE A or LFE T lines.)

name/O Give output library file name. Overrides default output filenames in D, I, M, N, or R lines.

Examples

```
LFE A/M Z1.LB Z2.LB)
```

Analyze as one library Z1.LB and Z2.LB. Analyze listings go to the line printer by default.

```
LFE I M.LB M1.LB/O R/A HB.RB)
```

Insert relocatable binary HB.RB after relocatable binary R in library file M.LB. Name the new output library file M1.LB.

LINK

Format

```
LINK linkname resfilename
```

Create a link entry to another filename which may be a resolution file or another link entry. linkname is created in the current directory, unless you specify otherwise.

Local Switches

/2 The link entry name is the same as that of the entry being linked to (resfilename must be in the parent directory of the current directory).

Examples

```
LINK ASM.SV/2)
```

Create a link entry, ASM.SV, in the current directory to resolution file ASM.SV.

```
LINK EDIT.SV DPO:EDIT.SV)
```

Create a link entry in the current directory to the text editor in DPO.

```
LINK ALINK.SV DP1 :APROG.SV)
```

Create a link entry, ALINK.SV, in the current directory to file APROG in DP1. A linkname to a .SV or .OL file must have the .SV or .OL extension.

LIST

Format

LIST *[filename$_1$ ... filename$_n$]*

List information from any directory about its files or link entries. This can include file size, access attributes, link access attributes, file creation date and time, date last opened, file starting address and file use count.

For link entries, link entry names, directory, specifiers, and the resolution files are listed. An @ sign precedes the resolution filename if it is in the link directory's parent directory.

You can use template characters (* and -) to list files in the current directory.

Global Switches

/A	List all files, permanent and nonpermanent.
/B	List only the filenames.
/C	List creation time (mo/day/yr/hr:min).
/E	List all file information. (Overrides /B, /C, /F, /O, and /U .)
/F	List logical address of first block in file, or if unassigned, list 0.
/K	Do not list links.
/L	List to line printer.
/N	List links only.
/O	List date file last opened (mo/day/yr).
/S	Sort list alphabetically.
/U	List file use count.

LIST (continued)

Local Switches

/A List only files created or modified this date or after. The argument has the form: mm-dd-yy; mm and dd can be one or two digits.

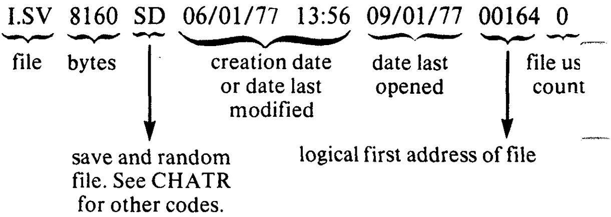
/B List only files created or modified before this date. The argument has the same form as local switch /A.

/N Do not list files matching this name.

Examples

```
LIST/E/A)
```

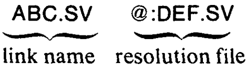
List all information for all files and link entries in the current directory on the console. A typical file entry would be:



```
I.SV 8160 SD 06/01/77 13:56 09/01/77 00164 0
```

file
bytes
creation date or date last modified
date last opened
file us count
save and random file. See CHATR for other codes.
logical first address of file

A typical link entry would be:



```
ABC.SV    @:DEF.SV
```

```
LIST/A/S -.SV 11/12/77/A)
```

List alphabetically all save files (.SV extensions) created on or after November 12, 1977.

LOAD

Format

LOAD inputfilename *[filename$_1$ ... filename$_n$]*

Load previously DUMPed files from inputfilename into the current directory. By default, all nonpermanent files are loaded. You can use global switches to load selected files (*filenames*) on all files, or simply to list files.

Template characters (* and -) are permitted.

Global Switches

/A	Load all files, including permanent ones.
/B	Brief listing.
/E	Suppress nonfatal error messages.
/I	Ignore checksum errors.
/K	Do not load links.
/L	List loaded files on the line printer (overrides /V and listing by /N).
/N	Do not load files; list the filenames on the console.
/O	Delete current file if it exists and replace with file being loaded having the same name.
/R	Load most recent version of file.
/V	Verify load names of files loaded on the console.

Local Switches

/A	Load only files created this date or later. (Preceding argument has the form mm-dd-yy, where mm and dd can be one or two digits.)
/B	Load only files created before this date. (Preceding argument has the same form as local /A).
/N	Do not load files matching this name.

LOAD (continued)

Examples

```
LOAD/N MT1:0)
```

Display nonpermanent filenames in tape file 0 of the tape on drive MT1; do not load files.

```
LOAD/A/L MT0:3 -.SV 5-15-77/A TEST-.SV/N)
```

Load from mag tape unit 0 tape file 3, all files with the .SV extension, except the files whose names begin with the characters TEST and files created before May 15, 1977; list the files on the line printer.

LOG

Format

LOG *[password] [directory/O]*

Open the log file, LOG.CM, and start recording CLI dialog. The name is FLOG.CM for a foreground CLI. *directory/O* can be used to open LOG.CM in a directory other than the current directory. *password* is up to ten alphanumeric characters and prevents LOG.CM from being closed inadvertently. LOG.CM can be closed only by an ENDLOG command containing the password (if it was given in the LOG command). LOG.CM can be printed or deleted only after it has been closed.

Global Switches

/H	Place a header at the beginning of the LOG file; include directory, system, and date information.
/T	Trace the execution of CLI commands, including indirects and macros. Before executing each command, the CLI outputs the command to LOG.CM. All trace lines are printed as ==> traceline.

Local Switches

/O	Output the file to this directory.

Example

```
LOG/H DELTA)
```

All CLI dialog is directed to LOG.CM in the current directory. ENDLOG DELTA) will be required to close the log file.

MAC

Format

MAC filename$_1$ *[...filename$_n$]*

Assemble one or more source files into relocatable binaries (RBs) using the macroassembler. You can create a save file from RBs with RLDR. See the error codes toward the end of this manual.

Global Switches

/A	Add semipermanent symbols to the cross reference listing.
/E	Suppress console error messages if there is a listing (global or local /L).
/F	Produce an even number of listing pages.
/K	Keep MAC.ST at the end of assembly (normally deleted).
/L	Produce a listing and cross reference.
/M	Flag multiply-defined symbols on pass 1.
/N	Do not produce a relocatable binary file.
/O	Override all listing suppression controls in source file(s).
/S	Skip pass 2 and retain as MAC.PS.
/T	Recognize symbol names of up to eight characters. This produces an *extended* .RB, which is described in the RLDR manual.
/U	Append user symbols to the RB file.

Local Switches

/B	Relocatable binary filename (overrides /N).
/E	Error message filename.
/L	Listing filename (overrides global /L).
/S	Skip this file on pass 2.
/T	Use this file as permanent symbol file for this assembly (normally, MAC uses MAC.PS).

Examples

```
MAC LIB/S A B C $LPT/L)
```

Assemble LIB, A, B, and C, producing A.RB. LIB contains macro definitions and is skipped on pass 2. Listing is output to the line printer.

```
MAC/L Z)
```

Assemble Z; produce listing to Z.LS and relocatable binary Z.RB.

MCABOOT

Format

MCABOOT/F MCAT *[1]* :n *[savefilenm/S] [filenm]*
MCABOOT MCAT *[1]* :n *[savefilenm/S]*

Transmit an operating system (or another program executable by BOOT) from one CPU to another via a multiprocessor communication adapter, MCAT or MCAT1.The adapter must be in the same network as the receiver n.The receiving CPU must be waiting for transmission.

By default, operating system SYS.SV/SYS.OL is transmitted. You can substitute another system or a user program by appending the local /S switch to the system or program, *savefilenm. savefilenm* must conform to BOOT conventions.

When doing a full initialization (/F global switch), the CLI files are also transmitted and you may transmit additional files (*filenms*). By default, partial initialization is performed and only one system or program can be transmitted.

Global Switches

/F Perform full initialization (this destroys all existing files on the receiving CPU's disk).

Local Switches

/S This is a user program or system other than SYS.SV/SYS.OL.

Examples

```
MCABOOT/F MCAT:2 FORT.SV FIV.SV FORT.LB)
```

Fully initialize CPU2's disk, and transmit a default RDOS system, CLI.SV, and CLI.OL to CPU2 in the first MCA system. Also send the FORTRAN IV compiler and library.

```
MCABOOT MCAT1:3 BASIC.SV/S)
```

Transmit a BASIC system to CPU3 on the second MCA network.

MDIR

Format

MDIR

Display (on the console) the name of the current master directory. This directory contains the system save and overlay files, the spool file (RDOS), and push space for program swaps.

Example

```
MDIR)
DP0
```

DP0 is the current master directory.

```
RELEASE %MDIR%)
```

Release the master directory.

MEDIT

Format

```
MEDIT terminals[clock-units]
```

Invoke the multiterminal Text Editor, which allows several text editing terminals to be active simultaneously. When invoked, MEDIT responds like the single-user editor, EDIT, and has a similar editing command repertoire (except that only the master console can terminate the program and return to the CLI). MEDIT commands are described later in this book.

The total number of terminals is given in decimal as follows: if **terminals** is given as 8 in the command line, terminal 0-7 may be accessed. *clock-units* gives the number of RTC clock units before the system will force task rescheduling.

Example

```
MEDIT 7)
```

Invoke the Multi-editor, and open terminals QTY:0 through QTY:6.

MESSAGE

Format

MESSAGE textstring

Display a message on the console, or insert a message for later display. You can use any ASCII character except carriage returns (↵) or semicolons (;) in textstring.

If you place textstring in quotes (“), it will be returned precisely as entered. If you omit quotes, the CLI will interpret certain characters (e.g., @) as commands; also other characters (including parentheses and angle brackets) are illegal. See the CLI manual for more detail.

Global Switch

/P Pause after displaying textstring and display the message “STRIKE ANY KEY TO CONTINUE”; the system will take no action until a console key is pressed.

Example

```
MESSAGE/P CURRENT DIRECTORY IS %GDIR%↵
CURRENT DIRECTORY IS DP1.
STRIKE ANY KEY TO CONTINUE
```

Here, the current directory is *DP1.* You might build this textstring into an indirect or macro file.

MKABS

Format

MKABS savefilename absbinfilename

Make an absolute binary file which can be loaded via the Binary Loader from a save file.

Global Switches

/S Use the starting address of savefile specified in USTSA as address of the absolute binary start block (default start block is null).

/Z Begin savefile at memory location zero; default is 16_8. (See RLDR global switch /Z).

Local Switches

/F First octal address, relative to savefile location zero, from which absolute binary file is to be created.

/S Give the absolute binary start block this octal address.

/T Last octal address, relative to the savefile location, to become a part of absolute binary file.

Example

```
MKABS/Z FSAVE $PTP 1000/S)
```

From save file, FSAVE, punch an absolute binary file with a starting address of 1000.

MKSAVE

Format

```
MKSAVE absbinfilename savefilename
```

Create a save file from an absolute binary file.

Global Switches

/Z Start save file at memory location 0 instead of 16_8. (See RLDR global /Z .)

Examples

```
MKSAVE/Z $PTR DPO:A)
```

From the absolute binary file in the tape reader, produce save file A.SV on DP0. Begin A.SV at location 0.

```
MKSAVE/Z MYPROGABS.SV MYPROG.SV)
```

Make absolute binary MYPROGABS.SV from save file MYPROG.SV. If MYPROG.SV was built according to the rules for stand-alone programs (BOOT command, CLI manual), MYPROGABS.SV can now be executed as a stand-alone program.

MOVE

Format

MOVE directory *[filename*$_1$ *...filename*$_n$ *]*

Copy a given file or files from the current directory to any directory, directory. *filename* cannot be a directory. You can pair *filenames* as follows:

oldfilename/S newfilename

newfilename will be the name of the entry in directory and oldfilename will be the name retained for the entry in the current directory. If you omit *filenames,* all nonpermanent files in the current directory are copied to the directory.

Global Switches

/A Move all files including permanent files.

/D Delete original files once the transfer is complete.

/K Do not move links.

/L List moved filenames on the line printer (overrides /V switch and console listing by /N).

/R Move a file only if it is more recent than the version in directory. Otherwise, do nothing with this file.

/V Verify moved files on the console.

MOVE (continued)

Local Switches

/A Move only files created or modified this date or after, where the preceding argument has the form: mm-dd-yy. mm and dd may be one or two digits.

/B Move only files created or modified before this date, where the argument has the same form as local switch /A.

/N Do not move files matching this name.

/S Assign a new name to the moved file but retain the old name in the current directory.

Example

```
MOVE NEWDIR JOE.SR/S JOEJR.SR A*.SR)
```

Copy the file JOE.SR to directory NEWDIR and rename it JOEJR.SR. Also, copy all nonpermanent files whose names have two characters starting with the letter A and the .SR extension.

```
GDIR)
DP1
R
MOVE/V DP1 MYFILE/S MYFILE.BU)
MYFILE.BU
R
```

Copy file MYFILE in the current directory under new name MYFILE.BU.

NSPEED

Format

NSPEED *[filename]*

Invoke the NOVA Supereditor program to edit ASCII text. Superedit allows multibuffer editing, multiple I/O files, macroprogramming and numeric variables. For ECLIPSEs, see SPEED. If the *filename* exists, Superedit opens it; if it doesn't exist, Superedit creates it.

Superedit commands are described later in this book.

Example

```
NSPEED MYFILE)
!                        SUPEREDIT prompt is !.
.
.                      \ You   can   issue   Superedit
.                      / commands.
UE$$
H$$                      H ESC ESC returns to the CLI.
R
```

OEDIT

Format

OEDIT filename

Invoke the octal editor to examine and modify the location of filename, which may be any disk file. You can edit in octal, ASCII or decimal. When invoked, the editor prompts you with a period (.) to indicate that it is ready for a command. Unless you specify otherwise, the editor is in octal.

You can determine the NMAX requirements for any save file or utility program by issing the command:

404-16/ *nnnnnn*

OEDIT returns *nnnnnn* which is the file's current NMAX.

To return to the CLI after using the octal editor, type ESC Z.

Example

OEDIT FOO.SV)	Invoke the octal editor.
.	
.	Issue commands.
.	
$Z	Return to the CLI.

OVLDR

Format

OVLDR savefilename $odscrip_1$ /N $olist_1$ ↑
[$odscrip_2$ /N $olist_2$ ...]

Create an overlay replacement file named savefilename.OR. This file holds one or more overlays that will replace overlays in savefilename.OL when a REPLACE command is issued. Overlay descriptor, odscrip, shows which overlay in savefilename.OL you wish to replace with the overlay list, olist, that follows. The overlay descriptor can be either a 16-bit representation of the overlay node and number or the overlay name (if ENTO was used in the overlay to give the overlay a name).

Global Switches

/A	Alphabetic/numeric memory map listing. (Local /L switch must also be given.)
/E	Output error messages. (To console by default or to file given by local /E .)
/H	Print output in hexadecimal.

Local Switches

/E	Error and information message file.
/L	Listing File.
/N	Must follow each overlay descriptor.

OVLDR (continued)

Examples

```
OVLDR/A/E Z 400/N M1 M2 401/N O X/L)
```

Create Z.OR with overlays M1 and M2 and O as future replacements for the first and second overlays for node 1 in Z.OL respectively. List in alphanumeric; send errors to file X.

```
OVLDR MYFILE OV1/N OV1A↑)
OV2/N OV2A)
```

Because the source program identified the overlays symbolically with the .ENTO pseudo-op, the octal odscrips were not necessary. MYFILE.OR contains the new overlays which will replace OV1 and OV2 when a REPLACE command is issued.

PATCH

Format

PATCH *[sfile/S] [pfile/P] [lmfile/L]*

Install patch(es) in a save or overlay file (sfile) on disk. Patchfile pfile contains the patches; pfile must have been created with ENPAT. If the patches contain symbols, there must also be a load map file (*lmfile*) available (SYSGEN or RLDR /Lswitch).

When it runs, PATCH creates a log file named sfile.PD,which describes the patches installed.

Global Switches

/I	Do not display comments from pfile on the console.
/N	There is no load map available (use this if you omit *lmfile/L*).

Local Switches

/L	Load map filename.
/P	Patch filename.
/S	Program save filename.

Example

```
PATCH SYS/S ERSPC/P SYS.LM/L)
1 APPLICABLE PATCH(ES)
1 PATCH(ES) NEEDED TO BE INSTALLED
R
```

Patchfile ERSPC was created in the ENPAT example. PATCH installs the patch in system SYS; the load map file for SYS is SYS.LM. (This wasn't really needed because the patch didn't use symbols.)

POP

Format

POP

Return to the next higher program level.

The CLI normally runs on level 0, the highest level. POP is useful when a program (which was on level 1 or below) has swapped in the CLI on level 2 or below. Programs can do this via system call .EXEC. If a program has swapped in the CLI on level 2, you can use the CLI normally, then type POP to resume execution of the program on level 1.

Example

See the CLI manual for an example.

PRINT

Format

PRINT filename$_1$ *[...filename$_n$]*

Print the contents of ASCII filename(s) on the line printer. To print a binary file, use FPRINT.

Example

```
PRINT FOO.SR DP4:COM.SR MT0:4)
```

Print source file FOO.SR (in the current directory), source file COM.SR (in DP4), and the file in MT0:4 on the line printer.

PUNCH

Format

PUNCH filename$_1$ *[... filename$_n$]*

Punch the contents of ASCII filename(s) on the high speed punch. To punch a binary file use BPUNCH.

Example

```
PUNCH DPO:A BETA $TTR)
```

Punch A (in DPO), file BETA (in the current directory), and the file on the teletypewriter reader on the high speed punch.

RDOSSORT (RDOS)

Format

RDOSSORT inputfilename$_1$ *[... inputfilename$_n$]* *[outputfilename/O]* key$_1$ *[... key$_n$] [arguments]*

Sort records from one input file, or merge records from up to 6 input files, according to specified keys.

The sort functions, which permit only one input file, are:

- Rearrange disk or tape records
- Delete disk or tape records
- Reformat records
- Produce sorted address file

The merge function merges records from up to six input files.

Data to be collated in inputfilename is ordered by key where at least one and up to 8 keys may be specified. These are compared against fields of the input records and are specified in the commands as:

b.f

where b is the starting byte number and f is the field length, e.g., 7.10 specifies a 10-byte key in positions 7-16 of the record. If you do not specify *outputfilename (/O),* no output file exists; this is useful if you want only a key file or listing. See *arguments* following.

arguments allow optional specification of the following:

work files (1-6) - /W allows you to name work files. By default they are named SORTWn.TP where n is 1-6.

RDOSSORT (continued)

output fields (1-8) - Specify a format for records in the output file. The format is the same as for specifying key except for use of a colon instead of a period (b:f). By default, the entire record is output.

output files - You may name files for listing (/L) (default is to the console), or for the names of sorted keys (/K) (default is no file).

input files - Use /S to specify a collating sequence; default is ascending ASCII. To specify a lower limit for the major key field, use /B; by default, all records are output. To specify an upper level for the major key field, use /U; by default, all records are output.

record size - /R allows you to specify in decimal the number of record bytes. Default is 80.

Global Switches

/D	Sort data in descending order. Default is ascending.
/M	Function to be selected is Merge. Default is Sort.
/N	Do not list sort or merge output. Default is a listing of the sort or merge.

Local Switches

/B	Name of file containing lower limit field.
/D	Delete the input file after sorting (overridden if no output file is specified).
/K	File giving names of sorted keys.
/L	Listing file.
/O	Output file of sorted records.
/R	Record size specifier in decimal.
/S	File giving collating sequence.
/U	File containing upper limit field.
/W	Work file.

Example

```
RDOSSORT MEMBERS.DA DGSORT/O↑)
COLLAT/S 61.12 141.10 $LPT/L 180/R↑)
1:30 61.60 121:30 DELIMIT/B DELIMIT/U)
```

The summary of statistics for this command might be:

```
RDOS Sort/Merge             06:43:07 07/02/77
PROGRAM                     :-Sort mode

Input filename(s)           :-MEMBERS.DA

Output filename             :-DGSORT

Record Size (Bytes)         :-180

Collating Sequence          :-User Specified

Sequence Filename           :-COLLAT

Sorted Key File             :-None Specified
  name

Lower Limit                 :-DELIMIT
  Filename

Upper Limit                 :-DELIMIT
  Filename

Input Field                 :-Start Byte/Length
  Specifiers                (bytes)
                                61-12
                                141-10

Output Field                :-Start Byte/Length
  Specifiers                (bytes)
                                1-30
                                61-60
                                121-30
```

RDOSSORT (continued)

Input File Records	:-3458
Sort In Record Count	:-3458
Sort Out Record Count	:-366

RELEASE

Format

$$\text{RELEASE}\begin{Bmatrix}\text{tapedrive}\\ \text{diskdirectory}\end{Bmatrix}$$

Logically remove an initialized tape drive or directory from the system. RELEASE tapedrive rewinds the tape on tapedrive. You must RELEASE a disk before physically removing it from its drive.

To shut down the system, release the master directory. Releasing the master directory releases all initialized directories and tape drives. In a dual-program environment, the master directory cannot be released from the foreground, or while the foreground is running.

Examples

```
RELEASE DP1)
```

Release DP1, and all initialized directories in it. You can then remove the disk from drive DP1.

```
RELEASE MT3)
```

Rewind the tape on MT3, and release drive MT3 from the system.

```
RELEASE %MDIR%)
MASTER DEVICE RELEASED
```

The variable %MDIR% contains the name of the master directory.

RENAME

Format

RENAME oldnm$_1$ newnm$_1$ *[... oldnm$_n$ newnm$_n$]*

Change the name of a file or files.

Example

```
RENAME DP1:A DP1:ALPHA B BETA)
```

Rename file A in DP1 to ALPHA, and rename B in the current directory to BETA.

REPLACE

Format

```
REPLACE savefilename
```

Replace one or more overlays in an existing file, savefile.OL, with one or more overlays in a replacement file, savefilename.OR, created by the overlay loader (OVLDR command).

Example

```
REPLACE MYPROGRAM)
```

Replace overlays in MYPROGRAM.OL as specified in the previous OVLDR command that created MYPROGRAM.OR.

REV

Format

REV filename *[.SV]*

Display the revision level of a save file. Save file filename must have the S attribute; you can omit the *.SV* extension.

Revision levels are assigned by the .REV assembler pseudo-op; they are displayed as a major level figure, followed by a period, and a minor level figure. Both major and minor figures can range from 0 to 99; the REV may include "PR" if this is a prerelease revision. If there is no .REV pseudo-op in the file, the level returned is 00.00. Some compiler programs assign *their* revision levels when they compile source files.

Examples

```
REV ATLC.SV)
ATLC.SV
03.07
R
```

Save file *ATLC* is at revision level *03.07*.

```
REV SYS)
SYS.SV
06.20
R
```

This system is revision *06.20*.

RLDR

Format

RLDR rb$_1$ *[..* [*orb... , ...*] *oct/s ... [orb$_n$ ...] rb$_n$]*

Load the relocatable binary files, rbs, and create a save file with the default name rb.SV. A corresponding overlay file rb.OL is created if you specify one or more relocatable binary files, *orbs,* as overlays. A left bracket ([ ) indicates the start of a new overlay node and *orbs* within the brackets are associated with that overlay node. A comma delimits overlays within the overlay segment on disk.

You may specify a number of octal values as arguments to RLDR with local switches.

Global Switches

/A Produce alphabetic symbol table (must have local /L also).

/B Use short (13-word) TCBs (unmapped NOVA multitask programs only).

/C Set INMAX to 440 (RTOS/SOS compatible; not executable under RDOS/DOS).

/D Load a symbolic debugger.

/E Output error messages to the console when a listing file (local /L) is specified.

/G Display warning messages about *all* new overlay common placed in root program (used with global /R).

/H Output numerics in hexadecimal.

/I Do not create UST or other system tables; start NREL at 445_8 and ZREL at 50_8. Program cannot execute under any DG system.

/K Keep the symbol table under rb.ST (normally, the table is deleted).

RLDR (continued)

/M Suppress the load map and all console output including error messages.

/N Do not search SYS.LB unless its name appears in the command line.

/O Omit the program symbol table although global /D was included.

/P Print the NREL value of each .RB loaded.

/R Place new common, declared in an overlay, in root program (if possible).

/S Leave the symbol table in high memory (used with /D).

/U Maintain a chain of undefined symbols (normally, all are resolved to -1).

/X Used by SYSGEN program.

/Y Used by SYSGEN program.

/Z Start the save file at location zero. (Program cannot execute under RDOS/DOS; use for stand-alone programs.)

Local Switches

/C Allot preceding octal number of channels.

/E Error message file.

/F Preceding octal is starting NREL address for execution in the foreground (unmapped RDOS only). (Loads are for execution in the background when you omit /F and /Z .)

/K Allot preceding octal number of tasks.

/L Send load map to this file.

/N Start loading the next file's NREL code at this octal address (must exceed present NMAX value).

/S Give the save and overlay files this name.

/U Include user symbols from this rb or *orb.*

/V Preceding brackets define a virtual overlay node (mapped RDOS only).

/Z Start all ZREL code at this octal address. (Also use this switch to load a program for execution in unmapped RDOS foreground.)

Examples

```
RLDR MYPROG MINE MYPROG.LM/L)
```

Create MYPROG.SV from RBs MYPROG and MINE; create file MYPROG.LM and send all messages and the load map to it.

```
RLDR R0 [A,B,C] R1 R2 [E,F G H, D])
```

Create save file R0.SV containing R0, R1 and R2. Also create overlay file R0.OL with two segments: A, B and C are overlays in the first segment (for node 0 in memory); E, FGH and D are overlays in the second segment (for node 1 in memory).

SAVE

Format

SAVE filename

Rename the break file to filename.SV. The system names the break file BREAK.SV (or FBREAK.SV for foreground). The break file contains the memory image of a program interrupted by a CTRL C break or by the debugger $V command.

Example

```
DEB A)
.
.
.
$V
BREAK
R

SAVE A)    Rename BREAK.SV to A.SV
R
```

SDAY

Format

```
SDAY month day year
```

Set the system calendar. You may give the year as either two or four digits (e.g., 77 or 1977).

You can separate arguments with either a space or a comma (but not a colon).

Example

```
SDAY 12 17 77)
```

Set the system calendar to December 17, 1977.

SEDIT

Format

SEDIT filename

Invoke the Symbolic Editor to examine and modify in octal, decimal, hex, or ASCII any location in a random or contiguous file.

You can specify a program's NMAX requirement by typing 404/ *nnnnnn.* SEDIT returns NMAX as *nnnnnn.*

To return to the CLI after using SEDIT, type ESC Z.

Global Switches

/N   Do not search for symbol table (or no symbol table exists with the file).

/Z   This file starts at location 0.

Example

```
SEDIT MYPROG.SV)
SEDITREVISIONx.x
.START+10/ 105433
.
.
ESC Z
DONE
R
```

SMEM (mapped RDOS)

Format

SMEM background

Change the number of 2,048-byte memory blocks allocated to the background and *foreground.* When you bootstrap RDOS, it gives all memory to the background. Use SMEM to create an area for *foreground* programs. You can issue SMEM from the background CLI only, while no foreground program is running.

Example

SMEM 24)

Allocate 24 2,048-byte blocks of memory to the background and all remaining memory to the foreground.

SPDIS (RDOS)

Format

```
SPDIS device-name₁ ...
```

Disable spooling on one or more spoolable devices. Normally, the spooling is enabled to the following device names: $DP0, $LPT, $LPT1, $PTP, $PTP1, $TTO, $TTO1, $TTP, and $TTP1. You can disable spooling on any of these by entering its device name in an SPDIS command. The plotter ($PLT, $PLT1) is spoolable, but is initially spool-disabled.

Example

```
SPDIS $LPT $PTP)
```

Disable spooling to the line printer and the paper tape punch. If output is currently being spooled to either device, the command takes effect after the current spool is completed.

SPEBL (RDOS)

Format

```
SPEBL device-name₁ ...
```

Enable spooling on a spoolable device. See SPDIS for the list of spoolable devices.

Example

```
SPEBL $LPT)
```

Resume spooling data output to the line printer.

SPEED (RDOS)

Format

SPEED *[filename]*

Invoke the ECLIPSE Supereditor to edit ASCII text. The Supereditor offers multibuffer editing, multiple I/O files, macro programming and numeric variables. For NOVAs, use NSPEED.

If the *filename* exists, Superedit opens it; if it doesn't exist, Superedit creates it.

Superedit commands are described later in this book.

Example

```
SPEED MYFILE)
!                 The Superedit prompt is !.
.               \
.                } You may now proceed with a
.               /  series of Superedit commands.
UE$$
H$$               The command H ESC ESC
                  terminates the Supereditor and
                  returns to the CLI.
R
```

SPKILL (RDOS)

Format

SPKILL devicename$_1$ ...

Delete the spool queue to devicename(s). See SPDIS for the list of spoolable devices. Spooling resumes at the next command to the device.

Example

```
SPKILL $LPT)
```

Stop spooling data to the line printer. Data on the output spool is lost.

STOD

Format

STOD *[hour] [minute] [second]*

Set the 24-hour system clock. You can separate arguments with either a space or comma (but not a colon).

Example

```
STOD 21 47 20)
```

Set the system clock to 9:47:20 p.m.

SYSGEN

Format

SYSGEN *[sysname/S] [loadmapfile/L]* ↑
[dialogfile/V] [tuningfile/T]

Generate a new RDOS or DOS system. The SYSGEN command depends on your system as follows:

System	Command
ECLIPSE with INFOS	ISYSGEN
Other ECLIPSE	BSYSGEN
Mapped NOVA 3	NSYSGEN
Other NOVA	SYSGEN

Before you generate an RDOS system, read the manual *How to Load and Generate Your RDOS System.* Before you generate a DOS system, read the appropriate section of the *DOS Reference Manual.*

SYSGEN asks questions about the hardware and software support you want in the new system; it then processes support modules from SYSGEN libraries with RLDR to produce the new system *(sysname)*. The *sysname* must not already exist. If you omit *sysname,* the system will be named SYS000.SV/SYS000.OL. The load map (which you can use for patching the system) will not be saved unless you specify a *loadmapfile.*

To execute the new system, type BOOT sysname.

SYSGEN (continued)

Global Switches

/N Do not build the system with RLDR (command line is in CLI.CM).

Local Switches

/A Generate the new system using answers in this dialog file. This file must have been specified with local /V in an earlier SYSGEN command.

/L Send the load map and messages to this file.

/S Give the finished system this name (always with .SV and .OL extensions).

/T Use this tuning file (RDOS) to generate a more efficient system (often used with local /A).

/V Save SYSGEN dialog in this file.

Examples

```
SYSGEN SYS.<SV/S SG/V> $LPT/L)
```

Generate a NOVA system named SYS.SV/SYS.OL; save the dialog in SYS.SG; send the load map to the $LPT.

```
NSYSGEN SYS64K.<SV/S SG/V LM/L>)
```

Generate the mapped NOVA 3 system SYS64K.SV/SYS64K.OL; save the dialog in SYS64K.SG, and the load map in SYS64K.LM.

TPRINT (RDOS)

Format

TPRINT *[sysname]*

Print the tuning file for *sysname.* You need not specify *sysname.*

If you omit switches, tuning file information is printed on the console and information on overlays is omitted.

Global Switches

/L	Print the tuning file on the line printer.
/O	Print the overlay frequency report.

Example

```
TPRINT/L SYS)
```

Print the tuning file SYS on the line printer.

TUOFF (RDOS)

Format

```
TUOFF
```

Stop recording in the tuning file. This command does not delete the tuning file.

Example

```
TUOFF)
```

Stop recording in the tuning file.

TUON (RDOS)

Format

```
TUON
```

Start recording, in the tuning file, the number of system requests and system failures for:

- any stack
- any cell
- any buffer
- any system overlay.

The tuning file has the name of the current operating system with the .TU extension. If it doesn't exist, TUON creates it.

Example

```
TUON)
```

If the current system is SYS, then the tuning file will be opened for recording as SYS.TU.

TYPE

Format

TYPE filename$_1$ *[...filename$_n$]*

Copy an ASCII filename(s) on the console. For a binary file, use FPRINT.

Example

```
TYPE A.SR B.SR DP1:XX.SR)
```

Type files A.SR and B.SR from the current directory and file XX.SR from DP1 on the console.

UNLINK

Format

UNLINK linkname$_1$ *[...linkname$_n$]*

Delete link entries from any directory. The resolution file is unaffected. You can use template characters (* and -) for links in the current directory.

Global Switches

/C Display each link name on the console and wait for confirmation that the link is to be unlinked. Type RETURN to remove the link entry or any other key to retain the link entry.

/L List deleted files on $LPT (overrides /V).

/V Verify deletions on the console.

Local Switches

/N Do not delete links matching this name.

Examples

```
UNLINK/V MYLK JOELK TESTLK.-)
MYLK
JOELK
TESTLK
TESTLK.1
```

Delete from the current directory: link entries MYLK, JOELK, and all TESTLKs with any extensions. Verify deletions on the console.

```
UNLINK/C -.SV -.LB DP1:ASM.SV)
```

Repeat each link entry in the current directory having a .SV or .LB extension, and link entry ASM.SV on DP1; wait for a carriage return before deleting the entry.

VFU (RDOS)

Format

VFU *[/S]* filename

Create, edit, or load a VFU file into a data channel line printer's memory. You direct VFU's action with global switches. If you omit switches, VFU tries to load VFU filename into the first line printer's memory. See the CLI manual for more detail.

Global Switches

/A Allow user programs to access the line printer's memory (used without an argument).

/C Create VFU file filename. VFU then asks questions about the tab and VFU control you want in filename (see the example below).

/D Disable user program access to the printer's memory.

/E Edit existing VFU file filename. VFU allows you to set or clear tab stops and VFU control entries.

/L Print existing filename settings on $LPT.

/S Second printer. Load into or list on $LPT1 (depending on other switches).

/V Display filename's settings on the console.

/X Load filename into the first line printer's memory (you can omit all switches to do this).

Local Switches

/L Send filename settings to this file.

Example

```
VFU/C ACCOUNTS8)

CREATING ACCOUNTS8. VF
TAB CONTROL:
WANT STANDARD TABS (EVERY 8 COLUMNS)?
ENTER Y/N  Y
VFU CONTROL:
WANT STANDARD (11 INCH)?
ENTER Y/N  Y
ENTER FORM SIZE IN LINES (1-143)  66)
ENTER LINE NUMBER OR CR  1)
ENTER CHANNEL NUMBER (1-12)  1)
ENTER LINE NUMBER OR CR  63)
ENTER CHANNEL NUMBER (1-12)  12)
ENTER LINE NUMBER OR CR )
R
```

This creates ACCOUNTS8.VF with standard VFU settings. To load ACCOUNTS8.VF into the first printer's memory, you'd type:

```
VFU ACCOUNTS8)

PREPARE TO LOAD ACCOUNTS8. VF
WAIT UNTIL ALL OUTPUT TO THE PRINTER
HAS COMPLETED. MAKE SURE PRINTER IS
READY AND ON-LINE.
STRIKE ANY KEY WHEN READY.
(strike key)
R
```

XFER

Format

XFER sourcefilename destinationfilename

Copy the contents of sourcefilename to destinationfilename, organizing the destination file differently as indicated. If you omit switches, the destination file will be sequentially organized (RDOS) or randomly organized (DOS). In RDOS, append local /R to destinationfile to XFER a save file. In both RDOS and DOS, you must CHATR destinationfile +S after the XFER if destinationfile is a save file.

Global Switches

/A Perform an ASCII transfer.

/B Append the source file to the destination file.

Local Switches

/C Organize the destination file contiguously (both must be disk files).

/R Organize the destination file randomly (must be on disk).

Examples

```
XFER MT0:2 MYFILE.SV/D)
R
CHATR MYFILE.SV +S)
```

Copy the contents of file 2 of the tape on MT0 into the current directory, under the name MYFILE.SV. Organize it randomly, then give it the S attribute so that it can execute.

```
XFER/A $PTP ALPHA.SR)
```

Copy the tape in the paper tape reader, in ASCII, to disk file ALPHA.SR.

```
R
XFER/A $TTI BYE.MC)
(PRINT, DELETE) LOG.CM; DISK;)
MESSAGE GOODBYE AT %TIME%)
RELEASE %MDIR%)
CTRL-Z
R
```

This file, named BYE, consists of several CLI commands input from the console. It can be executed by the command:

```
BYE
```

End of CLI Commands

CLI Error Messages

The following CLI error messages are listed alphabetically. Where the meaning of the message may not be clear, an explanation follows the error message text. See the CLI manual for full explanations.

A ZERO .XMT OR .IXMT MESSAGE

Program tried to transmit a zero word.

ADDRESS ERROR IN .SYST ARGUMENT

Program tried to access address outside user space.

ALM LINE NOT READY

ATTEMPT TO CREATE A ZERO LENGTH CONTIGUOUS FILE

ATTEMPT TO READ INTO SYSTEM SPACE

Program tried to access system space.

ATTEMPT TO RELEASE AN OPEN DEVICE

ATTEMPT TO RESTORE A NONEXISTENT IMAGE

POP from level 0.

ATTEMPT TO WRITE AN EXISTING FILE

BLANK TAPE

You forgot to INIT/F the new tape.

CANNOT CHECKPOINT CURRENT BG

Causes can be: outstanding MUX I/O; user device interrupt service; outstanding read/write operator message, etc. Also, if one BG program is already checkpointed, the current BG program cannot also be checkpointed.

CHANNEL ALREADY IN USE

CHANNEL CLOSED BY ANOTHER TASK

CHECKSUM ERROR

COMMAND LINE TOO LONG

COMMON SIZE ERROR

Interprogram communications area too small.

COMMON USAGE ERROR

Other program has no defined communications area.

CONSOLE INTERRUPT RECEIVED

From a multiplexed line on a read or write call.

DEVICE ALREADY IN SYSTEM

You tried to re-initialize a directory/device or a system device as a user device.

DEVICE NOT IN SYSTEM

You tried to access an uninitialized device or directory.

DEVICE PREVIOUSLY OPENED

DEVICE TIMEOUT

DIRECT I/O ACCESS ONLY

DIRECTORY DEPTH EXCEEDED

You tried to LOAD (FLOAD) a DUMPed (FDUMPed) directory into an equivalent current directory, or create a directory in an equivalent directory.

DIRECTORY IN USE

You tried to release a directory whose files are not closed.

DIRECTORY NOT INITIALIZED

You tried to release an uninitialized directory or device.

DIRECTORY SHARED

Released directory in use by another program. (Warning message only.)

DIRECTORY SIZE INSUFFICIENT

DUPLICATE READ OR DUPLICATE WRITE

To same multiplexed line.

DISK FORMAT ERROR

If INIT or DIR succeeded, dump the disk and run the disk initializer; the disk may need reformatting.

END OF FILE

ERROR:message

From PATCH command. Usually, this means an invalid patchfile. Create new file with ENPAT.

ERROR IN USER TASK QUEUE TABLE

Bad input to .QTSK.

EXEC ERROR ON CHAIN

CLI will restart.

FATAL OUTPUT ERROR

CLI will restart.

FATAL SYSTEM UTILITY ERROR

Bad input to utility, or missing a utility file.

FILE ALREADY EXISTS

FILE ATTRIBUTE PROTECTED

FILE DATA ERROR

If on tape, this could mean dirty heads. On disk, it means a checksum error.

FILE DOES NOT EXIST

In the current directory.

FILE IN USE

FILE NOT OPEN

FILE POSITION ERROR

Program tried to .SPOS to an illegal position in a file.

FILE READ PROTECTED

FILE SPACE EXHAUSTED

FILE WRITE PROTECTED

FILES MUST EXIST IN SAME DIRECTORY

FOREGROUND ALREADY RUNNING

You can terminate the FG with CTRL-F.

ILLEGAL ARGUMENT

Illegal character in argument.

ILLEGAL ATTRIBUTE

ILLEGAL BLOCK TYPE

Attempt to XFER a non- XFER file, LOAD a non-DUMPed file, or FLOAD a non- FDUMPed file.

ILLEGAL CHANNEL NUMBER

ILLEGAL COMMAND FOR DEVICE

Attempt to perform illegal I/O, e.g., direct I/O on disk data.

ILLEGAL DIRECTORY NAME

Your link command has illegal character(s) in the resolution file specifier.

ILLEGAL FILE NAME

ILLEGAL NUMERIC ARGUMENT

ILLEGAL OVERLAY NUMBER

Overlay does not exist.

ILLEGAL PARTITION VALUE

On SMEM: (1) not enough space for BG CLI; (2) SMEM not issued from BG level 0 CLI.

ILLEGAL SYSTEM COMMAND

This RDOS/DOS system does not support your command.

INSUFFICIENT CONTIGUOUS BLOCKS

To load or create contiguous file.

INSUFFICIENT MEMORY TO EXECUTE PROGRAM

INSUFFICIENT ROOM IN DATA CHANNEL MAP

Improper data channel size in .IDEF.

INVALID BAD BLOCK TABLE

Run disk initializer program.

INVALID TIME OR DATE

LINE TOO LONG

132 characters exceeded on line read/write.

LINK ACCESS NOT ALLOWED

N attribute set for resolution file.

LINK DEPTH EXCEEDED

More than 10 levels of links.

MAP.DR ERROR

Dump disk if possible; INIT/F it.

MT OP ER n

You specified wrong numbers for files stacked on one tape (FDUMP/FLOAD).

MTn NOT READY

Put tapedrive ON-LINE (FDUMP/FLOAD).

MCA REQUEST OUTSTANDING

This MCA line is in use.

NO DEBUG ADDRESS

Include global /D in RLDR command.

NO DIRECT I/O

NO FILES MATCH SPECIFIER

No match on argument containing * or -.

NO MCA RECEIVE REQUEST OUTSTANDING

NO MORE DCBS

You can (1) release one or more currently initialized directories or (2) SYSGEN a new system, specifying a larger number of subdirectories/subpartitions accessible at a time.

NO ROOM FOR UFTS

Not enough channels specified at SYSGEN time.

NO SOURCE FILE SPECIFIED

NO STARTING ADDRESS FOR LOAD MODULE

RLDR command. Specify an address after a .END pseudo-op.

NO SUCH DIRECTORY

NOT A COMMAND

Fatal CLI error due to modification of CLI.SV or CLI.OL.

NOT A LINK ENTRY

NOT A SAVE FILE

Program must have SD attribute and .SV extension.

NOT ENOUGH ARGUMENTS

name OPEN ERR-FILE NOT DUMPED

File was open -- could not be dumped (FDUMP).

OUT OF TCB'S

Specify more tasks in the program (RLDR or COMM TASK).

PARITY ERROR

If on tape, try cleaning heads.

PERMANENT FILE

PHASE ERROR

Attempt to reset location counter back over current value (e.g., in MKSAVE, setting the counter ahead is permitted).

PROGRAM NOT SWAPPABLE

The program specified in .EXEC can't be swapped.

PUSH DEPTH EXCEEDED

Attempt to swap from level four.

QTY ERROR

Simultaneous read or write to mux line.

name READ-LOCKED-NOT DUMPED

File can't be read, thus can't be dumped (FDUMP).

RDOS ERROR

CLI can't interpret the error code in AC2. See the CLI manual.

READ FRAMING ERROR

Multiplexor hardware framing error.

READ OVERRUN ERROR

Multiplexor hardware overrun error.

SIGNAL TO BUSY ADDRESS

SPOOL FILES ACTIVE

You tried to BOOT or RELEASE the master device while spooling was active. Wait, or use SPKILL to delete the spool queue.

STACK OVERFLOW

Fatal error. Often caused by modification of CLI.OL or by nonmatching CLI.SV and CLI.OL creation dates. May be caused by attempt to replace CLI while running. The CLI will restart.

SYNTAX ERROR:message

Unmatched characters in command line. Correct command.

SYS.DR ERROR

Dump disk if possible; INIT/F it.

SYS ERR RTN OFFSET nnnnnn

Release drive and try command again (FDUMP/FLOAD).

SYSTEM DEADLOCK

Out of buffers. Hit CTRL-A and wait for the task or program to finish its I/O.

SYSTEM STACK OVERFLOW

TAPE HAS WRONG REEL NO.

Mount the correct FDUMPed reel (FLOAD).

TASK ID ERROR

TASK NOT FOUND FOR ABORT

TEXT ARGUMENT TOO LONG

Message textstring exceeds 72 characters.

TOO MANY ARGUMENTS

TOO MANY SOFT ERRORS

The DOS diskette error count has reached maximum. Use the DOSINIT copy command.

.TOVLD NOT LOADED FOR QUEUED OVERLAY TASKS

Insert a .EXTN .TOVLD statement in the program; re-assemble and reload it.

TRANSMISSION TERMINATED BY RECEIVER

UNIT IMPROPERLY SELECTED

A tape controller or disk unit adapter is not turned on or a tape unit is not on-line.

VIRTUAL BUFFER FILE ERROR

Fatal CLI error, relating to CLI temporary files. The CLI will restart.

YOU CAN'T DO THAT

Attempt to end console logging without giving the password; or any grossly bad input to the CLI (e.g., lowercase ASCII characters); or an attempt to run an ECLIPSE program on a NOVA computer.

End of CLI Error Messages

RDOS/DOS Calls

You can program commands to RDOS or DOS in assembly source code. There are two kinds of system commands: system calls; which you precede with the mnemonic .SYSTM; and task calls, which do not use this mnemonic. System calls are shown with the introduction .SYSTM in the following table.

For most commands, your program must pass a value to RDOS/DOS in an accumulator - generally this is AC0, and occasionally AC1. The system generally uses AC2 to return error codes. In the following table, commands that do not take an error return are marked with a dash (-) under AC2; commands that take neither an error return nor a normal return are marked with a double dash (--) under AC2.

Some commands require you to pass a channel number to the system. All such commands have an n (for channel number) in column one following the command. Other exceptions to passed and returned values in AC2 are noted in the list of commands.

Values that are returned in AC0, AC1, or AC2 are shown as shaded values in the appropriate column. If the left and right bytes of a single data word contain different information, they are separated by a slash; e.g., block offset/slot offset.

Enter all numbers (abbreviated *no.*) in octal.

Command	AC0	AC1	AC2
.ABORT		Task ID	
.AKILL	Priority of tasks		-
.SYSTM .APPEND n	Bptr to filename	Device characteristic mask	
.ARDY	Priority of tasks		-
.ASUSP	Priority of tasks		-
.SYSTM .BOOT	Bptr to opsys name		
.SYSTM .BREAK			
.SYSTM .CCONT	Bptr to filename	Integer no. of disk blocks	
.SYSTM .CDIR	Bptr to dirname		
.SYSTM .CHATR n	File attributes		
.SYSTM .CHLAT n	File attributes		
.SYSTM .CHSTS n	Start address of data area		
.SYSTM .CLOSE n			
.SYSTM .CONN	Bptr to filename	No. of disk blocks	

Command	AC0	AC1	AC2
.SYSTM .CPART	Bptr to partition name	No. of contiguous blocks	
.SYSTM .CRAND	Bptr to filename		
.SYSTM .CREAT	Bptr to filename		
.SYSTM .DDIS	Device code		
.SYSTM .DEBL	Device code		
.SYSTM .DELAY		No. of RTC ticks	
.SYSTM .DELET	Bptr to filename		
.SYSTM .DIR	Bptr to dirname		
.DQTSK		Task ID	▨ *
.DRSCH			▨ -
.SYSTM .DUCLK	No. of RTC ticks	Address of user interrupt routine	
.SYSTM .EOPEN n	Bptr to filename	Inhibit mask	
.SYSTM .EQIV	Bptr to global specifier	Bptr to temporary specifier	

* Returns base address of released area.

Command	AC0	AC1	AC2
.SYSTM .ERDB n	Blk offset/slot offset	Starting relative blkno of file	*
.ERSCH			-
.SYSTM .ERTN			@
.SYSTM .EWRB n	Blk offset/slot offset	Starting relative blkno of file	*
.SYSTM .EXBG	Bptr to BG program	New BG priority	#
.SYSTM .EXEC	Bptr to filename	Swap/chain options	#
.SYSTM .EXFG	Bptr to filename	Priority/ control options	#
.SYSTM .FGND	0=no FG 1=FG	Program level code	
.SYSTM .GCHAR	Character (bits 9-15)		

* You pass the number of 256-word blocks to be read or written in bits 0-7. Passing n in AC2 is optional unless n is 77. If it is 77, pass 77 in bits 8-15. If there is an EREOF error on return from .ERDB, a partial read is returned in bits 0-7 and the error in 8-15.

@ Lower-level program's error status word.

Gives message to new program.

Command	AC0	AC1	AC2
.SYSTM .GCHN			*
.SYSTM .GCIN	Bptr to input console name		
.SYSTM .GCOUT	Bptr to output console name		
.SYSTM .GDAY	Day	Month	Yr
.SYSTM .GDIR	Bptr to dirname area		
.SYSTM .GHRZ	RTC freq		
.SYSTM .GMCA	MCA device code	Unit number	
.SYSTM .GPOS n	Bptr (high order word)	Bptr (low order word)	
.SYSTM .GSYS	Bptr to area for sysname		
.SYSTM .GTATR n	File attributes	Device characteristics	
.SYSTM .GTOD	Seconds	Minutes	Hrs
.SYSTM .ICMN	Starting address common area	Size of area in words	

* Returns free channel number.

Command	AC0	AC1	AC2
.SYSTM .IDEF	Device code	Address of device DCT	*
.IDST	Task Status	Task ID (right byte)	@
.SYSM .INIT	Bptr to directory specifier	-1=full 0=partial	
.SYSTM .INTAD			
.IOPC	Queue area address	Overlay chanl./ Maxm queues	#
.SYSTM .IRMV	User device code		
.IXMT	Message address	Message	
.KILAD	Kill processor address		-
.KILL			--
.LEFD	(Contents lost on return)		-
.LEFE	(Contents lost on return)		-

* Number of 1K blocks of memory needed by the data channel map (mapped RDOS only).

@ Returns TCB base adress (no error return exists).

# Program table address.

Command	AC0	AC1	AC2
.LEFS	User status word		-
.SYSTM .LINK	Bptr to linkname	Bptr to string or 0	
.SYSTM .MAPDF	No. blks for ext'd address	Start blkno of window	*
.SYSTM .MDIR	Bptr to device name area		
.SYSTM .MEM	HMA	NMAX	
.SYSTM .MEMI	NMAX increment	New NMAX	
.SYSTM .MTDIO n	Memory address if data xfer	Command word fields	@
.SYSTM .MTOPD n	Bptr to tape global specif.	Disable mask or 0	
.MULTI			
.SYSTM .ODIS			
.SYSTM .OEBL			
.SYSTM .OPEN n	Bptr to filename	Disable mask or 0	

* Window Size in 1K blocks.

@ System error code or transport status word.

Command	AC0	AC1	AC2
.OVEX	Node no./ overlay no.		*
.OVKIL	Node no./ overlay no.		
.SYSTM .OVLOD n	Node no./ overlay no.	0=Cond. load -1=uncond. load	
.SYSTM .OVOPN n	Bptr to overlay filename		
.OVREL	Node no./ overlay no.		
.SYSTM .OVRP	Bptr to overlay replacement file	Bptr to overlay file	
.SYSTM .PCHAR	character (bits 9-15)		
.PRI	New priority (right byte)		-
.QTSK			@
.SYSTM .RDB n	Starting address for data	Starting file relative blkno	#

* Contains return address on normal return.

@ Starting address of User Task Queue Table.

Pass the number of blocks to be read in bits 0-7, and the channel number (if n =77) in 8-15. If an EREOF error occurs, bits 0-7 return the partial read count and bits 8-15 contain the error code.

Command	AC0	AC1	AC2
.SYSTM .RDCM	Word address to read into	Offset into communications area	*
.SYSTM .RDL n	Bptr to user memory area	Byte count read	
.SYSTM .RDOP	Bptr to message area	Byte count	
.SYSTM .RDR n	Memory address to receive record	Record number	
.SYSTM .RDS n	Bptr to memory buffer	Bytes to read @	
.SYSTM .RDSW	Console switch position		
.REC	Message address	Message	-
.REMAP	Contents lost upon return	Blkno of map/ blkno of window	#
.SYSTM .RENAM	Bptr to old name	Bptr to new name	
.SYSTM .RESET			

* Word count.

@ Returns partial read count of EOF.

Number of 1K blocks to be remapped. (Note that .REMAPdestroys AC0 and AC1.)

Command	AC0	AC1	AC2
.SYSTM .RLSE	Bptr to dir/device specifier		
.SYSTM .ROPEN n	Bptr to filename	Inhibit mask	
.SYSTM .RSTAT	Bptr to filename	Starting address of UFD	
.SYSTM .RTN			
.SYSTM .RUCLK			
.SYSTM .SDAY	Day	Month	Yr
.SINGL			
.SMSK		New interrupt mask	
.SYSTM .SPDA	Bptr to device name		
.SYSTM .SPEA	Bptr to device name		
.SYSTM .SPKL	Bptr to device name		
.SYSTM .SPOS n	High order of bptr	Low order of bptr	
.SYSTM .STAT	Bptr to filename string	Start address to receive UFD	

Command	AC0	AC1	AC2
.SYSTM .STMAP	Device code	Start address of device buffer	
.SYSTM .STOD	Seconds	Minutes	Hrs
.SUSP			-
.TASK	Task ID/ Task priority	Task entry point address	*
.TIDK		Task ID	
.TIDP		Task ID	
.TIDR		Task ID	
.TIDS		Task ID	
.TOVLD	Node no./ overlay no.	0=cond. load -1=uncond. load	@
.TRDOP	Bptr to message area	Byte count	
.SYSTM .TUOFF			
.SYSTM .TUON	Zeroed		
.TWROP	Bptr to message area	-1 to suppress task ID	

* Message to new task.

@ Channel no. on which overlays were .OVOPNed.

Command	AC0	AC1	AC2
.UCEX[1]		Nonzero value if rescheduling	--
.UIEX[2]		Nonzero value if rescheduling	--
.SYSTM .ULNK	Bptr to link entry		
.SYSTM .UPDAT n			
.UPEX[2]		Nonzero value if rescheduling	--
.SYSTM .VMEM	No. of available blocks		
.SYSTM .WRB n	Starting memory address	Starting relative blk no.	*
.SYSTM .WRCM	Word address of message	Offset into communications area	@

[1] Input the return address in AC3 (unmapped systems only).

[2] Input return address in AC3 in unmapped systems. (Not required in mapped systems.)

* You must pass the number of disk blocks to be written in bits 0-7, and the channel number, if n = 77, in bits 8-15. If an ERSPC error occurs, bits 0-7 return the partial write count, and bits 8-15 return the error code.

@ Word count.

Command	AC0	AC1	AC2
.SYSTM .WREBL	Starting address of series	Ending address of series	
.SYSTM .WRL n	Bptr to memory buffer	Write byte count	
.SYSTM .WROP	Bptr to text string		
.SYSTM .WRPR	Starting address of series	Ending address of series	
.SYSTM .WRR n	Memory address of record	Record number	
.SYSTM .WRS n	Bptr to memory buffer	No. of bytes to be written	*
.XMT	Message address	Message	
.XMTW	Message address	Message	

* Input the number of MCA retries in bits 0-7 and the channel number, if n =77, in bits 8-15.

End of RDOS/DOS Calls

Codes Returned in AC2 from RDOS/DOS Errors

Number	Mnemonic	Meaning
0	ERFNO	Illegal channel number.
1	ERFNM	Illegal filename.
2	ERICM	Illegal system command.
3	ERICD	Illegal command for device.
4	ERSV1	File needs save attribute and random characteristic.
6	EREOF	End of file.
7	ERRPR	File read protected.
10	ERWPR	File write protected.
11	ERCRE	Attempt to alter permanent file.
12	ERDLE	File does not exist.
13	ERDE1	Attempt to alter permanent file.
14	ERCHA	Illegal attempt to change file attributes.
15	ERFOP	Reference of unopened file.
16	ERFUE	Fatal utility error.
17	EREXQ	Instructs RDOS to execute CLI.CM on return to the CLI.

Name	Mnemonic	Meaning
20	ERNUL	Null error code.
21	ERUFT	Channel already in use.
22	ERLLI	Line limit exceeded.
23	ERRTN	Image nonexistent.
24	ERPAR	Parity error on read line or tape.
25	ERCM3	Trying to push too many levels.
26	ERMEM	No more memory available.
27	ERSPC	Out of disk or tape space.
30	ERFIL	File read error-bad tape.
31	ERSEL	Unit improperly selected.
32	ERADR	Illegal starting address.
33	ERRD	Attempt to read into system area (unmapped system only).
34	ERDIO	Direct block I/O on sequential file.
35	ERDIR	Files specified in different directories (RENAM).
36	ERDNM	Device not in system or illegal device code.
37	EROVN	Illegal overlay number.

Name	Mnemonic	Meaning
40	EROVA	File not accessible by direct I/O.
41	ERTIM	Attempt to set illegal time/date.
42	ERNOT	Out of TCBs.
43	ERXMT	Message address in use.
45	ERIBS	Device already in system.
46	ERICB	Not enough free contiguous disk blocks.
47	ERSIM	Simultaneous read/write attempt on same QTY or ALM line.
50	ERQTS	Illegal information in task queue table.
51	ERNMD	Attempt to open too many devices or directories.
52	ERIDS	Illegal directory specifier.
53	ERDSN	Directory specifier unknown.
54	ERD2S	Partition is too small (RDOS).
55	ERDDE	Directory depth exceeded.
56	ERDIU	Directory in use.
57	ERLDE	Link depth exceeded.
60	ERFIU	File is in use.

Name	Mnemonic	Meaning
61	ERTID	Task ID error.
62	ERCMS	Communication area size error.
63	ERCUS	Communication area usage error.
64	ERSCP	File position error.
65	ERDCH	Insufficient room in data channel map.
66	ERDNI	Directory/device not initialized.
70	ERFGE	Foreground already exists (RDOS).
72	EROPD	Attempt to release a directory in use by other program.
73	ERUSZ	Not enough room for UFTs in USTCH.
74	ERMPR	Address outside address space (mapped system only).
75	ERNLE	Attempt to unlink entry lacking link characteristic.
76	ERNTE	Program not checkpointable.
77	ERSDE	Error detected in SYS.DR.
100	ERMDE	Error detected in MAP.DR.
101	ERDTO	Device timeout.

Name	Mnemonic	Meaning
102	ERENA	Link not allowed.
103	ERMCA	No complementary MCA request.
104	ERSRR	Short MCA receive request.
106	ERCLO	Channel closed by another task.
107	ERSFA	Spool files are active.
110	ERABT	Task abort is not allowed.
111	ERDOP	Mag tape or cassette unit already open.
112	EROVF	System stack overflow (current command aborted).
113	ERNMC	No MCA receive request outstanding.
114	ERNIR	Attempt to release tape unit with currently open file.
115	ERXMZ	Attempt to transmit a zero word message.
117	ERQOV	.TOVLD not loaded for overlay task.
120	EROPM	Operator messages not specified at SYSGEN.
121	ERFMT	Disk format error.
122	ERBAD	Disk has invalid bad block table.

Name	Mnemonic	Meaning
123	ERBSP	Not enough space for memory bad block pool.
124	ERZCB	Attempt to create a zero-length contiguous file.
125	ERNSE	Program is not swappable.
126	ERBLT	Blank tape.
127	ERRDY	MUX line not ready.
130	ERINT	MUX line console interrupt received.
131	EROVR	MUX line hardware overrun error.
132	ERFRM	MUX line hardware framing error.
133	ERSPT	Too many soft errors on DOS diskette.

End of RDOS/DOS Error Messages

Bootstrapping RDOS

For all machines except microNOVAs (see next section for microNOVAs):

1. Power up equipment and insert a system disk in drive 0 (if this applies). Then flip the drive to RUN and wait for the READY light.

2a. If your computer has the Program Load feature:

- To bootstrap from a disk on the first controller, enter $1000nn_8$ in the data switches. nn depends on the disk model, as follows:

Disk Model No.	**nn**	**Switches Up**
6063/6064	26	0, 11, 13, 14
6001-6008	20	0, 11
6060-6061	27	0, 11, 13, 14, 15
others	33	0, 11, 12, 14, 15

- Press RESET, then PROGRAM LOAD (on a SUPERNOVA, press CHANNEL START). Proceed to step 3.

2b. If your computer *lacks* automatic Program Load hardware:

- Set the data switches to 000376_8 (switches 8 through 14 up, others down) and lift EXAMINE.

- For a disk on controller 1, set switches to $0601nn_8$ as above, don't set switch 0; set 1, 2, 9 and nn switches. Press DEPOSIT.

- Set switches to 00377_8 (8 through 15 up, others down), and press DEPOSIT NEXT.

- Set switches to 000376_8 (8 through 14 up, others down); lift RESET, then START. Go to step 3.

3. BOOT.SV now displays:

```
FILENAME?
```

Type the name of your RDOS system save and overlay files. If the system is not in DP0, DS0, DK0, or DZ0, type the directory specifier before its name; e.g., DPOF:SYS1 If it is in DP0, DS0, DK0 or DZ0 and has the default name (SYS), you can simply press RETURN, without typing the name.

The directory which holds the specified system becomes the master directory for this session. RDOS now displays its name, and asks date/time questions. Answer these to receive the CLI prompt "R".

Signing Off RDOS

To sign off RDOS, type:

```
RELEASE masterdirectoryname
```

or

```
RELEASE %MDIR%
```

Now, turn off the power to all equipment.

End of Bootstrapping RDOS

Bootstrapping DOS (microNOVAs)

This applies to microNOVAs only; for DOS on other machines, see the previous section.

1. Power-up equipment; insert the system diskette in drive 0; turn drive 0 ON.

2a. If you have a hand-held console, press RESET, CLR D, enter 33 in the display, and press PR LOAD. Go to step 3.

2b. If you are using the console option to the terminal controller board, type 33L. Go to step 3.

2c. If you have the CPU program load option, set the jumpers for code 33.

Press the front panel rocker switch to the PL/START position. Go to step 3.

3. BOOT.SV now displays:

```
FILENAME?
```

Type the name of your DOS system. If this is the default name (SYS), and the system is in DP0, simply press RETURN; if its name is not SYS, type its name and RETURN.

DOS now displays its name and asks date/time questions. Answer these; then DOS will display the CLI prompt "R".

Signing Off DOS

To sign off DOS, type:

```
RELEASE masterdirectoryname
```

or

```
RELEASE %MDIR%
```

Now, turn off power to all equipment.

End of Bootstrapping DOS

Exceptional Status

Certain serious system errors can halt RDOS/DOS entirely in a *crash,* or suspend processing and display the contents of the ACs on the console in Exceptional Status.

After either event, you should try to discover the cause (see the Exceptional Status appendix of your system reference manual to produce a core dump). In an exceptional status, the system displays the contents of the four ACs, then an error code; e.g.:

```
011760   000012   000022   002462   100013
```

100013 is the error code; in exceptional status, bit 0 is always 1. Number 13 describes the cause.

Exceptional Status Codes

Last 2 Digits	Meaning
01	Attempt to return to master device block for which there is no record in MAP.DR. *This requires full initialization.*
02	SYS.DR error on access to the master device directory. If (AC0) = 16, the entry count in a directory block exceeds 16_8. If (AC0) = 16, a free entry in the block was indicated but none was found. *This requires full initialization.*
03	An interrupt stack overflow, where the stack address is in the low-order bits of AC2. An overflowed interrupt stack indicates a continually interrupting device. The code of the last device to cause an overflow is in AC1.
04	Inconsistent system data such as an illegal device address or a partial system overwrite by user program.
05	Master device data error. *Run a disk reliability test.*
06	Master device timeout. *Run a disk reliability test.*
07	Illegal device address, which can be caused by a misreading of the disk. *Run a disk reliability test.*

Last 2 Digits	Meaning
10	An undefined interrupt was detected that can't be cleared via an NIOC. The device code is displayed in the right byte of AC2. Often caused by a faulty connection.
12	Insufficient contiguous blocks to build push space indices. *Perform full initialization.*
13	Attempted .RTN from level 0 in the background.
14	Inconsistent IPB data. *Perform an IPB reliability test.*
15	Map violation received while control was in user interrupt code. The AC0 data field output on the console will contain the PC, not the contents of AC0, when the error occurred.
16	ECLIPSE ERCC multibit memory error. AC0 bit 15 and AC1 bits 0-15 contain the address.
17	NOVA3 semiconductor memory parity error. See release notice.

If a *system* error condition causes exceptional status, bit 0 of the error word is reset to 0 and the rest of the word gives the system error code.

End of Exceptional Status

Text Editor (EDIT/MEDIT) Commands

Command	Meaning
A	Append a page to current buffer.
↑A	Interrupt command execution.
B	Move CP to start of buffer.
$\text{Cstr}_1\ \$\text{str}_2\ \$$	Find str_1 and change to str_2.
Cstr$$	Delete string str.
↑C	Emergency return to CLI.
nD	Delete n characters from CP.
E	Output buffer and input file.
F	Output FF (form feed).
nF	Output FF after n inches of leader.
GC$	Close input and output files.
GOoutfile$	Close the current file and open a new outfile.
GRinfile$	Open and read input file infile.
GWoutfile$	Open and write output file outfile (this cannot already exist).

Command	Meaning
H	Return to the CLI. (In MEDIT, master console only.)
Istr$	Insert str at CP.
↑Istr$	Insert str after tabulation.
I↑L	Insert form feed at CP.
nI	Insert the ASCII character equivalent of decimal n at CP.
'nI	Insert ASCII character equivalent of octal n at CP.
nJ	Jump pointer to beginning of line n.
nK	Delete n lines from CP.
L	Move CP to the beginning of the line.
nL	Move CP n lines from current position.
nM	Move CP n characters.
Nstr$	Search for string; if not found, output buffer and continue search through input file, outputting the file until string is found.
P	Output buffer ending with a FF.
nP	Output n lines from CP, ending with a FF.

Command	Meaning
PW	Output edit buffer without a FF.
nPW	Output n lines from CP without a FF.
Qstr$	Search for string; if not found, continue search through all input files but output nothing.
R	Output edit buffer and read next page.
nR	Do R n times.
Sstr$	Search edit buffer for str.
T	Type edit buffer.
nT	Type n lines of buffer.
UCfilenm$	Rename output file to filenm.
UDfilenm$	Delete filenm.
UE	Write rest of input file (infile) to output file (infile.SC); delete infile; rename output file as infile.; close output file.
UH	Close all I/O files.
UNinfile$	Create and open infile.
URfile$_1$ $file$_2$ $	Rename file$_1$ to file$_2$.
US	Write rest of input file (infile.) to output (infile.SC). Rename output file to infile; rename infile. to file.BU. Close I/O files.

Command	Meaning
UYinfile$	Open existing infile for input. Open infile.SC for output and yank a page.
UZinfile$	Remove CLI-assigned P attribute from infile.
U?	Type names of all I/O files.
nW	Set window mode of n lines.
W?	Display window mode (0 or n).
W	Set page mode.
↑X	Delete current command line.
Y	Yank an input page into buffer.
nY	Yank n lines into buffer.
Z	Move CP to the end of the buffer.
↑Z	Appears in search string; used to match any character.
. (period)	Print number of line containing CP.
: (colon)	Print number of lines in buffer.
=	Print number of characters in buffer.

MACRO Commands

Command	Meaning
XD	Delete current macro.
$\text{XMcom}_1\ \$\ldots\text{com}_n\ \$$	Define current macro as commands com_1 com_n.
nX	Execute current macro n times.
X?	Type contents of current macro.
n#	Set number register to n.
#+	Increment number register.
#-	Decrement number register.
#?	Type contents of number register.
#0	Output number register to text.
n#!...\$!...\$	If number register ≠ n skip command characters between the first ! and next \$!.

End of EDIT/MEDIT Commands

Extended Assembler (ASM) Error Codes

A	Addressing error.
B	Bad character.
C	Colon error.
D	Radix error.
E	Equivalence error.
F	Formatting error.
G	Global symbol error.
I	Parity error on input.
K	Conditional assembly error.
L	Location counter error.
M	Multiply-defined symbol error.
N	Number error.
O	Overflow field error.
P	Phase error.
Q	Questionable line error.
R	Relocation error.
S	Symbol table overflow error.
T	Symbol table pseudo-op error.

U Undefined error.

X Text error.

Z Expression contains illegal operand.

End of ASM Error Codes

Superedit (NSPEED, SPEED) Commands

↑A -- Abort

↑A	Abort current command.

A -- Append*

*[:]*A	Append a page to current buffer.

B -- Buffers

BAx	Activate buffer x.
BCx	Copy the current buffer to buffer x.
[-] nBCx	Copy n lines from CP to buffer x.
m,nBCx	Copy the m + 1 through nth characters to buffer x.
BGx *[file]* $	Execute a Gx (x = R, W, or C) local to the current buffer.
BKx	Delete buffer x.
BSx	Make buffer x the current buffer.

* Commands A, C, N, Q, R, S and Y can be preceded by an optional colon that tells the command to return +1 if successful and 0 otherwise.

BTx	Perform a BCx, deleting the moved characters from current buffer. (Same for -nBTxand m,nBTx.)
BUx *[file]*$	Execute a Ux (x = Y, E, or S) local to current buffer.
B?$	Type status of all active buffers.
B?x$	Type status of buffer x.
↑Bx	Insert contents of buffer x in command string in place of this command.

↑ C -- Return to CLI

↑C	Emergency return to CLI.

C -- Search a Given Area for str1; Replace with str2.

[:] Cstr1 $str2$	From CP through buffer.
[:] [-] nCstr1 $str2$	n lines from CP (a - sign means toward start of buffer).
[:] m,nCstr1 $str2$	m + 1 through nth character.
[:] 0Cstr1 $str2$	Current line to CP.
@ *[:]* -Cstr1 &str2& @ *[:] [-]* nCstr1 &str2& @ *[:]* m,nCstr1&str2& @ *[:]* 0Cstr1 &str2&	Perform a search and replace as above but delimit strings in the text with the specified character (& is any character not in str.)

D -- Delete (see also K)

[-] nD	Delete n characters from CP.

E -- Output Buffer

E	Output current buffer and rest of input file; clear current buffer.

G -- File I/O (see also U, BG and BU)

GC$	Close current output file.
GR$	Close current input file.
GRfile$	Perform GR and open file for input.
GWfile$	Create and open sequential output file.
:GWfile$	Create and open random output file.
↑Gfile$	Insert the contents of file in command string at this command.

H -- Return to CLI (see also ↑C)

[UE] H	Normal exit from editor.
[US] H	Exit and save original file with .BU extension.

I -- Insert a String or Character

Istr$	Insert string at CP.
↑Istr$	Insert string after tab.
nI	Insert ASCII decimal equivalent of n.
n\	Insert ASCII equivalent of n where n is decimal 0 through 9.
@I&str& @↑I&str&	Perform the I command as above, using delimiter &, where & is any character.

J -- Jump Character Printer (see also L and M)

J	Move CP to start of buffer.
nJ	Move CP to start of line n.

K -- Delete (see also D)

K (or 0K)	Delete characters from start of this line to CP.
[-] nK	Delete n lines of characters from CP.
m,nK	Delete m + 1 through nth characters.

L -- Move Character Pointer (see also J and M)

L	Move CP to start of current line.
[-] nL	Move CP to start of nth line from current line.

M -- Move Character Pointer (see also J and L)

[-] nM	Move CP across n characters.

N -- Search Buffers (see also C, Q, and S)

[:] Nstr$	Search remaining input file for str, outputting buffers until found.
@ *[:]* N&str&	Perform N as above but include delimiters represented by &.

P -- Output Indicated Text to Output File

*[:]*P	Edit buffer with form feed (FF)*.
[:] PW	Edit buffer without form feed.
[:] *[-]* nP	n lines from CP with FF.
[:] *[-]* nPW	n lines from CP without FF.

* In all P commands, a colon means clear the buffer after executing the command.

[:] m,nP	m + 1 through nth characters with FF.
[:] m,nPW	m + 1 through nth characters without FF.
[:] OP	Current line to CP with FF.
[:] OPW	Current line to CP without FF.

Q -- Search Buffers (see also C, N, and S)

[:] Qstr$	Search remaining input file for str; do not output buffers (as in N).
@ *[:]* Q&str&	Perform Q as above but include delimiters represented by &.

R -- Output Current Buffer and Yank Next Page

*[:]*R	Output edit buffer; yank next page.
[:] nR	Perform R command n times.

S -- Search Specified Area for str (see also C, N, and Q)

*[:]*Sstr$	From CP through buffer.
[:] [-] nSstr$	n lines from CP.
*[:]*m,nSstr$	m + 1 through nth characters.
*[:]*OSstr$	Current line to CP.

@ *[:]* S&str& @ *[:]* *[-]* nS&str& @ *[:]* m,nS&str& @ *[:]* OS&str&	Perform search as above but look for the delimited str (& represents any character not in str).

T -- Type Out Specified Line(s) or Characters

T	Type current line, showing CP.
[-] nT	Type n lines from CP.
m,nT	Type m + 1 through nth character.
0T #T [illegible]	Type current line up to CP. Insert : (colon) before T commands to print lines on line printer (e.g., :0T).

U -- File I/O (see also G, BU, and BG)

US$	Write rest of input file to output file and close all I/O files. If opened by a UY, create backup.
UE$	Write rest of input file to output file and close all I/O files. If opened by a UY, delete input file and rename output file to input filename.
UYfile$	Open file for input, open file.SC for output, and yank a page.
U?$	Type global and local I/O file status.

V -- Numeric Variables

Vv	Return value of variable v.
VDv	Decrement v and return.
VIv	Increment and return value of v.
nVSv	Set value of v to n and return n.
VL	Return as value the number of lines containing the CP.
VN	Return as value the number of lines in current buffer.

WC -- Upper/Lowercase Control

WC	Return case control mode value.
0WC$$	Deactivate case control.
[-] nWCx$$	Shift up or down *(-)* with x as the shift character.
[-] nWCxy$$	Shift up or down *(-)* with x as the shift character and y as shift-lock character.

WM -- Page/Window Modes

WM	Return the mode value.
nWM	Set window mode of n lines.
0WM	Set page mode.

X -- Execute a CLI Command

Xstr$	Execute string as a CLI command.

Y -- Yank a Page

*[:]*Y	Clear buffer; read a page from input file.

Z -- Buffer Characters (see also #)

Z	Return total number of characters in current buffer.

; -- Jump From This Command Loop on a Given Condition

;	If last search command failed.
:;	If last search command succeeded.
n;	If $n \leq 0$.
n:;	If $n > 0$.

n -- Execute This Command String if Condition True

n"Ecstr'	True if $n = 0$.	n is a numeric argument.
n"Gcstr'	True if $n > 0$.	
n"Lcstr'	True if $n < 0$.	
n"Ncstr'	True if $n = 0$.	

cstr is the part of the command string between the condition and a single quote mark. It executes on true. On false, execution starts in the string after the first single quote mark.

Control Character (↑) in Search Strings

↑Nx	Accept any character but x here.
↑P	Position CP here if a search is successful.
↑T	Accept any number of spaces or tabs here.
↑V	Accept any number of the character x here.
↑Z	Accept any single character here.
↑←x	Interpret x literally, not as a special character.

Command Strings and Command String Labels

n<comstr>	Execute comstr n times; skip if n ≤ 0.
←n	Put the previous comstr in buffer n. (First command after a prompt.)
!labstr!	Define labstr as a label in the command string.
Olabstr$	Transfer control to label labstr.

Miscellaneous Commands

n=	Type the value of n.
. (period)	Returns and represents the number of characters in the buffer from the start of the buffer to the CP.
#	Represents the number of charcters in the buffer as a double argument: 0, Z (i.e., can replace m,n in C, T commands, etc.).
?	Complement the trace mode flag.

End of (N)SPEED Commands

Macroassembler (MAC) Error Codes

A	Addressing error.
B	Bad character.
C	Macro error.
D	Radix error.
E	Equivalence error.
F	Format error.
G	Global symbol error.
I	Parity error on input.
K	Conditional or repetitive assembly error.
L	Location counter error.
M	Multiply-defined symbol error.
N	Number error.
O	Overflow field or stack error.
P	Phase error.
Q	Questionable line error.
R	Relocation error.
U	Undefined symbol error.
V	Variable label error.
X	Text error.

End of MAC Error Codes

Debugger (DEB) Commands

General Commands

adr!	Open location adr.
adr/	Open location adr and print its contents.
)	Close open location.
↓	Close open location and open next location.
↑	Close open location and open previous location.
↑A	Return to CLI (standard Debug only).
%name	Enable local symbols (/U switch required in assembly and load). Also see K.
$A	Print contents of all accumulators.
n$A	Display contents of accumulator n.
$B	Print locations and numbers of all breakpoints.
adr$B	Insert breakpoint at location adr.
$D	Delete all breakpoints.
n$D	Delete breakpoint n. (n = 0-7).
$K 1$K	Remove all symbols from input/output; then enable the global symbols.

$P	Resume execution from a breakpoint.
n$P	Set breakpoint count register to n.
n$Q	Open break proceed counter for breakpoint n(n = 0-7).
$R	Start execution at address stored in USTSA.
adr$R	Restart execution at adr.
$V	Put debugged program in file (F)BREAK.SV; return to CLI.

Format Commands

Type ESC before any format character to display *all* subsequent words in the specified format.

Char.	**Means "display last word in:"**
:	Symbol format.
;	Instruction format.
?	.SYSTM command format.
'	ASCII format.
*	Symbol format (0B0).
&	Byte pointer format.
←	Half-word format.
=	Octal format (default).

Search Commands (Use Mask and Word Registers)

$S	Search all memory.
adr$S	Search memory from location 0 through adr.
adr>$S	Search memory from location adr to the limit of memory.
adr_1 <adr_2 $S	Search memory from location adr_1 through adr_2.

Register Commands

$C	Open carry register.
$E	Open extended save address register (USTSV).
$F	Print contents of floating-point accumulators.
$H	Open output register (0=console, 1=$LPT).
$I	Open interrupt register (IDEB only).
$J	Open search increment register.
$L	Open starting location register (USTSA).
$M	Open mask register.
$N	Open number register.
$T	Open task control block register (USTCT) (standard Debug only).

$T	Open console input done register (IDEB only).
$W	Open word register.
$Y	Open symbol table pointer register (USTSS).

Other Symbol Commands

$K	Remove all local and global symbols from input and output.
sym$K	Remove symbol sym from output permanently.

End of Debugger Commands

FORTRAN IV Compiler Error Messages

Number	Meaning
0	Working space exhausted.
1	Multiply-defined parameter.
2	Mixed precision operands. *
3	Unknown statement type. *
4	Statement does not end with blank(s).*
5	Syntax error in DATA variable list.
6	Syntax error in DATA literal list.
7	Syntax error in statement function.
10	Missing integer in FORMAT. **
11	Error in parameter list of CALL.
12	Array identifier not followed by a comma or a left or right parenthesis.
13	Illegal element in expression.
14	Improper use of array name.
16	Missing operator.
17	Illegal sequence of adjacent operators.
20	Illegal element when parentheses or literal or variable expected.

* Not necessarily fatal.

** Scan continues on syntax error only.

Number	Meaning
21	Premature statement end for an IF.
22	Operator (e.g., .EQ.) missing trailing period.
23	Illegal continuation line.
24	Period (.) not followed by a letter or number.
25	Format error. **
26	Format error after repeat count. **
27	Abnormal end to FORMAT statement.
30	Expression not closed at statement end.
31	Multiply-defined error.
32	Variably dimensioned array not a dummy.
33	DATA variable list longer than literal list.
34	Same identifier typed > once.**
35	Unclosed DO loop in program.
36	Common variable previously declared EXTERNAL, subprogram, or dummy.
37	Dummy identifier predefined. **
40	Dimension error. **

** Scan continues on syntax error only.

Number	Meaning
41	Improper statement terminating DO loop.
42	Variable dimension (main program array). **
43	Array size greater than 32K.
44	Parentheses not closed by statement end.
45	Compiler expected numeric operand (unary -1).
46	Expected logical operand (.NOT.).
47	Illegal operand type for operator. *
50	Types do not match in DATA statement.**
51	Equivalenced pair both in COMMON.
52	Start of common extended by equivalence.
53	Irrecoverable format error.
54	Statement function name conflicts with previous declaration.
56	Not enough subscripts (DATA or EQUIVALENCE). **
60	Formal statement syntax in error. **
61	Undefined label.

* Not necessarily fatal.

** Scan continues on syntax error only.

Number	Meaning
62	Attempted load/store of external or array.
63	Array element specified for dummy array.
64	External identifier previously declared. **
65	Variable dimension is not a dummy.**
66	DATA list variable not in labeled common.
67	Equivalenced pair, neither in common.
70	Subscript not of type integer.
71	Wrong number of arguments (reserved name function).
72	Wrong type arguments (reserved name function).
73	Non-digit in label field. *
74	Carriage return in label field. *
75	Improper statement in BLOCK DATA.
76	Unreferenced label. *
77	Stack variable referenced in statement function.
100	No room for all runtime variables. **

* Not necessarily fatal.

** Scan continues on syntax error only.

Number	Meaning
101	Undeclared identifier (statement function expression).
102	RETURN statement in main program.
104	$ not followed by digit.
105	End of file without END.
106	Wrong number of subscripts.
111	Hollerith constant not ended at statement end.
112	Integer > (2**15)-1; truncated. **
114	Exponent error in real. **
115	Exponent error in double precision. **
116	Illegal character. **
120	Literal error: 1) 2 operands not both literals. 2) 2 literals of different types. 3) source line is: (literal, literal operator and operator is not a right parenthesis.
121	Attempted equivalence to dummy argument.

* Not necessarily fatal.

** Scan continues on syntax error only.

Number	Meaning
122	Error in CHANTASK statement.
123	Identifier used as array element in DATA statement but not declared as an array.
124	Illegal literal value for increment in DO.
125	Illegal complex relational tests.
126	Missing comma in I/O list.
127	Extraneous comma.
130	Illegal variable name.
140-160	These are compiler errors for debugging only.

End of FORTRAN IV Compiler Error Messages

FORTRAN IV Runtime Error Messages

Number	Meaning
1	Stack overflow.
2	Computed GOTO error.
4	Division by zero.
5	Integer overflow.
6	Integer power error.
7	Floating point underflow.
8	Floating point overflow.
9	Illegal format syntax.
11	Logic conversion error.
13	Number conversion error.
14	I/O error.
15	Field error (e.g., F5.10, E5.4).
16	Square root of negative number.
17	Logarithm of negative number.
18	Channel not open.
19	Channel already open.
20	No channels available.
21	System exceptional status.

Number	Meaning
24	Exponential over/underflow.
25	Array element out of bounds.
26	Negative base for floating-point power.
27	Number stack overflow.
28	BACKSPACE not implemented.
29	Attempt to restore status to unsaved channel.
30	Queued task error.
31	Seek on nonrandom file.
32	Overlay aborted.
33	Illegal argument.
34	Delete error (file open).
35	Overlay error in overlay kill.
36	Undefined entry.

FORTRAN IV System Error Messages

There are three FORTRAN IV system error messages that are FORTRAN generated. They are:

Code	Meaning
0	Indeterminate error.
1	Call successfully completed.
2	Activity temporarily in progress.

Other FORTRAN IV system errors are mapped to RDOS error codes returned through AC2 to IER (see system errors, preceding) according to the formula:

FORTRAN IV error = RDOS error code + 3_{10}

End of FORTRAN IV Error Messages

Batch Commands (RDOS)

Most BATCH commands are subsets of RDOS commands and have similar formats (except for the initial ! and use of # for continuation lines). Other differences between BATCH and RDOS formats arise from differences in system handling, e.g., BATCH has no switch specifying line printer output on commands such as LOAD and LIST. The BATCH commands are listed below; those marked with an asterisk (*) are not within the RDOS subset and are described on the pages following.

!ALGOL

!APPEND

!ASM

!BASIC

!BPUNCH

!CCONT

!CHATR

!CHLAT

!COMMENT*

!CRAND

!CREATE

!CTA*

!DELETE

!DIR

!DISK

!DKP*

!DUMP

!EOF*

!EXEC*

!FILCOM

!FORT

!FORTRAN

!FPRINT

!GDIR

!GMEM

!GSYS

!GTOD

!JOB*

!LFE

!LINK

!LIST

!LOAD

!MAC

!MDIR

!MKABS

!MKSAVE

!MOVE

!MTA*

!OVLDR

!PAUSE

!POP

!PRINT

!PUNCH

!RDOSSORT

!RELEASE

!RENAME

!REPLACE

!REV

!RLDR

!SAVE

!TPRINT

!TUOFF

!TUON

!UNLINK

!XFER

!COMMENT

Format

```
!COMMENT message-string
```

Output an operator message during job execution. message-string may be up to 72 characters long.

Example

```
!COMMENT COMPOUND INTEREST ROUTINE
```

!CTA

Format

!CTA cassette name *[cassette-label]*

Assign an available cassette unit to a logical cassette referenced in your program. You may specify an optional cassette label.

Example

```
!CTA MYFILE 12345)
```

MYFILE is the logical name of the cassette having label 12345. When the command is encountered, the operator message

```
MOUNT CASSETTE 12345
```

is output and the operator mounts an appropriate cassette (e.g., CT1) and responds

```
CT1)
```

associating CT1 with MYFILE.

!DKP

Format

!DKP logical-device-name *[cartridge-label]*

Assign a moving-head disk drive to a logical disk unit referenced in the program. You may specify an optional *cartridge label.*

Example

```
!DKP MYPAK 11223)
```

MYPAK is the logical disk unit and 11223 is the label of the pack. When the command is encountered, the operator message

```
MOUNT DISK PACK 11223
```

is output and the operator selects a free disk drive, e.g., DP1, mounts it and responds

```
DP1)
```

associating DP1 with MYPAK.

!EOF

Format

```
!EOF
```

Declare the end of a job stream. The command is optional; you can use an end-of-file mark in its place.

Example

```
!EOF
```

The end-of-file is the last command in the job stream.

!EXEC

Format

!EXEC *[$prognam_1$ $prog_1$ args...$prognam_n$ $prog_n$ args]*

Execute a save file, *prognam*. If you omit program names, TMP.SV is executed. The optional program arguments, *progargs,* are passed to the CLI communications file COM.CM or FCOM.CM created for the program.

The command:

```
!EXEC program-name
```

is identical to the command:

```
!program-name
```

Example

```
!JOB SMITH)
!FORT SMITH)
!RLDR TMP FORT.LB SMITH/S)
!EXEC SMITH)
!EOF)
```

After compilation and loading of job SMITH, the command !EXEC SMITH (or !SMITH) executes the loaded program.

!JOB

Format

```
!JOB label
```

Identify the beginning of a job within the job stream and associate a label with the job. The label may be up to 132 characters long, including alphanumerics and $.

Example

```
!JOB SMITH)
  .
  .
  .
!EXEC SMITH)
!JOB JONES)
  .
  .
  .
!EXEC JONES)
!EOF)
```

The job stream consists of two jobs labeled SMITH and JONES.

!MTA

Format

```
!MTA logical-device-name [tape-label]
```

Assign an available magnetic tape unit to a logical magnetic tape unit referenced in the program. You may specify an optional tape label.

Example

```
!MTA MYTAPE 10000)
```

MYTAPE is the logical tape unit and the label of the tape reel is 10000. When the command is executed, the operator message:

```
MOUNT TAPE 10000
```

is output. The operator selects a free transport (e.g., MT3), mounts the tape and responds:

```
MT3)
```

associating MT3 with MYTAPE.

!PAUSE

Format

```
!PAUSE message
```

Output a messageto the operator during execution of a job and suspend system activity until the operator responds with

```
GO)
```

via the system console.

Example

```
!PAUSE INSERT INVENTORY FORMS IN $LPT)
```

This message will be output on the system console at the point where the batch user placed it in his job file.

End of BATCH Commands

Extended BASIC Error Messages

00 FORMAT

Unrecognizable.

01 CHARACTER

Illegal.

02 SYNTAX

Invalid argument.

03 READ/DATA TYPES

Inconsistent.

04 SYSTEM

Hardware or software.

05 LINE NUMBER

Not 1 through 9999.

06 EXCESSIVE VARIABLES

Greater than 286.

07 COMMAND

Illegal.

08 SINGULAR MATRIX

One-dimensional array with MAT.

10 ATTRIBUTE

Invalid CHATR arg.

11 PARENTHESES

Not paired.

14 PGM OVFL

Not enough storage to ENTER.

15 END OF DATA

Not enough for READ.

16 ARITHMETIC

Value too large or small.

18 GOSUB NESTING

Greater than that specified at SYSGEN.

19 RETURN-NO GOSUB

20 FOR NESTING

Greater than that specified at SYSGEN.

21 FOR-NO NEXT

22 NEXT-NO FOR

23 DATA OVFL

Not enough storage for variables.

24 DIRECTORY EMPTY

No files within it.

25 OPTION

Feature not SYSGENed.

27 FILE NUMBER

Invalid in I/O statement.

28 DIM OVFL

Exceeds dimensions.

29 EXPRESSION

Too complex to evaluate.

30 ILLEGAL FILE MODE

Invalid in I/O statement.

31 SUBSCRIPT

Exceeds array DIMensions.

32 UNDEFINED FUNCTION

33 FUNCTION NESTING

Too deep.

34 FUNCTION ARGUMENT

Range exceeded.

35 ILLEGAL MASK

Illegal PRINT USING.

37 USER ROUTINE

Not found in storage.

39 DUP MATRIX

Same on both sides of MAT multiply/transpose.

40 MATRICES SIZES

Differ.

41 UNDIMENSIONED VARIABLE

Attempt to use undimensioned MATRIX.

43 MATRIX NOT SQUARE

On inversion.

45 DATA>LRECL

Logical record exceeded.

46 INPUT

Too many responses to INPUT

48 NOT A CORE IMAGE

On CHAIN, RUN, or LOAD.

49 NO ROOM FOR DIRECTORY

(System file directory).

51 USER NOT ACTIVE

When message sent.

52 USER IN NOMSG STATE

When message sent.

54 STATEMENT LENGTH

Greater than 132 characters.

55 EXECUTE-ONLY

Attempt to read an execute-only file.

56 RANGE

Record greater than 262,144.

58 INCOMPATIBLE CORE IMAGE

Attempt to LOAD file saved under different floating-point precisions.

59 ZERO STEP

On FOR-NEXT.

60 TIME-OUT

Timed input decremented to 0.

61 INVALID DECIMAL STRING

Nonnumeric in string arithmetic.

62 STAR OVERFLOW

Computes to more than 18 digits.

64 SYSTEM ACTIVE

Attempt to BYE system with user(s) logged on.

End of Extended BASIC Error Messages

Extended BASIC File I/O Error Messages

The error messages that follow originate in RDOS or DOS:

01 ILLEGAL FILE NAME

02 ILLEGAL SYSTEM COMMAND

Not defined in RDOS/DOS.

03 ILLEGAL COMMAND FOR DEVICE

04 NOT A CORE IMAGE FILE

Not in SAVE format.

06 END OF FILE

07 READ-PROTECTED FILE

08 WRITE-PROTECTED FILE

09 FILE ALREADY EXISTS

10 FILE NOT FOUND

11 PERMANENT FILE

12 ATTRIBUTE PROTECTED

13 FILE NOT OPENED

14 SWAPPING DISK DATA CHECK

Error in swap file.

15 REVISION CHECK

Not revision 5 of BASIC.

16 CHECKSUM

Disk read error.

17 CHANNEL NOT AVAILABLE

System channel limit reached.

18 LINE LIMIT

Too many characters.

20 PARITY

On read line.

23 NO FILE SPACE

Left in directory.

24 READ ERROR

25 SELECT STATUS

Unit not ready, or write-protected.

29 DIFFERENT DIRECTORIES

30 ILLEGAL DEVICE CODE

Or device not in system.

31 ILLEGAL OVERLAY

System error.

37 DEVICE ALREADY INITIALIZED

38 INSUFFICIENT CONTIGUOUS BLOCKS

Reorganize directory or disk.

41 NO MORE DCBS

System limit for devices/directories reached.

42 ILLEGAL DIRECTORY SPECIFIER

43 UNKNOWN DIRECTORY SPECIFIER

44 DIRECTORY TOO SMALL

(Operator) Minimum is 48 blocks.

45 DIRECTORY DEPTH

46 DIRECTORY IN USE

End of BASIC File I/O Error Messages

Extended BASIC System Error Messages

44 DIRECTORY TOO SMALL

(Operator) Minimum is 48 blocks.

45 DIRECTORY DEPTH

(Operator) Attempt to create directory within an equivalent directory.

46 DIRECTORY IN USE

By program in the other ground.

47 LINK DEPTH

Exceeded. Maximum chain is 10 links.

48 FILE IN USE

Contact Operator if file is in your directory.

52 FILE POSITION

Read too far in modes 4 or 5.

54 DIRECTORY NOT INITIALIZED

58 DIRECTORY SHARED

File space exhausted.

59 DISK IS FULL

File space exhausted.

End of Extended BASIC System Error Messages

Notes

Notes

Data General

Data General Corporation, Westboro, Massachusetts 01580