NOVA® -LINE

DISKEDIT

User's Manual

093-000186-00

Ordering No. 093-000186

Printed in the United States of America
Rev. 00, November 1975

NOTICE

Data General Corporation (DGC) has prepared this manual for use by DGC personnel, licensees and customers. The information contained herein is the property of DGC and shall neither be reproduced in whole or in part without DGC prior written approval.

DGC reserves the right to make changes without notice in the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including but not limited to typographical, arithmetic, or listing errors.

(RDOS 05) Original Release - November 1975

The following are trademarks of Data General Corporation, Southboro, Massachusetts:

U.S. Registered Trademarks			Trademarks
CONTOUR I	NOVA	NOVALITE	INFOS
DATA PREP	NOVADISC	SUPERNOVA	
ECLIPSE			

CONTENTS

DISKEDIT

FEATURES

The DISKEDIT program provides powerful tools for editing a disk. DISKEDIT enables the user to examine, modify and rearrange disk words; it can search for a specific word, fill different parts of disk with a word, and display any part of disk. The program offers seven different output formats, and a choice of number bases ranging from 2 to 16.

DISKEDIT is designed for the following disk types:

- 4047A & 4047B Disk Cartridge Subsystem
- 4048A Disk Pack Subsystem
- 4057A Disk Pack Subsystem
- 4231A Disk Pack Subsystem
- 4234A Series Disk Cartridge Subsystem
- 4237/4238 Disk Cartridge Subsystem
- Models 6001 through 6008 Fixed-Head Disk System
- 6030 Series Diskette Subsystem

LOADING AND STARTING DISKEDIT

DISKEDIT is supplied as DISKEDIT.SV, and runs as a stand alone program. The program is loaded by the Binary Loader, TBOOT, MCABOOT, CBOOT, or BOOT depending on its medium and the equipment which will run it. After loading, DISKEDIT will request the disk type and the physical disk unit; valid responses will put the program in command mode.

DISKEDIT's first messages are:

```
"DISKEDIT"

"DISK DRIVE MODEL NUMBER?"
```

The valid response is the disk drive model number and a carriage return ()). Drive model numbers include 4047, 4048, 4057, 4231, 4234, 4237/4238, 6001-6008, or 6030. Any other response evokes the following message:

```
"UNKNOWN DRIVE TYPE"
```

After a valid response, DISKEDIT asks:

```
"DISK UNIT?"
```

The response must be the RDOS-supported device mnemonic for the disk (for example, DK0, DP7, DP5F). An incorrect entry evokes the error message:

```
"ILLEGAL DISK UNIT DECLARATION"
```

The following message is output for fixed-head disks only:

```
"DISK CONTAINS _______ (____) BLOCKS"
```

This is a convenience message; DISKEDIT will output the number of disk blocks, in decimal, and within parentheses, in octal.

DISKEDIT indicates that it is ready for editing commands by outputting a period (.).

FORMAT

Commands consist of expressions, followed by a valid command code. There are two command types: LOCAL and ESCAPE. LOCAL commands concern specific locations, while ESCAPE commands perform general disk functions. LOCAL command format does not use ($) otherwise, both formats are:

```
[disk address:displacement][$]command code
```

Where: disk address is the block number being accessed. If the disk address is missing, DISKEDIT assumes the last block number accessed.

displacement is the displacement in the block at disk address. If the displacement is missing the last -used displacement is assumed.
$ (produced by typing ESCape) preceeds ESCAPE command codes. ESCAPE commands perform general disk functions, like dumps and searches. Local commands do not use ESCape.
command code is a single character that represents a command. The command code also terminates a command.

The two command types, LOCAL and ESCAPE, and their commands are discussed after the next topic.

CONTROL-A aborts commands.

To distinguish DISKEDIT output from user input in this text, all user input is underlined.

KEYBOARD CALCULATIONS

The value of any field in a command may be calculated with: addition (+), subtraction (-), and exclusive ORing (!). Results are in octal. These calculations may be specified in both the disk address and displacement portions of DISKEDIT's command format, in both ESCape ($) and Local commands.

Parenthesis may be used within individual calculations and are evaluated from left to right.

Within a calculation, a period can have two functions. When used as a character, it implies the current location value; after a number, it indicates decimal notation for that number.

For example 14+.= will result in 21 if the current location is 5. A period followed by an equals sign (.=) outputs the current location value. 14.+.= will result in 23; the first period indicates a decimal number.

The following command:

(10.+2)+(17-7):4+5$C calculates a disk address of 24 and a displacement of 11.

The Local command: (2+5)-(4+2)= calculates the value of 1.

Calculation arguments may have the following forms:

double precision octal or decimal numbers (decimal numbers must be followed by a period [.]).

single precision number/single precision number.

"charactercharacter"

number;filename

head, sector, cylinder

Where: double precision - represents any number
single precision number/single precision number represents a double precision disk address as two words.
"charactercharacter" specifies one or two characters packed left to right in a word, with null fill. For instance:

```
""=0
"character"=character * 400
"charactercharacter"=char-
     acter * 400 + character
```

number;filename - returns the hash number of a filename. The number is the frame size; filename is an RDOS file name.
head, sector, cylinder - these figures indicate an address on moving head disks only. Commas separate them.

LOCAL COMMANDS

Local commands are entered in the Command code portion of the format. Local commands must not be preceeded by an ESCape key ($). A summary of local commands follows:

Command	Description and Meaning
/	Right slash. Open current word and display contents in current format.
'	Apostrophe. Display the last word or expression in ASCII format.
←	Left arrow. Display the last expression in half-word format as two octal numbers.
=	Equals sign. Display the last expression in word format in current number base.
*	Asterisk. Display the last expression as octal word.
⤸	Carriage return. Store result of expression in currently opened word. If no word is currently opened, this is a no op.

Command	Description and Meaning
LINE FEED (↓)	Store result of expression in currently opened word and display the next word.
↑	Up arrow. Store result of expression in currently opened word and display the previous location.
$	ESCape - enters ESCape command mode; the character following $ is a command.

ESCAPE COMMANDS

ESCAPE commands affect general functions on the disk. These commands must be preceeded by an ESCape key ($) in command lines; each is a terminating character. The following summary describes each ESCAPE command.

Command	Meaning
M	Open the Mask register for display and modification.
W	Open the Word register for display and modification.
N	Open the Number register for display and modification. This register defines the output format of numbers. It is coded as follows: 100000 = treat the number as signed 40000 = do preceeding zero suppress base = define the base of the number when output. This can range from base 2 to base 16.
H	Open the output register for display and modification. The output register directs the output of search and dump commands to the teletypewriter if 0, or the line printer if non-zero.
J	Open the increment register for display and modification.
S	Search the disk for a specified value.
D	Dump specified disk locations.
F	Fill specified disk locations with a word from the word register.
R	Forces a block to be Read into core. Any block already in core is not written out even though it may have been modified.
O	Force block to be Output to disk.
A	Print block number currently in core.
B	This command changes the name (address) of the Block in core. The $O command can then write the block to the address specified in $B.
C	Calculate expression and print result. If used alone, this command will print current location.
U	This command restarts DISKEDIT at the "DISKEDIT DISK DRIVE MODEL NUMBER?" query, and resets registers to startup values.
Z	Writes any modified block in core and halts program. The CONTINUE switch at the computer console will restart DISKEDIT at the "DISKEDIT DISK DRIVE MODEL NUMBER" query.

The following ESCAPE commands are elaborated under Output Format Commands, at the end of this manual.

Command	Meaning
=	Output numbers in word format.
'	Output numbers in ASCII format.
	Output numbers in half-word octal format.
*	Output numbers in word format. The asterisk sets number register to 8 for octal output.
:	Output the disk block number(s) in double precision octal format.
\	Output the disk block number(s) as two octal words.
,	Output the disk block number(s) in head, sector, cylinder format (Fixed-head disks only).

ESCAPE COMMAND DESCRIPTIONS

Word ($W) and Mask ($M) Registers

DISKEDIT uses both the Mask and Word registers to search the disk.

For searches, the user sets the Word register to represent the value sought on the disk. The mask register is set to mask specific bits of words (or to mask no bits). As the search proceeds, the value of each location is first ANDed with the Mask register, then compared with the contents of the Word register. If they match, the location and its value are printed out. When DISKEDIT starts, the Word and Mask registers each contain 0, which will print all disk locations in a search command.

The Word register is opened by the command:

```
$W
```

The Mask register is opened by the command:

```
$M
```

An example of each command is:

```
.$W 000000 1500)  Open Word register and
                  store 1500
.$M 000000 17700) Open Mask register and
                  mask bits 0-9.
```

Number Register ($N)

The Number register determines in which base numbers will be printed, when displayed in the word format. This register also defines the output format of numbers and is coded as follows:

100000	treat the number as signed
40000	suppress leading zeroes
base	base is an integer from 2 to 16, which defines the base of the number on output.

By default the Number register is set to 0, which prints out the register contents in octal. If the user sets the contents of the Number register to a base number, that number becomes the output base number. Only base 2 through base 16 are valid. Output formats are discussed at the end of this manual.

The Number register is opened for examination and modification by the command:

```
.$N
```

Some examples of the command are:

```
.$N 000010 10.)    - Open Number register and
                     set number base to 10 (note
                     period: 10.)
.$N 00010 16.)     - Open Number register and
                     set number base to 16.
.$N 0010 10 )      - Open Number register and
                     set number base to 8. (No
                     period follows base
                     specification).
.$N 000010         - Open Number register, sup-
     40000+10 )      press leading zeros, and
                     set base at 8.
.$N 40010)         - Open and examine the Num-
                     ber register.
```

Increment Register ($J)

DISKEDIT uses the Increment register to search the disk. The user specifies a value in this register which adjusts the increment between search locations. The value in this register can also be used in Fill and Dump commands. If the value is zero or negative, then 1 becomes the increment.

The Increment register is opened for examination and possible modification by the command:

```
$J
```

An example of the command is:

```
.$J 000000 12)   - Open Increment
                   register and store 12.
```

Search Command ($S)

This command searches all or part of disk for a specified value, and prints locations on disk which contain the value.

The following steps will perform a search:

- set the Mask register to the mask desired for the search.

  set the Word register to the value sought.

- set the Increment register to the desired increment value.
- specify the Search command. The Search Command ($S) will search the entire disk unless the search command line specifies otherwise (see next page).

The Search command combines the following information to find a match:

Mask-Word AND Disk Word = Word for match condition

Where: Mask is the contents of the Mask Register and Word is the contents of the Word Register. disk word is a word from the disk. AND represents logical AND.

When DISKEDIT starts, both Word register and Mask register contain 0, and the search increment is 1.

On matching conditions the location is printed. After a match, DISKEDIT adds the value in the Increment register to the match location, and continues searching. Comparing and incrementing continues until all the specified disk area is searched.

The Search Command is:

```
$S
```

The Search command can restrict a search to specific portions of the disk.

The command

```
disk address:displacement$S
```

searches the disk from the beginning to the specified address and displacement.

The command

```
disk address:displacement<$S
```

searches from the specified address and displacement to the end of the disk.

The command

```
disk address:displacement<disk address:
     displacement$S
```

searches between the two specified addresses and displacements.

An example of the Search command is:

To find all words containing 1 on the disk:

```
.$W 000000 1 )  ← Set Word register for
                  search: value 1.
.$M 000000 -1 ) ← Setting Mask register to
                  -1 causes default: register
                  will compare all bits of
                  words.
.$J 000000 )    ← Set Search Increment
                  register-default increment
                  value is 1
.$S             ← Search the disk
```

Results in:

```
000000:102\000001
001020:052\000001
```

This example indicates that only the two listed words contained a 1.

Dump Command ($D)

This command dumps all or part of disk to an output device. Unless the user specifies otherwise in the command line, the entire disk is dumped. The incremental value in the Increment register can dump selected incremental locations. All dumped outputs include the disk address and all the data within the location. Samples of dumping include:

The command:

```
$D
```

dumps the whole disk.

The command:

```
disk address:displacement$D
```

dumps from the beginning of the disk to the specified address

The command:

```
disk address:displacement<$D
```

dumps all locations from the specified address and displacement to the end of disk.

The command:

```
disk address:displacement<disk address:
     displacement$D
```

dumps all locations between the two specified addresses and displacements.

An example of this command is:

```
.1045:0<20$D    ← Dump words 0-20 of block
                  1045.
```

The results are:

```
    001045:000/005126 000023 000000 004010
         001342 000000 000000 003524
    001045:010/000000 003013 002041 000000
         000000 006551 000000 003311
    001045:020/005322
```

Fill Command ($F)

The Fill command writes a word from the word register into specific location(s). The user selects the Fill word and places it in the word register, and the $F command writes the word to specified locations. The incremental value in the Increment register can Fill incremental locations with this word. After specifying the Fill command the user is asked: "REALLY DO IT?"

Samples of the Fill Command are:

The Fill command alone:

```
$F
```

fills the entire disk with the value in the Word register.

The command:

```
disk address:displacement$F
```

fills the disk from the beginning to the specified address with the value in the Word register.

The command:

```
disk address:displacement<$F
```

fills all locations from the specified address to the end of disk with the value in the Word register.

The command:

```
disk address:displacement<disk
      address:displacement$F
```

fills all locations from the first address and displacement through the ending disk address and displacement with the value in the Word register.

Once the Fill command is specified, DISKEDIT asks:

```
"REALLY DO IT?"
```

Valid responses are Y (Yes) or N (No). Y writes the specified word into the specified location(s); N aborts the command:

An example of the Fill command is:

```
.$W 000000 -1)     ← set Word register.
.$J 000000)        ← open Increment register for
                     display
.$F                ← no restriction; fill the entire
                     disk.
REALLY DO IT? Y
```

The Fill command then writes a -1 to all blocks of the disk.

Force Block to be Read Into Core Command ($R)

This command forces a user-specified disk block to be read into core. This command is useful when the original (unedited) version of the previously requested disk block is required. The edited copy of the block is eliminated when the original version is read into core.

Changes to a block are not written onto the disk until a different block is accessed, or until an $O or $Z command is entered.

After the original version of the block is in core, editing can continue.

The format of the $R command is:

```
disk address:$R
```

An example of the $R command is:

To force current block to be read in again type:

```
.$R
.
```

Force Block to be Output to Disk ($O)

This command forces the block currently in core to be output to disk. $O is often used to clear core for a new block or to change a block's location on the disk. After specifying the $O command, the user is asked "REALLY DO IT?"

The command to force the current block in core back to where it came from is:

```
$O
```

The command to force the current block in core to a new location is:

```
disk address:$O
```

DISKEDIT will then output this message:

```
REALLY DO IT?
```

Valid responses are Y (Yes) or N (No). A Y response outputs the block; an N response aborts the command.

Print the Block Number Currently in Core ($A)

This command prints the block number currently in core, in the selected block number format.

The format of this command is:

```
$A
```

For example:

```
.5:01/000000)      ← Type block into core
.$A                ← Print number of block in
                     core
000005
.102:53/126412)    ← access another block
.$A                ← print its number
000102
```

Set the Current Core Block to a Different Address($B)

Ordinarily, the block in core is sent back to its original disk address. The $B command specifies a different address for the core block; the $O command will then write the block to the address specified in $B.

The format of the $B command is:

```
disk address:$B
```

For example:

```
.5:01/002300)      ← Type block into core
.$A                ← Print block
000005
.10:$B             ← Set block to different
                     destination
.$A                ← Print new destination
000010
.01/002300LINE FEED
000010:0021/123456
```

Print Out Results of Expression Evaluation ($C)

This command calculates an expression and prints the results.

$C command formats include:

```
$C
```

which prints the current address and displacement.

The command

```
disk address:displacement$C
```

Calculates the specified expression and prints the results.

Three examples are:

```
.10:53/100245      ← displays contents of disk
                     location
.$C                ← evaluate expression using all
                     defaults
000010:053
.5+5:50+30$C       ← evaluate expression and
                     display result
000012:100
.$C                ← display current address and
                     displacement.
000010:053
```

Restart the DISKEDIT Program ($U)

This command restarts the program at the "DISKEDIT DISK DRIVE MODEL NUMBER?" query, and resets all registers to their initial values.

The command format is $U.

Write Modified Core Block and Halt ($Z)

This command writes any current modified core block back to disk and halts the DISKEDIT program. The CONTINUE console switch restarts the program at the "DISK TYPE" question.

The format of this command is:

```
$Z
```

OUTPUT FORMAT COMMANDS

These commands enable the user to specify and change the form of his output. A specified format will apply to all future printouts unless it is explicitly changed.

The format of these commands is:

```
$x
```

Where: x represents one of the following characters:

= - word format
' - ASCII format
← - half-word octal format
* - word format (set number register to 10 for octal retrieval)
: - output disk block numbers in double precision output format
\ - output as two octal words
, - output in head, sector, cylinder format (Fixed-head disks only)

Examples of the commands include:

```
.$*                  ← set to output in octal word
                       format
.15+2+3=000022)      ← display results in octal
.$N 000010 10.)      ← set to output numbers in
                       decimal
.15+2+3=00018)       ← display expression in
                       decimal
.$N 00010 40000+     ← output numbers in decimal,
  10.)                 suppress leading zeroes
.15+2+3=18)          ← try calculation again
.$*                  ← set to output in octal word
                       format
.0:0/040501_'AA)     ← display a disk word, examine
                       it in ASCII
.$'                  ← set to display words in
                       ASCII
.0/AA=040501←        ← display disk word in ASCII,
  101 101)             then in half-word format
.$←                  ← set to display word in half-
                       word format
.0/ 101 101)         ← do it
.$=                  ← set to display in word
                       format
.0/040501)           ← try it
.$:                  ← set to output in double
                       precision
.15:45+2$C           ← calculate disk octal address
                       and display result
000015:047
.$,                  ← set to output disk address in
                       head, sector cylinder
                       format
.5:45$C              ← display disk address
0,5,0:045
.$\                  ← set to output in double word
                       format
.5432:53$C           ← disk address, calculate
000000\005432:       ← try it
    053
```

END OF MANUAL

DataGeneral

Software Documentation Remarks Form

How Do You Like This Manual?

Title ____ No. ______________

We wrote the book for you, and naturally we had to make certain assumptions about who you are and how you would use it. Your comments will help us correct our assumptions and improve our manuals. Please take a few minutes to respond.

If you have any comments on the software itself, please contact your Data General representative. If you wish to order manuals, consult the Publications Catalog (012-330).

Who Are You?

□ EDP Manager
□ Senior System Analyst
□ Analyst/Programmer
□ Operator
□ Other ______________________

What programming language(s) do you use? ____________
____

How Do You Use This Manual?

(List in order: 1 = Primary use)

_____ Introduction to the product
_____ Reference
_____ Tutorial Text
_____ Operating Guide
_____ ______________________

Do You Like The Manual?

Yes	Somewhat	No	
□	□	□	Is the manual easy to read?
□	□	□	Is it easy to understand?
□	□	□	Is the topic order easy to follow?
□	□	□	Is the technical information accurate?
□	□	□	Can you easily find what you want?
□	□	□	Do the illustrations help you?
□	□	□	Does the manual tell you everything you need to know?

Comments?

(Please note page number and paragraph where applicable.)

From:

Name ____________________ Title ____________________ Company ____________________
Address ______________________________________ Date ____________________

FOLD DOWN FIRST FOLD DOWN

FIRST
CLASS
PERMIT
No. 26
Southboro
Mass. 01772

BUSINESS REPLY MAIL

No Postage Necessary if Mailed in the United States

Postage will be paid by:

Data General Corporation

Southboro, Massachusetts 01772

ATTENTION: Software Documentation

FOLD UP SECOND FOLD UP

STAPLE