PART 14

THE DOS/BATCH VERIFY PROGRAM

PART 14

CHAPTER 1

INTRODUCTION TO VERIFY

VERIFY is a DOS/BATCH program that evaluates the file structure on a specified disk or DECtape unit and informs the user if the structure is damaged through multiple allocation of a block or if other abnormal conditions exist. VERIFY's diagnostic information covers hardware malfunctions, system program errors, and user errors.

VERIFY is intended primarily for use by experienced programmers. It is written principally in FORTRAN, with one assembly language module. VERIFY is usually run with overlays and requires a minimum of 16K of core. However, when verifying an RPØ3 disk, 24K of core is needed.

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

- RKØ3 disk unit
- RKØ5 disk unit
- RPØ3 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. To use VERIFY in Batch Mode, it is necessary to assign Batch Input (BI:) to device number 6, as shown below.

```
$JOB VERIFY [uic]
$ASSIGN BI:,6
$RUN VERIFY
.
.
program commands (described in Chapter 14-3)
.
.
$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 hardware disk error is encountered during execution of VERIFY, the Monitor generates a fatal error (FØ35) and aborts the program. There is no way to proceed with the VERIFY program on that device.

PART 14

CHAPTER 2

VERIFY OPTIONS

The user-specified options in the VERIFY program are:

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

2.1 NORMAL OPTION

The NORMAL option performs the following:

a. Checks each UIC entry in the Master File Directory (MFD),
b. Checks each file of each UIC,
c. Constructs a bit map, marking off each block accessed as "in use", and
d. Compares the newly constructed bit map with the map stored on the device, reporting any discrepancies.

If the NORMAL option is chosen, the only information printed on the specified device is error messages, if any. When the NORMAL option is completed, the VERIFY program exits, causing a $ to be printed at the keyboard terminal.

2.2 LIST OPTION

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

a. Issues a 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, and
    3. Size (in words) of the UFD entry for each UIC file.

b. Issues a 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, and
6. Bootstrap present, if any.

c. Issues a 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, and
13. Average file size (in blocks, with number of words/block).

d. Prints additional identification information for CIL's[1] 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.

---

[1]For more information on CIL, the Core Image Library, see Part 1.

e. Issues a 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 on the specified device:

```
******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 message also lists the UIC associated with the file 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 failure 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 file containing cross-allocated blocks; these file 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:

1. Performs verification as described under the NORMAL option,
2. Provides listings as described under the LIST option, and
3. Searches for a user-specified block as described under the SEARCH option.

PART 14

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[1]. The user responds to the above message by specifying the device he wants verified or by pressing the RETURN key which performs a NORMAL verification on the system device. Device options and their meanings are listed below.

SY	system device
DK	RKØ3 or RKØ5 disk unit
DF	RF11 disk unit
DP	RPØ3 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)?
```

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

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 14-2.

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 Ø 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.

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 terminal, VERIFY lists all error messages (if any) applicable to the system device.

EXAMPLE 2: The user wishes to verify unit 3 of an RKØ5 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 (O6 FORMAT)?
177

SEARCHING FOR BLOCK 177

$
```

Before the final $ is printed at the keyboard terminal, VERIFY lists 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 (O6 FORMAT)?
352

SEARCHING FOR BLOCK 352

$
```

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

PART 14

CHAPTER 4

VERIFY OUTPUT

Depending upon the option chosen by the user, the VERIFY program can list either (or both) of two types of information: (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 Appendix K.

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

4.1 STANDARD VERIFY OUTPUT

Depending upon the option chosen, VERIFY lists certain information during execution. For the NORMAL option, only error messages are listed. For all other options, one or more of the following can be listed:

1. Listing of the UFD for the current UIC on the device being verified,
2. One or more bit maps for the device,
3. File name and UIC information for a specified block.

4.1.1 MFD Listing

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

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

     UIC   FIRST UFD BLOCK   UFD ENTRY SIZE

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

where: Vaa-a = current version number of VERIFY (e.g., Ø3-3),
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, and
s = size in words (decimal) of the UFD entry for each UIC.

4.1.2 UFD Listing

When the LIST, FIX, or ALL option is chosen, a User File Directory (UFD) listing for the current UIC is produced 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 = contiguous),
u = USAGE count for the file (normally Ø),
k = LOCK count for the file (normally Ø),
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, and
www = number of words per block (depends upon device).

4.1.3 MAP Listing

When the LIST, FIX, or ALL option is chosen, one or more bit maps (depending upon the device) are listed at the specified output device.

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


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

bbbbb bbbbb bbbbb bbbbb bbbbb bbbbb bbbbb bbbbb bbbbb bbbbb
bbbbb bbbbb ...
```

where: bbbbb = 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, and
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 (bbbbb). A bit is on (has a value of 1) if the block it represents is used; a bit is off (has a value of Ø) 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 = 111Ø111Ø111Ø111Ø

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 Ø-15 decimal (Ø-17 octal). The bit expansion, reading from right to left, shows that the Øth, 4th, 8th, and 12th blocks are unused (octal blocks Ø, 4, 1Ø, and 14) the remainder of blocks within the group are used.

EXAMPLE 2: File data item #5 = 424
16-bit expansion = ØØØØØØØ1ØØØ1Ø1ØØ

This item represents data blocks 64-79 decimal (1ØØ-117 octal). The bit expansion, redding from right to left, shows that the 2nd, 4th, and 8th blocks are used (octal blocks 1Ø2, 1Ø4, and 11Ø); 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 192-2Ø7 decimal (3ØØ-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.

4.1.4 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 specified output device, 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, and

uic = UIC for the file containing the block.

Interpretation of the above information is described in Section 14-2.3.