DEC-11-UVERA-A-D

DOS/BATCH
Verification Program (VERIFY)
Programmer's Manual

FOR THE DOS/BATCH OPERATING SYSTEM

Monitor Version VØ9

August 1973

For additional copies, order No. DEC-11-UVERA-A-D from Digital Equipment Corporation, Software Distribution Center, Maynard, Massachusetts Ø1754.

Your attention is invited to the last two pages of this document. The "How to Obtain Software Information" page tells you how to keep up-to-date with DEC's software. The "Reader's Comments" page, when filled in and mailed, is beneficial to both you and DEC; all comments received are acknowledged and considered when documenting subsequent manual.

Associated documents:

DOS/BATCH Monitor
Programmer's Manual, DEC-11-OMPMA-A-D

DOS/BATCH User's Guide, DEC-11-OBUGA-A-D

DOS/BATCH Assembler (MACRO-11)
Programmer's Manual, DEC-11-LASMA-A-D

DOS/BATCH FORTRAN Compiler and Object Time System
Programmer's Manual, DEC-11-LFRTA-A-D

DOS/BATCH System Manager's Guide, DEC-11-OSMGA-A-D

DOS/BATCH File Utility Package (PIP)
Programmer's Manual, DEC-11-UPPAA-A-D

DOS/BATCH Debugging Program (ODT-11R)
Programmer's Manual, DEC-11-UDEBA-A-D

DOS/BATCH Linker (LINK)
Programmer's Manual, DEC-11-ULKAA-A-D

DOS/BATCH Librarian (LIBR)
Programmer's Manual, DEC-11-ULBAA-A-D

DOS/BATCH Text Editor (EDIT-11)
Programmer's Manual, DEC-11-UEDAA-A-D

DOS/BATCH File Compre Program (FILCOM)
Programmer's Manual, DEC-11-UFCAA-A-D

DOS/BATCH File Dump Program (FILDMP)
Programmer's Manual, DEC-11-UFLDA-A-D

DOS/BATCH Disk Initializer (DSKINT)
Programmer's Manual, DEC-11-UDKIA-A-D

PREFACE

This manual describes the features and operation of VERIFY, a file verification program for the DOS/BATCH operating system. VERIFY is meant to be a tool for use by knowledgeable system programmers or system managers. In simple applications (normal usage) it provides rapid verification that the file structure on a given device is or is not good. If problems are encountered, VERIFY provides diagnostic information which must be interpreted by a knowledgeable person. The causes of problems are many, ranging from hardware malfunctions to system program errors and user errors. VERIFY cannot necessarily correct these errors; it is simply a tool by which experienced programmers can attempt to isolate the causes of such errors.

The manual is divided into four chapters:
Chapter 1 provides an introduction to VERIFY describing the features and capabilities of the program. Chapter 2 describes the operation of the various options available in the VERIFY program. Chapter 3 describes the keyboard commands required to run VERIFY. Chapter 4 describes output from VERIFY -- error messages and informational output.

DOCUMENTATION CONVENTIONS

1. In all examples of keyboard dialogue, system typeouts and program typeouts are underlined, whereas user typeins are not.

2. In all examples of keyboard dialogue and printer output, upper-case letters and numerics represent information printed exactly as shown; lower-case letters represent variable information which is explained in subsequent text.

3. The ↵ symbol indicates that the user must press the RETURN key at the keyboard.

NOTE

The software described in this manual is furnished to purchaser under a license for use on a single computer system and can be copied (with inclusion of DEC's copyright notice) only for use in such system, except as may otherwise be provided in writing by DEC.

This document is for information purposes and is subject to change without notice.

DEC assumes no responsibility for the use or reliability of its software on equipment which is not supplied by DEC.

CONTENTS

CHAPTER 1

INTRODUCTION

VERIFY is a DOS/BATCH program that verifies the file structure on a specified disk or DECtape unit, and informs the user as to the condition of that file structure; i.e, whether the structure is damaged through multiple allocations of a block or other abnormal conditions.

VERIFY is written principally in FORTRAN with one assembly language module that reads absolute blocks on the disk or DECtape and provides other utility routines. VERIFY is usually run with overlays and requires a minimum of 16K of core. When verifying an RP03 disk, VERIFY requires 28K of core.

VERIFY assumes a User File Directory (UFD) format corresponding to DOS/BATCH Monitor version V08 or later. Because of rearrangement of LOCK and USAGE count fields since version V4A, information will not be listed properly for those fields when VERIFY is run with UFD formats prior to version V08. Locked files will also not be detected properly when VERIFY is run with UFD formats prior to version V08.

The following device configurations can be handled by the VERIFY program:

- RK03 disk unit
- RK05 disk unit
- RP03 disk unit
- RF11 disk unit (up to 4 platters)
- DECtape unit

VERIFY assumes that a line printer is available for output, and directs output to FORTRAN device number 5; the keyboard is assumed for input, which is expected from FORTRAN device number 6. Thus, to use VERIFY in Batch Mode, it is necessary to assign Batch Input (BI:) to device numer 6, as shown below.

```
$JOB VERIFY [uic]
$ASSIGN BI:,6
$RUN VERIFY
.
.
program commands (described in Chapter 3)
.
.
$FINISH
```

If the system does not contain a line printer, it is necessary to assign the keyboard to device number 5 before running VERIFY, as shown in the additional statement below.

```
$ASSIGN KB:,5
```

If a hard disk error is encountered during execution of VERIFY, the Monitor generates a fatal error (F035) and aborts the program. There is no way to proceed with the VERIFY program on that device.

CHAPTER 2

VERIFY OPTIONS

When the VERIFY program is run, the user specifies which options he wishes VERIFY to perform. The options, listed below, are described in detail in this chapter.

1. NORMAL
2. LIST
3. SEARCH
4. FIX
5. ALL

2.1 NORMAL OPTION

THE NORMAL option performs the following:

Checks each UIC entry in the Master File Directory (MFD)

Checks each file of each UIC

Constructs a bit map, marking off each block accessed as "in use"

Compares the newly constructed bit map with the map stored on the device, reporting any discrepancies.

When the NORMAL option is chosen, the only information printed at the line printer is error messages, if any. Possible error messages are listed in Chapter 4. When the NORMAL option is completed, the VERIFY program exits, causing a $ to be typed at the keyboard.

2.2 LIST OPTION

The LIST option performs the following, in addition to verification as performed for the NORMAL option:

Issues a line printer listing of the MFD for the device, including the following information.

1. Listing of UIC's on that device
2. First User File Directory (UFD) block for each UIC
3. Size (in words) of the UFD entry for each UIC file

Issues a line printer listing giving data about the hardware and software configuration, including:

1. Device drivers (software) being used by the Monitor
2. Size of the Resident Monitor
3. Physical core size of the system
4. Processor type and options
5. Clocks present, if any
6. Bootstrap present, if any

Issues a line printer listing of the User Directory for each UIC, including the following information:

1. Address of each UFD block for each UIC
2. File names and extensions in each block
3. Creation date of each file
4. Type of file (linked or contiguous) for each file
5. Usage count for each file
6. Lock status for each file
7. Starting block address for each file
8. Length (in blocks) of each file
9. Ending block address for each file
10. Protection code for each file
11. End Byte Pointer (EBP) for each file
12. Total number of blocks occupied by all files for each UIC.
13. Average file size (in blocks, with number of words/block)

Additional identification information is printed for CIL's and Load Modules, including:

1. Whether CIL is hooked to the bootstrap
2. CIL creation time and date
3. CIL block size
4. Size of the CIL
5. Number of blocks used within the CIL
6. Number of modules in the CIL
7. Load Module information, including
   a. .TITLE
   b. .IDENT
   c. creation time and date
   d. size

Issues a line printer listing of the bit maps for the device, showing which blocks are in use.

2.3 SEARCH OPTION

The SEARCH option searches for a user-specified block number in addition to performing verification as in the NORMAL option. If the block number can be located on the device, the SEARCH option prints a message in the following format at the line printer:

```
******SEARCHING FOUND BLOCK nn******
FOR FILE filnam.ext [uic]
```

where nn is the block number, and filnam.ext is the filename and extension of the file containing block nn. The UIC associated with the file is also listed in the message, as shown above.

If the filename and extension are listed as all asterisks (i.e.,******.***), the block is not associated with a file and is usually a UFD, MFD, or MAP block.

If more than one filename and extension are listed for block nn, block nn has been cross-allocated for both files.

If the user-specified block cannot be found in the file, the SEARCH option simply performs a NORMAL verification, and exits.

2.4 FIX OPTION

The FIX option performs verification as described under the NORMAL option. In addition, it replaces current system bit maps with the bit maps generated during the verification process.

The FIX option can be used for disk units only; it can not be used on DECtape units. If disk blocks are lost, the FIX option is an efficient method of repairing the disk bit map; however, if the disk is severely corrupted, the FIX option should not be used without extensive knowledge of the disk problem.

NOTE

> The FIX option should not be used unless the user has very extensive knowledge of the failures being encountered on the file system. If used indiscriminately, the FIX option can corrupt the entire file system on the device being verified.

As a safety precaution, the FIX option is not performed if there are cross-allocated blocks on the disk. In such cases, an error message is issued. To remedy this situation, the user should perform the SEARCH option to detect the files containing cross-allocated blocks; these files should then be copied to another device (via PIP) and deleted from the original device.

2.5 ALL OPTION

The ALL option performs combined functions of the NORMAL , SEARCH, and LIST options, as follows:

performs verification as described under the NORMAL option

provides printer listings as described under the LIST option

searches for a user-specified block as described under the SEARCH option

CHAPTER 3

VERIFY COMMANDS

The VERIFY program can be run any time the Monitor prints a dollar sign ($) at the keyboard, signifying readiness to accept commands. The VERIFY program is called for execution with the following command:

```
RUN VERIFY↵
```

VERIFY then responds with the following keyboard message:

```
VERIFY Vxxx
WHICH DEVICE(SY,DK,DF,DP,DT)?
```

where xxx is the latest version number of the VERIFY program.* The user responds to the above message by specifying the device he wants verified. These device options and their meanings are listed below.

SY↵	system device
DK↵	RK11 or RK05 disk unit
DF↵	RF11 disk unit
DP↵	RP03 disk unit
DT↵	DECtape unit

After the user has responded with the device specification, VERIFY prints the following message (if the device specified is unit-oriented):

```
UNIT NO.
```

The user responds to this message by specifying the unit number of the device he wishes verified.

After the user has specified the unit number, VERIFY prints the following message at the keyboard.

```
OPTION(NORMAL,LIST,SEARCH,FIX, OR ALL)?
```

The user responds to this message with any of the following commands, specifying the VERIFY option he wishes to use.

N↵	NORMAL option
L↵	list option
S↵	SEARCH option
F↵	FIX option (for disk units only)
A↵	ALL option

These options are described in detail in Chapter 2.

----

*Note that the VERIFY program does not type out the # symbol when input is expected from the keyboard. This is because it is a FORTRAN program and cannot read from a logical dataset (in this case, CMI).

If the SEARCH or ALL option is specified, VERIFY prints the following additional message at the keyboard:

```
     BLOCK NUMBER TO SEARCH FOR (06 FORMAT)?
```

The user responds to this message by typing a block number in 06 format (six characters with leading spaces if necessary) from 0 through 177777.

The VERIFY program begins execution as soon as the RETURN key is pressed following the final user command.

Several examples of console dialogue for the VERIFY program are shown below. In each example, VERIFY typeouts are underlined, and user typeins are not underlined.

EXAMPLE 1: The user wishes to verify the system device (a DF disk), using the NORMAL option.

```
$RUN VERIFY↵

VERIFY Vxxx
WHICH DEVICE(SY,DK,DF,DP,DT)?
SY↵

OPTIONS(NORMAL,LIST,SEARCH,FIX, OR ALL)?
N ↵

$
```

Before the final $ is printed at the keyboard, VERIFY lists at the printer all error messages (if any) applicable to the system device.

EXAMPLE 2: The user wishes to verify unit 3 of an RK05 disk, using the SEARCH option to locate block number 177.

```
$RUN VERIFY ↵

VERIFY Vxxx
WHICH DEVICE(SY,DK,DF,DP,DT)?
DK↵

UNIT NO.
3↵

OPTIONS(NORMAL,LIST,SEARCH,FIX, OR ALL)?
S↵

BLOCK NUMBER TO SEARCH FOR (06 FORMAT)?
   177↵

SEARCHING FOR BLOCK      177

$
```

Before the final $ is printed at the keyboard, VERIFY lists at the printer pertinent information for block number 177, if block 177 is in use.

EXAMPLE 3: The user wishes to verify an RF11 disk using the ALL option. In addition to information about all files on the disk, he wishes to know the file(s) and UIC(s) that correspond to block 352 on the disk.

```
$RUN VERIFY ↵

VERIFY Vxxx
WHICH DEVICE(SY,DK,DF,DP,DT)?
DF ↵

OPTIONS(NORMAL,LIST,SEARCH,FIX, OR ALL)?
A ↵

BLOCK NUMBER TO SEARCH FOR(06 FORMAT)?
   352↵

SEARCHING FOR BLOCK    352

$
```

Before the final $ is printed at the keyboard, VERIFY lists at the printer pertinent information about the RF11 disk, including specific information about block number 352.

CHAPTER 4

VERIFY LINE PRINTER OUTPUT

Depending upon the option chosen by the user, the VERIFY program can list either (or both) of two types of information at the line printer. This information can be (1) error messages, or (2) standard VERIFY output.

NOTE

> Most of VERIFY is written in Fortran, and in some cases it may be possible to get an error message printed by the FORTRAN Object Time System. Such messages are explained in detail in the DOS/BATCH FORTRAN Compiler and Object Time System Programmer's Manual.

As a convention in numeric printouts, decimal numbers are followed by a decimal point, whereas octal numbers are not.

4.1 Error Messages

Table 4-1 lists error messages that can be produced at the line printer by the VERIFY program. The first line of an error message is delimited by asterisks, thus:

```
* * * * * * message * * * * * * *
```

Table 4-1

VERIFY Error Messages

Message	Meaning
BLOCK NUMBER OUT OF RANGE = n	This message can occur during execution of any VERIFY option. It is produced by the subroutine MASK during bit map reconstruction. It indicates that an octal block number (n) is too large for the array MAP, which is allocated by VERIFY to hold the reconstructed bit maps. One of two conditions is possible: (1) The named block is garbage in the file, and is a valid error.

(2) VERIFY is not capable of handling a disk being verified. VERIFY typically has enough space in MAP to verify an RK disk, RP disk, or 4-platter RF disk.

```
DEVICE ERROR DURING TRAN
BLOCK NUMBER  = n
FUNCTION WORD = f
```

A hardware error has been encountered during a TRAN operation. The octal number of the block being read (n) and the octal number of the TRAN function word (f) are listed. See the DOS/BATCH Monitor Programmer's Manual for a complete description of the TRAN function word.

```
FILE SIZE IS INCORRECT
filnam.ext[uic]
SYSTEM THINKS FILE SIZE = s
ACTUAL FILE SIZE = a
```

The file (filnam.ext) with UIC (uic) has an incorrect file length. The number of blocks actually used by the file (a) does not agree with the size (s) stored in the file directory.

```
FILE END BLOCK IS INCORRECT
filnam.ext[uic]
SYSTEM THINKS END BLOCK = e
ACTUAL END BLOCK = a
```

The file (filnam.ext) with UIC (uic) has incorrect directory information. The last block used by the file (a) is different from the last block (e) indicated by the file directory.

```
MAP VERIFICATION ERROR
MAP NUMBER =  n, WORD NUMBER = w
RECONSTRUCTED ENTRY = rrrrrr
SYSTEM ENTRY = ssssss
(LOST)
(    )BLOCK NUMBER = d DECIMAL(o OCTAL)
(FREE)


TOTAL FREE BLOCKS = f TOTAL LOST = t
```

This message is printed during map verification when the bit map on the device does not agree with the recontructed bit map produced by VERIFY. The map number (n) and word number (w) are provided. The reconstructed bit map entry (rrrrrr) and system bit map entry (ssssss) are followed by a list of the block numbers (both decimal and octal) of the blocks in error. A block in error is marked as FREE or LOST. A FREE block is one marked as unused (bit = 0) in the system map, but is used (bit = 1) according to the reconstructed map. A LOST block is one marked as used (bit=1) in the system map, but is unused (bit=0) according to the reconstructed map. LOST blocks cannot be used by any file, but do not endanger file integrity on the device. If the device being verified is a disk, the FIX option of VERIFY can be run to recover LOST blocks. However, a file containing one or more FREE blocks can be damaged by other files created later on the device. Files containing FREE blocks should be copied (to another device) using the PIP program, and then deleted from the original device. To determine the names of the files containing FREE blocks, run VERIFY in search mode (SEARCH or ALL option), specifying the block(s) in error. After all map verification error messages have been listed at the printer.

LOCKED FILE
filnam.ext[uic]

The file (filnam.ext) with UIC (uic) was locked, and therefore in an error state, when VERIFY attempted to perform file verification on it. The UNLOCK switch (/UN) in the PIP program can be used to unlock this file; once the lock bit has been turned off, the file can be renamed or deleted.

OPEN FILE, USAGE COUNT = n
filnam.ext[uic]

The file (filnam.ext) with UIC (uic) has a nonzero USAGE count. This indicates that the file is in an intermdiate state, and is likely to be the cause of bit map errors. The file's USAGE count can be zeroed through use of the UNLOCK switch (/UN) in the PIP program.

MULTIPLE ALLOCATION OF BLOCK n
filnam.ext[uic]

The file (filnam.ext) with UIC (uic) contains octal block n, which is also contained in another file on the same device. This indicates that two or more files are "cross-allocated," i.e., they intersect at block n. This situation is dangerous, since deletion of one of those files would cause part of the other(s) to be deleted also. For linked files, the data contained in such cross-allocated blocks is data for the file created most recently. It may be possible to recover all or part of such crossed files by transferring them individually to other devices through the PIP program. When two or more files are found to intersect, none should be deleted until the desired files have been backed up by PIP transfers.

This message is not sufficient by itself to identify all intersecting files for block n. After this error message has been issued, VERIFY should be run in search mode (SEARCH or ALL option) to identify all files that intersect at block n.

4.2 Standard Verify Output

Depending upon the option chosen, VERIFY lists certain information at the line printer during execution. For the NORMAL option, only error messages are listed, as described earlier in this chapter. For all other options, one or more of the following can be listed:

Listing of the UFD for the current UIC on the device being verified.

One or more bit maps for the device

File name and UIC information for a specified block.

4.2.1 MFD Listing

When the LIST, FIX, or ALL option is chosen, an MFD listing is produced at the line printer, as shown below:

```
VERIFY Vxxx

* * * * * LISTING OF MFD FOR dev * * * * * ON dd-mmm-yy AT hh:mm:ss

      UIC  FIRST UFD BLOCK  UFD ENTRY SIZE

[ uic ]                bbb              s
  .                     .               .
  .                     .               .
  .                     .               .
```

where aa-a = current version number of VERIFY (e.g.,03-3)
dev = name of the device being verified (e.g.,DK2)
dd-mmm-yy = current date in day-month-year format (e.g.,28-FEB-73)
hh:mm:ss = current time in hours:minutes:seconds format (e.g., 16:45:35)
uic = each UIC in the MFD
bbb = octal address of the first UFD block for each UIC
s = size in words (decimal) of the UFD entry for each UIC

4.2.2 UFD Listing

When the LIST, FIX, or ALL option is chosen, a User File Directory (UFD) listing for the current UIC is produced at the line printer, as shown below:

```
* * * * * * LISTING OF [ uic ] USER DIRECTORY * * * * * *

  FILE EXT       DATE  TYPE   USAGE LOCK START  LENGTH  END  PROT  EBP

UFD BLOCK = aaa

filnam.ext dd-mmm-yy     t       u    k     s       n     e     p    b
    .          .         .       .    .     .       .     .     .    .
    .          .         .       .    .     .       .     .     .    .
    .          .         .       .    .     .       .     .     .    .


TOTAL BLOCKS = ttt
AVERAGE FILE SIZE = cc BLOCKS, www WORDS EACH
```

```
WHERE uic     = current UIC
      aaa     = octal address of the UFD block containing  entries  for
                files listed.
filnam.ext    = filename and extension for each file under the  current
                UIC
dd-mmm-yy     = date of creation of each file in day-month-year format
            t = file type (L = linked, C = continguous)
            u = USAGE count for the file (normally 0)
            k = LOCK count for the file (normally 0)
            s = octal address of the first block of the file
            n = length of the file in blocks (decimal)
            e = octal address of the last block of the file
            p = octal protection code of the file
            b = End Byte Pointer for the file
          ttt = total number of blocks used by files on the device
           cc = average file size  in  blocks  (decimal),  obtained  by
                dividing total blocks used (ttt) by number of files
          www = number of words per block (depends upon device)
```

4.2.3 MAP Listing

When the LIST, FIX, or ALL option is chosen, one or more bit maps (depending upon device) are listed at the printer, as shown on next page:

```
* * * * * * MAP VERIFICATION * * * * * *
* * * * * * MAP HEADER INFORMATION * * * * * *

LINK = r
MAP NUMBER = s
WORDS IN MAP = t
LINK TO FIRST MAP = u

bbbbbb bbbbbb bbbbbb bbbbbb bbbbbb bbbbbb bbbbbb bbbbbb bbbbbb  bbbbbb
bbbbbb bbbbbb ...
```

```
where bbbbbb = octal representation of block usage (file data)
           r  = link to next bit map
           s  = number of this map
           t  = number of data words in this map
           u  = link to first bit map
```

Block usage is specified in units of six octal digits, which are formed from 16-bit groups (in which each bit represents one block). The first four words in a bit map contain map header information and the remaining items in the map represent actual file usage data (bbbbbb). A bit is on (has a value of 1) if the block it represents is used; a bit is off (has a value of 0) if the block it represents is unused. Blocks are represented in increasing order from right to left within items; items are represented in increasing order from left to right in the map, as shown in the examples below:

EXAMPLE 1: File data item #1 = 167356
16-bit expansion = 1110111011101110

The first file data item is actually the fifth item in the bit map, since the first four items contain map header data. This item thus represents data blocks 0-15 decimal (0-17 octal) The bit expansion, reading from right to left, shows that the 0th 4th, 8th, and 12th blcoks are unused (octal blocks 0, 4, 10, and 14) the remainder of blocks within the group are used.

EXAMPLE 2: File data item #5 = 424
16-bit expansion = 0000000100010100

this item represents data blocks 64-79 decimal (100-117 octal). The bit expansion, reading from right to left, shows that the 2nd, 4th, and 8th blocks are used (octal blocks 102, 104, and 110); the remainder of blocks within the group are unused.

EXAMPLE 3: File data item #13 = 177777
16-bit expansion = 1111111111111111

This item represents data blocks 192-207 decimal (300-317 octal). The bit expansion shows that all blocks within the group are used.

NOTE

When a bit representing a data block is on (i.e., has a value of 1), it indicates that the corresponding block is used, OR that the corresponding block does not exist on the device (i.e., block address is too high). The user should be able to determine whether the last few items in a bit map represent used blocks or nonexistent blocks, if he knows the capacity of the device being verified.

4.2.4 Search Information For a Block

If the SEARCH or ALL option is chosen, the VERIFY program searches for a user-specified block, and lists information about that block at the printer, as shown below:

```
* * * * * * SEARCHING FOUND BLOCK   bbb * * * * * *
FOR FILE filnam.ext [uic]
```

where bbb = octal address of the user-specified block

filnam.ext = name and extension of the file containing the specified block

uic = UIC for the file containing the block

Interpretation of the above information is described in Chapter 2 under "SEARCH Option."

INDEX

HOW TO OBTAIN SOFTWARE INFORMATION

Announcements for new and revised software, as well as programming notes, software problems, and documentation corrections, are published by Software Information Service in the following newsletters.

DIGITAL Software News for the PDP-8 and PDP-12
DIGITAL Software News for the PDP-11
DIGITAL Software News for 18-bit Computers

These newsletters contain information applicable to software available from DIGITAL'S Software Distribution Center. Articles in DIGITAL Software News update the cumulative Software Performance Summary which is included in each basic kit of system software for new computers. To assure that the monthly DIGITAL Software News is sent to the appropriate software contact at your installation, please check with the Software Specialist or Sales Engineer at your nearest DIGITAL office.

Questions or problems concerning DIGITAL'S software should be reported to the Software Specialist. If no Software Specialist is available, please send a Software Performance Report form with details of the problems to:

Digital Equipment Corporation
Software Information Service
Software Engineering and Services
Maynard, Massachusetts 01754

These forms, which are provided in the software kit, should be fully completed and accompanied by terminal output as well as listings or tapes of the user program to facilitate a complete investigation. An answer will be sent to the individual, and appropriate topics of general interest will be printed in the newsletter.

Orders for new and revised software manuals, additional Software Performance Report forms, and software price lists should be directed to the nearest DIGITAL field office or representative. USA customers may order directly from the Software Distribution Center in Maynard. When ordering, include the code number and a brief description of the software requested.

Digital Equipment Computer Users Society (DECUS) maintains a user library and publishes a catalog of programs as well as the DECUSCOPE magazine for its members and non-members who request it. For further information, please write to:

Digital Equipment Corporation
DECUS
Software Engineering and Services
Maynard, Massachusetts 01754

DOS/BATCH
Verification Program
(VERIFY)
Programmer's Manual
DEC-11-UVERA-A-D

READER'S COMMENTS

Digital Equipment Corporation maintains a continuous effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback--your critical evaluation of this document.

Did you find errors in this document? If so, please specify by page.

____

____

____

____

____

How can this document be improved?

____

____

____

____

____

How does this document compare with other technical documents you have read?

____

____

____

____

____

Job Title__________________________________Date:________________

Name:___________________________Organization:___________________

Street:_________________________Department:_____________________

City:___________________State:__________Zip or Country___________

-- Fold Here --

-- Do Not Tear - Fold Here and Staple --