RECEIVED BY TIC DEC 16 1975

NOVEMBER 1975

MATT-1178

VIPER: A PROGRAM FOR
MANIPULATING CARD-IMAGE
FILES

BY

WILLIAM G. PRICE, JR.

PLASMA PHYSICS
LABORATORY

MASTER

PRINCETON UNIVERSITY
PRINCETON, NEW JERSEY

This work was supported by U. S. Energy Research and Development Administration Contract E(11-1)-3073. Reproduction, translation, publication, use and disposal, in whole or in part, by or for the United States Government is permitted.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Printed in the United States of America.

Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151

Price: Printed Copy \$ * ; Microfiche \$1.45

<u>*Pages</u>	<u>NTIS Selling Price</u>
1-50	\$ 4.00
51-150	5.45
151-325	7.60
326-500	10.60
501-1000	13.60

VIPER: A Program for Manipulating Card-Image Files.

by

William G. Price, Jr.

Plasma Physics Laboratory, Princeton University

Princeton, New Jersey 08540

ABSTRACT

VIPER is a program designed to allow the easy manipulation of card-image files in IBM OS/360 systems. It combines, improves, and extends the features found in the earlier programs MANAGERY, EDITY, and CLICHE.

VIPER may be applied to both sequential and partitioned files, which may be created, replaced, renamed, deleted, copied, edited, listed, or punched. Directory and compress operations can be applied to partitioned data sets.

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Principal Features

VIPER is a program designed to allow the easy manipulation of card-image files in IBM OS/360 systems. It combines, improves, and extends the features found in the earlier programs MANAGERY, EDITY, and CLICHE.

VIPER may be applied to both sequential and partitioned files, and more than one file may be processed during a single job step. Files may be created, replaced, renamed, deleted, copied, listed, and punched. Directory, list-all, punch-all, and compress operations can be applied to partitioned data sets. Implicit inclusion of blocks of cards specified by code names can be requested.

The EDIT command activates a subsystem which will edit one file into another or back into itself. In addition to the usual deletion and insertion by sequence number, this subsystem allows forward searching for a specified string, and replacement of substrings rather than whole cards.

Although VIPER has been domesticated, it will still bite if provoked. In particular, the use of logical records not 80 bytes long may be successful but is not recommended.

Syntax of VIPER

VIPER is controlled by a command input file assumed to consist of 80-column card images, with optional labels in columns 73-80. The first character (col. 1) on the first card is picked up as the "command character" which flags all subsequent commands. The input file requires no termination other than its natural end-of-file.

Each card in the command file which starts with the command character is scanned for a "command code word" and two parameter strings. The command code word is a string starting in column 2 and ending at the first blank; blanks also delimit the parameters. Each string is truncated after the first 8 characters. The whole scan is restricted to columns 2-72, with missing parameters taken to be blanks.

Each command card is listed in the VIPER output file, along with the time and the assumed decomposition into command and parameters. If Column 2 is blank, i.e. a command code word is not given, the card is assumed to be a comment and is ignored. Other acceptable commands are described below.

VIPER Commands

All the VIPER command code words can be abbreviated to three characters. Each command description below is headed by the full code word, with the abbreviation underlined, and the command parameters, set off by quotes.

PDSNAME "NAME"

This fundamental command puts VIPER into PDS mode (as opposed to SEQ mode). Subsequently the object files being manipulated are assumed to be individual members of the Partitioned Data Set specified on the DD card with ddname=NAME. The data set name and volume serial number specified by this ddname are listed in the output file.

SEQNAME "NAME"

This fundamental command puts VIPER into SEQ mode (as opposed to PDS mode). Subsequently the object files being manipulated are assumed to be whole sequential data sets. The parameter NAME is optional. If it is present, the data set name and volume serial number specified by the DD card with ddname=NAME are listed on the output file. SEQ mode is the initial default.

ABEND "SWITCH"

This command is used to allow or suppress an Abend after detection of a logical or I/O error. When SWITCH=ON (the initial default) error processing is terminated with a deliberate Abend with UCC=8. A core dump will be written if a SYSUDUMP DD card is present. When SWITCH=OFF, error processing ends with a normal return to the system with Condition Code=8. (Normal VIPER termination sets COND=0.)

CREATE "NAME1" "NAME2"

This command is used to newly create file "NAME1" from "NAME2". In SEQ mode, NAME1 is assumed to be the ddname of a DD card defining a sequential data set which will be opened and written into. In PDS mode, NAME1 is assumed to be the name of a new member to be created in the partitioned data set specified when PDS mode was entered. Specification of an already-existing member name is an error. In either mode, NAME2 is optional. If it is present, it is assumed to be the ddname of a DD card specifying a sequential data set which is to be read and copied, card-by-card, up to an end-of-file, into NAME1. If NAME2 is absent, cards are read instead from the

command input file and copied to NAME1 until a VIPER command is flagged in column 1 or the end-of-file is detected.

REPLACE "NAME1" "NAME2"

This command is nearly identical to CREATE. The exception is that in PDS mode, specification of an existing member name in NAME1 is legal and causes replacement of that member.

DELETE "NAME"

This command is legal only in PDS mode. It causes the deletion of the member named NAME in the partitioned data set specified when PDS mode was entered. This command is not restricted to card image files.

RENAME "NAME1" "NAME2"

This command is legal only in PDS mode. It causes the member named NAME1 to be renamed NAME2 in the partitioned data set specified when PDS mode was entered. This command is not restricted to card image files.

COPY

"NAME1" "NAME2"

In SEQ mode, NAME1 and NAME2 are assumed to be the ddnames of DD cards specifying sequential data sets, and NAME1 is copied up to its end-of-file into NAME2. In PDS mode NAME1 is assumed to be the ddname of a DD card specifying a partitioned data set, while NAME2 is optional. If it is present, a member named NAME2 is replaced (or created if not yet existing) in the PDS data set, and it is filled from the member named NAME2 in the NAME1 partitioned data set. If NAME2 is absent, all the members in NAME1 are copied to the PDS data set, with creation or replacement of members as necessary. Automatic merging is allowed when a %MERGE Card is encountered if the MERGE ON command (described below) is in effect.

DIRECTORY

"NAME" (note the Y is optional)

In SEQ mode NAME is required, and this command first performs the functions of the PDSNAME command (i.e. PDS mode is entered). In PDS mode NAME is optional, but triggers the PDSNAME functions when it is present.

The Directory command proceeds to give a directory in alphameric order of the members in the PDS data set.

This listing gives the name of the member, its starting location (by relative track and block within the data set, in decimal digits), the directory code byte, and whatever "user data" is in the data set directory. If the first digit of the code byte is >7 then the name is an alias. If the last alteration of the member was done by VIPER, the "user data" consists of the date of alteration, the current number of cards in the member, and the time of the alteration.

The DIRECTORY command is not restricted to card-image files. In particular, it may be applied to libraries of load modules.

COMPRESS "NAME1" "NAME2"

This command is valid only in PDS mode; it causes the IBM Utility program IEBCOPY to be loaded into core and executed. NAME1 and NAME2 are optional. When both are absent, an in-place compress is performed on the PDS data set. (This is the operation which squeezes unused space out to the end of a partitioned data set.) When only NAME1 is present, it is assumed to be the ddname of a DD card specifying a partitioned data set, and the compress is performed

on this data set without changing the PDS data set name. If both NAME1 and NAME2 are present, they are both assumed to be ddnames of DD cards specifying partitioned data sets, and all the members in NAME1 are copied to NAME2, with member creation and replacement as necessary. Since all these operations are performed by IEBCOPY, this command is not restricted to card-image files. In particular, it may be applied to load module libraries.

LABEL "LABEL" "NAME"

This command inserts a label and sequence number on each card in the object file. LABEL is required; its first four characters are put into columns 73-76 of every card. A sequence number starting at 1 and incremented by 1 (reset to 0 at 10000) is placed in columns 77-80. In SEQ mode NAME is required, and is assumed to be the ddname of a DD card specifying a sequential data set to be sequenced as described. In PDS mode NAME is optional. If it is present, it is assumed to be the name of the member, in the PDS data set, which is to be sequenced. If NAME is absent, every member in the PDS data set is sequenced with the same LABEL, but each starting with the number 1.

ALTER "NAME" "NUMBER"

This is a primitive editing command which allows the total replacement of a single card in the object file by the next card in the command input file. NUMBER must be a string of decimal digits equal to the sequence number in the object file of the card to be replaced. In SEQ mode, NAME is assumed to be the ddname of a DD card which specifies the sequential data set requiring alteration. In PDS mode, NAME is assumed to be the name of the member, in the PDS data set, which requires alteration.

MERGE "SWITCH" "NAME"

This command is used to allow or suppress merging - the automatic insertion into the object file of members retrieved from a special Merge.partitioned data set. NAME is optional; if present it is assumed to be the ddname of a DD card which specifies the Merge partitioned data set. (It is legal for this to be the same as the PDS ddname.) If SWITCH=OFF, merging is suppressed; if SWITCH=ON, merging is allowed. (Note that a MERGE OFF command may be required in order to define a Merge data set to be used by the EDIT subsystem's explicit GET command, described in a later section.)

When implicit Merging is allowed, each card in the object file is scanned during LIST, FLIST, COPY, PUNCH, and EDIT operations for the characters "%MERGE " in columns 1-7. When such a card is encountered, it is scanned further, in columns 8-72, for a name of 1 to 8 characters, delimited by blanks. In PDS mode the Merge name is assumed to be the name of a member (the Merge file) in the partitioned data set specified by the MERGE command. In SEQ mode the Merge name is assumed to be the ddname of a DD card for a sequential data set, which becomes the Merge file. All the cards in the Merge file are substituted for the %MERGE card. Only this one level of implicit merging is recognized; any %MERGE cards in the Merge file are simply copied.

LIST "NAME"

This command requests an ordinary list of the cards in the object file into the VIPER output file. In SEQ mode, NAME is assumed to be the ddname of a DD card which specifies the sequential data set to be listed. In PDS mode NAME is optional; when present it is assumed to be the name of the member, in the PDS data set, listed. If NAME is absent, all the members in the PDS data set are listed (the

"LIST-ALL" command) in alphameric order. Each listing is begun with a page eject. Each card image is bracketed and is labeled on both the right and left sides with a card sequence number. When automatic merging is allowed and a %MERGE card is encountered: this card is listed in its proper place; then the bracket characters are changed from "><" to "+ +"; and the merged cards are listed with their own sequence numbers.

FLIST "NAME"

This command is almost identical to the LIST command. The exception is that a page eject is issued whenever a card beginning with the 16 characters " END " (i.e. a FORTRAN END-card) is encountered.

EXCLUDE "TEST"

If the parameter TEST is present, its first character is saved for use during any subsequent PDS-mode PUNCH commands. Any member name whose first character matches this Exclude character will be excluded from a "Punch-all" operation. If TEST is not present, such an Exclude check will not be performed.

PUNCH "NAME1" "NAME2"

This command is legal only in PDS mode. NAME1 is required, and is assumed to be the ddname of a DD card which specifies a sequential data set to receive the punched output. This "punching" is really a card-by-card copy from a partitioned data set member to a sequential data set, and true punching occurs only if the NAME1 DD specifies SYSOUT=B.

NAME2 is optional; if present it is assumed to be the name of the member, in the PDS data set, to be punched. If NAME2 is absent, then all the members are punched (the Punch-all option) in alphameric order, except for those screened out by the Exclude test. Automatic merging is allowed when a %MERGE card is encountered if the MERGE ON command is in effect.

CLICHE "NAME1" "NAME2"

This command performs the basic functions of the CLICHE program. NAME1 and NAME2 are both assumed to be the ddnames of DD card specifying sequential data sets. The NAME1 file starts with a number of card sets each preceeded by a "***CREATE NAME" card and

followed by a "***END" card. Following these sets is a card stream which may contain "***INSERT NAME" cards. The CLICHE command processor scans past the last **END card, and then copies cards from NAME1 to NAME2; but any **INSERT card is replaced by the deck of cards between the **CREATE and **END cards which it designates. Since these decks are stored as members in a temporary partitioned data set there is no limit on the number which may be created. (Note that NAME1 may actually specify a concatenated series of members stored in a partitioned data set.) It is felt that the Merge facility can provide service superior to CLICHE.

EDIT

"NAME1" "NAME2"

This command activates the EDIT Subsystem. NAME1 is required and specifies the object file to be edited. NAME2 is optional; if present it receives the edited version of NAME1. If NAME2 is absent, the object file is edited into a temporary file (named "VIPERUT1") which is copied back onto the NAME1 file just before EDIT returns to VIPER. In SEQ mode, NAME1 and NAME2 are assumed to be the ddnames of DD cards specifying sequential data sets. In PDS mode, NAME1 and NAME2 are assumed to be the names of

members in the PDS data set. (NAME2 will be created or replaced, as necessary.)

Syntax of EDIT

The EDIT Subsystem is controlled by records in the VIPER command input file following the EDIT command. These records are assumed to be 80-column card images with optional labels in columns 73-80. The first character (col. 1) on the first card after the EDIT command is picked up as the EDIT "command character". This character flags subsequent commands during the EDIT operation.

An EDIT operation is normally terminated by an end-of-file on the command file, by execution of a HALT command, or by detection of a VIPER command in the command file. Note that different EDIT command characters may be established for successive EDIT operations; the command character can never, however, be the same as that for VIPER.

Each card in the command file which starts with the EDIT command character is scanned for a "command code word" and two parameter strings. The command code word is a string starting in column 2 and ending at the first blank; blanks also delimit the parameters. The whole scan is restricted to columns 2-72, with missing parameters taken to be blanks.

Each command card is listed in the EDIT output file, which is the VIPER output file. If Column 2 is blank i.e.

a command code word is not given, the card is assumed to be a comment and is ignored. Other acceptable commands are described below.

EDIT Commands

All the EDIT command code words can be abbreviated to one character. Each command description below is headed by the full code word, with the abbreviation underlined, and its symbolic parameters. Here "M" and "N" represent digit strings which give a card sequence number in the object file. Each sequence number is relative to an origin of 0 which initially is set to proceed the first card, but which may be reset by the FIND command.

PRINT "SWITCH"

This command controls the listing of cards from the object file into the output file. When SWITCH=OFF, no cards are listed; when SWITCH=ON, all cards are listed. When SWITCH=CHANGES then only cards which are added, deleted, or altered are listed. Each card listed is preceded, as appropriate, by its sequence number in the input file and/or its sequence number in the output file. CHANGES is the default specification which is established whenever EDIT is entered.

LABEL "SWITCH"

SWITCH=ON is the default which is established whenever EDIT is entered. In this condition each object card is assumed to consist of 72 columns of data with an 8-column label. Only the first 72 columns are scanned by the FIND and SWAP commands. The label field in the output file is over-written with the current output sequence number in columns 73-76 and the current input sequence number in columns 77-80; this conveniently ties the edited output to the input file. When SWITCH=OFF, the cards are assumed to consist of 80 columns of data. All 80 columns are scanned by the FIND and SWAP commands, and the label described above is not generated.

HALT "M"

This command copies cards from the input file, through sequence #M, to the output file without change. It then terminates the EDIT operation and returns control to VIPER. If "M" is omitted, the input file is copied through its end-of-file.

An end-of-file when attempting to read an EDIT command or the discovery of a VIPER command generates

an implicit HALT command for copying through the end-of-file.

GET "M" "NAME"

This is a command which requests the explicit inclusion of some remote deck of cards. It complements the "%MERGE" facility, which is set on or off by the VIPER command MERGE. The GET command first copies cards from the input file through sequence #M. If the VIPER mode is SEQ, NAME is assumed to be the ddname of a DD card specifying a sequential data set from which cards are to be copied (through end-of-file) to the output file. If the mode is PDS, then NAME is assumed to be the name of a member, in the Merge partitioned data set, from which cards are to be copied (through end-of-file) to the output file. Note that a Merge partitioned data set must have previously been defined by a VIPER "MERGE ON or OFF NAME" command for this to be valid. Finally any following cards in the command file are copied to the output file until an EDIT command, VIPER command, or end-of-file is detected.

AFTER

"M"

This command requests that cards be copied, through sequence #M, from the input file to the output file. Then any following cards in the command file are copied to the output file until an EDIT command, VIPER command, or end-of-file is detected.

BEFORE

"M"

This command requests that cards be copied, up to but not including sequence #M, from the input file to the output file. Then any following cards in the command file are copied to the output file until an EDIT command, VIPER command, or end-of-file is detected.

CHANGE

"M" "N"

This command combines a deletion and an insertion. First cards are copied, up to but not including sequence #M, from the input file to the output file. Then cards with sequence numbers through N are read and discarded; if N is omitted, it is set equal to M. Finally, any following cards in the command file are copied to the output file until an EDIT command, VIPER command, or end-of-file is detected.

DELETE "M" "N"

This command requests that cards be copied, up to but not including sequence #M, from the input file to the output file. Then cards with sequence numbers through N are read and discarded. If N is omitted, it is set equal to M. Finally, any following cards in the command file are copied to the output file until an EDIT command, VIPER command, or end-of-file is detected.

ELECT "M" "N"

This command combines a delete and a copy. First, cards are read, up to but not including sequence #M, from the input file and discarded. Then cards with sequence numbers through N are copied from the input file to the output file; if N is omitted it is set equal to M. Finally, any following cards in the command file are copied to the output file until an EDIT command, VIPER command, or end-of-file is detected.

FIND "L" 'STRING'

This command serves two purposes. If "L" is an asterisk *, cards are copied from the input file

through its end-of-file into the output file, and the origin of the sequence number counter is set to 0. Then any following cards in the command file are copied to the output file until an EDIT command, VIPER command, or end-of-file is detected.

If "L" is a digit string (possibly prefixed with a minus sign), it is assumed to be a card column number. The rest of the command card is now scanned for a test string in the following manner: the first non-blank character beyond L is taken to be a delimiter, and the test string consists of those characters (which may include blanks) between the first delimiter and its next occurrence, e.g. 'STRING' or (STRING) both would imply STRING. Cards are read from the input file and checked for a match to the test string starting in column #L. (L-1 + string length must fit in the 72 or 80 column data area, as set by LABEL.) Cards which do not match are copied to the output file unless the first character of "L" is a minus sign. When a match is found, the input file is backspaced, and any following cards in the command file are copied to the output file until an EDIT command, VIPER command, or end-of-file is detected. Finally the sequence counter origin is reset to 0, so that the matching card becomes sequence #1.

SWAP

"M" 'OLDSTRING'NEWSTRING'

This command allows the replacement of a given substring on a card by another string. First cards are copied, up to but not including sequence #M, from the input file to the output file. Then the command card is scanned for two strings in the following manner: the first non-blank character beyond "M" is taken to be a delimiter, and the OLDSTRING and NEWSTRING are delimited by the first, second, and third occurrences of this delimiter character on the card. Thus any character, including the blank, may be included in either string.

The card with sequence #M is read from the input file, and scanned throughout its data area (72 or 80 columns, as set by LABEL) for a match to the OLDSTRING, which is replaced by NEWSTRING. If NEWSTRING is shorter, the remaining data area is shifted left and padded with blanks. If NEWSTRING is longer, the remaining data is shifted right. Excess blanks will be truncated, but an overflow of non-blanks is illegal. If "M" is an asterisk *, the SWAP operation is applied to all cards from the input file, up to its end-of-file.

Finally, any following cards in the command file are copied to the output file until an EDIT command, VIPER command, or end-of-file is detected.

Execution and Data Sets

A substantial number of IBM/360 Job Control Language statements are required to execute VIPER. For convenience, most of these have been incorporated into a catalogued procedure, and thus are provided automatically to the user. The following description of the procedure for using VIPER is unavoidably keyed to the Operating System and file structure on the Princeton University Computer Center circa July 1975.

The VIPER procedure is stored in the file "U.RS.PROCLIB", thus the card `//*PROCLIB=U.RS.PROCLIB` must be included to make it available. VIPER is executed by naming the procedure on an EXEC card, e.g.

```
//BEWARE EXEC VIPER .
```

Three optional parameters may follow the procedure name. Each consists of: one of the keywords PDS, SEQ, or PASS; an equal sign; and a data set name enclosed in apostrophes. The parameters are separated by commas and must not contain any blanks. These procedure parameters -

//BEWARE EXEC VIPER, PDS='d.s.n', SEQ='d.s.n', PASS='d.s'
may be used to provide values for the DSN parameters on
three of the built-in DD cards described below.

The DD cards in the VIPER procedure are in
alphabetical order, to avoid confusion when override cards
are required. However, they will be described below in a
more logical order.

A "//STEPLIB DD DISP=SHR,DSN=U.PPL.BINARY" card is
provided to indicate where the executable version of VIPER
is to be found. A "//SYSUDUMP DD SYSOUT=A" card is not
included so that VIPERabend dumps will not be printed.

The VIPER (and EDIT) output is written by the FORTRAN
I/O routines onto unit 6, so a "//FT06F001 DD SYSOUT=A"
card is provided. The command input file is assumed to be
defined with the ddname=VIPERIN, so a
 "//VIPERIN DD DDNAME=SYSIN" card is provided. This
redirects the command input to the ddname=SYSIN. Thus the
input data may be provided either as cards "//SYSIN DD *"
(or "//SYSIN DD DATA" if the data includes "//" cards) or
by specification of a data set on the SYSIN DD card.

Two DD cards useful for card output are provided.
The first is "//CARDS DD SYSOUT=B" which defines
ddname=CARDS to be a physical system punch. Records which
are routed to CARDS, e.g. by a "PUNCH CARDS" or
"COPY SEQ CARDS" command, should turn up as real

punched cards. An output option card `"//*FORMAT PU,DDNAME=CARDS,DEST=?"` may be used to route these to a desired punch unit.

The `"/PASS DD DISP=(MOD,PASS),DSN=&PASS"` card defines a temporary sequential card image file which may be used to pass cards to subsequent job steps. The "MOD" disposition is required to ensure that multiple decks routed to PASS do not simply overlay each other at the front of the file. This is an important detail which should be followed on any user-defined output DD cards. The name of the PASS file will default to `DSN=&&PASS`, unless another name is provided with the `PASS=` option on the EXEC VIPER card.

Six temporary utility files are provided, with ddnames of the form `VIPERUTn`. The EDIT command uses `VIPERUT1` and occasionally a member named `VIPERUT1` in the PDS-mode data set. The CLICHE command uses `VIPERUT2` to store the `**CREATE` decks as members. `VIPERUT3` and `4` are used for scratch by `IEBCOPY` in the `COMPRESS` command; `"VIPERUT6 DD SYSOUT=A"` is the `COMPRESS` output, and `"VIPERUT5 DD DUMMY"` is the input. (This input file could be overridden by a daring user to obtain other than a basic total copy operation.)

Two DD cards are provided to allow catalogued data sets to be made available easily on the EXEC card: names provided with the optional `PDS=` and `SEQ=` parameters are substituted on the cards `"/PDS DD DISP=OLD,DSN=&PDS"` and

"//SEQ DD DISP=OLD,DSN=&SEQ". The ddnames PDS and SEQ can then be used within VIPER, e.g. on the "DIRECTORY PDS" command, to access these data sets. If the files are not catalogued, then override DD cards must be used to supply the UNIT and VOL data required to locate them.

Additional files are made available to VIPER by adding DD cards with unique ddnames after the EXEC VIPER card and any override DD cards. Such files may have any format and blocking (RECFM and BLKSIZE) but should specify a logical record length (LRECL) of 80. (Except that the DIRECTORY and COMPRESS commands are valid for any partitioned data set.)

The minimum amount of core required to execute VIPER has not yet been established. With no COMPRESS operations, less than 100K seems sufficient; 150K has been enough to allow IEBCOPY to be loaded for the COMPRESS command.

Credits

The command processors of VIPER and EDIT are incorporated into two monstrous FORTRAN programs, compiled with the IBM Level-H compiler. The size and structure of these programs is a matter of expediency; great improvements could be achieved with a top-to-bottom rewrite in a higher level language (PL/I).

VIPER is a descendent of MANAGERY, a program for the manipulation of partitioned data sets. MANAGERY was itself an adaptation of the MANAGER program originated (?) by J.R. Comfort of the Princeton University Physics Department.

The EDIT subsystem was written so that VIPER could use the features available in the EDITY program, with a number of improvements. EDITY, a program for editing sequential files, was written by N. Winsor at the Princeton Plasma Physics Laboratory.

Special input and output services are provided through Fortran entry points in two Assembler-coded I/O packages. The FTNIOPDS routines, developed by K. Rich at the National Accelerator Laboratory, are used principally for directory manipulation. The SPAM routines, written by H.H. Cobb at the Princeton University Computer Center, perform most of the VIPER and EDIT I/O.

Acknowledgment

This work was supported by Energy Research and Development Administration (formerly Atomic Energy Commission) contract E(11-1)-3073; use was made of computer facilities supported in part by the National Science Foundation Grant NSF-GP579.

Example 1

Preparation and compilation of a Fortran program.

```
// *PROCLIB=U.RS.PROCLIB
// *
// EXAMPLE EXEC VIPER,PDS='U.FORTRAN.SOURCE'
// SYSIN DD *
@PDSNAME PDS
@CREATE STAR
        DUST=0.0
        CALL SHIP(DUST)
        STOP
        END
@LABEL STAR STAR
@CREATE SHIP
        SUBROUTINE SHIP(DUST)
        DUST=1.0/DUST
        RETURN
@EDIT SHIP
,A 3
END
@LIST
@COMPRESS
@DIRECTORY
@PUNCH PASS
/*
//FORTRAN EXEC FORTHCLG
//FORT.SYSIN DD DISP=(OLD,DELETE),DSN=&&PASS
```

Example 2

A silly demonstration of some other commands.

```
//*PROCLIB=U.RS.PROCLIB
//*
//EXAMPLE EXEC VIPER,PDS='U.SILLY.EXAMPLE'
//SYSIN DD *
@PDSNAME PDS
@CREATE V
  VERY
@CREATE R
  ROUTINE
@REPLACE I
  INGENIOUS ?
  OBSCURE
@REPLACE E
%MERGE V
%MERGE I
  EDITING
%MERGE R
@EDIT E VIPER
@LIST VIPER
@CREATE P
  PRICE'S
@MERGE ON PDS
@EDIT I
.S 1 '?'!!
.D 2
.G 2 P
@EDIT VIPER
.PRINT OFF
.B 1
  VIPER:
@LIST
/*
```

Summary of VIPER Commands

<u>Code</u>	<u>Mode</u>	<u>Parameter</u>	<u>Action</u>
<u>PDSNAME</u>	either	ddname	sets PDS mode & ddname
<u>SEQNAME</u>	either	(ddname)	sets SEQ mode (initial default)
<u>CREATE</u>	PDS	memb. (DD)	makes memb. from cards or DD
	SEQ	DD1 (DD2)	copies to DD1 from cards or DD2
<u>REPLACE</u>	PDS	memb. (DD)	replace memb. from cards or DD
	SEQ	DD1 (DD2)	copies to DD1 from cards or DD2
<u>COPY</u>	PDS	DD (memb.)	copy memb. or all from DD to PDS
	SEQ	DD1 DD2	copy DD1 to DD2
<u>RENAME</u>	PDS	memb1 memb2	changes name from 1 to 2
<u>DELETE</u>	PDS	memb.	deletes member
<u>DIRECTORY</u>	either	(DD)	sets PDS mode, directory of DD or PDS
<u>COMPRESS</u>	PDS	(DD1 (DD2))	compress 1 into 2, default PDS
<u>LABEL</u>	PDS	label (memb.)	label & seq.# for memb. or all
	SEQ	label DD	label & seq.# for DD
<u>ALTER</u>	PDS	memb. number	puts next card at line # in memb.
	SEQ	DD number	puts next card at line # in DD
<u>EDIT</u>	PDS	memb1 (memb2)	edits 1 into self or 2
	SEQ	DD1 (DD2)	edits 1 into self or 2
<u>CLICHE</u>	either	DD1 DD2	performs Cliche from 1 to 2
<u>MERGE</u>	either	ON/OFF DD	allows/suppresses Merge [†] from DD
<u>LIST</u>	PDS	(memb)	lists memb. or all
	SEQ	DD	lists DD
<u>FLIST</u>	PDS	(memb)	lists memb. or all, eject on END
	SEQ	DD	lists DD, eject on END
<u>EXCLUDE</u>	either	one char	excludes "char" from punch-all
<u>PUNCH</u>	PDS	DD (memb.)	copies to DD, memb. or all (but X)
<u>ABEND</u>	either	ON/OFF	allows/suppresses Abend on error
/*(e-o-f)	either	-	normal termination
blanko (no command)			comment

[†]if MERGE ON, then "%MERGE seqdd/memb." card is replaced by memb. from DD, or seqdd.

Summary of EDIT Commands

<u>Code</u>	<u>Parameter</u>	<u>Action</u>
<u>AFTER</u>	m	put following cards after #m
<u>BEFORE</u>	m	put following cards before #m
<u>CHANGE</u>	m (n)	put following cards in place of #m thru n
<u>DELETE</u>	m (n)	delete #m thru n, then add following cards
<u>ELECT</u>	m (n)	delete to #m, then take #m thru n, then add following cards
<u>FIND</u>	* m 'string'	look for "string" starting in col.m (or eof if *) then backspace & set counter (for m) to 0, then add following cards
<u>GET</u>	m name	after #m, copy from memb. name or seq. name, then add following cards
<u>HALT</u>	(m)	copy thru #m or eof, then stop
<u>LABEL</u>	<u>ON/OFF</u>	if ON, use 72 col. & put label; if OFF use 80 col.
<u>PRINT</u>	<u>ON/OFF/CHANGES</u>	controls print
<u>SWAP</u>	* m 'old string' new string	replaces old string by new on line #m, or all lines if *, then add following cards
/* (e-o-f) or VIPER command		copy thru eof, then stop
blank (no command)		comment