



October 19, 1995

In order to provide something useful to the Federal and State Government M  
community, I am making this document available for free copying and use by  
U.S. Federal and State Government Agencies.

Enjoy!

*Gregory R Kreis*

Greg Kreis



# VPE User Guide

**Gregory R. Kreis**  
**Paragon Training, Inc.**

3783 Upland Dr.  
Marietta, GA 30066

770-509-7022

770-509-7329 FAX

Greg\_Kreis@atlmug.org

kreis.greg@forum.va.gov

This manual is copyrighted by Paragon Training, Inc  
with all rights reserved. Under the copyright laws, this manual  
may not be copied, in whole or in part, without the written consent of  
Paragon Training, Inc. Such exception does not allow copies  
to be made for others, whether or not sold.

© 1991-1995 Paragon Training, Inc



# TABLE OF CONTENTS

---

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1	CLASS GOALS AND OBJECTIVES.....	1
1.2	PREREQUISITES.....	1
1.3	CONVENTIONS.....	1
<b>2</b>	<b>WHAT IS THE VICTORY PROGRAMMER ENVIRONMENT?.....</b>	<b>4</b>
2.1	OVERVIEW.....	4
2.2	ORGANIZATION.....	4
2.3	WHO SHOULD USE VPE?.....	6
2.4	ADAPTABLE TO THE ENVIRONMENT.....	6
<b>3</b>	<b>VSHELL - IMPROVED PROGRAMMER MODE.....</b>	<b>7</b>
3.1	CONCEPT.....	7
3.2	COMMON FEATURES: THE VKERNEL.....	7
3.2.1	CLH: COMMAND LINE HISTORY	7
3.2.2	STANDARD KEYS	7
3.2.3	SCROLLING LISTS	7
3.2.4	USER ID	8
3.3	QWIKS.....	9
3.3.1	CONCEPT9	
3.3.2	USER QWIKS	9
3.3.3	SYSTEM QWIKS	13
3.3.4	VENDOR SPECIFIC CODE	16
3.3.5	ADVANCED QWIK TOPICS	17
3.4	MISCELLANEOUS FEATURES.....	20
3.4.1	MONITORING FOR GLOBAL KILLS	20
3.4.2	USER PARAMETERS	20
3.4.3	TIME OUT PROCESSING	21
3.4.4	ONLINE HELP	21
<b>4</b>	<b>MUMPS TOOLS.....</b>	<b>22</b>
4.1	VGL - GLOBAL LISTER.....	22
4.1.1	SHORTCUTS	23
4.1.2	NODE AND SUBSCRIPT EDITING	24
4.1.3	FILEMAN AWARE	25
4.2	VRR - ROUTINE READER.....	27
4.3	E - ROUTINE EDITOR.....	28

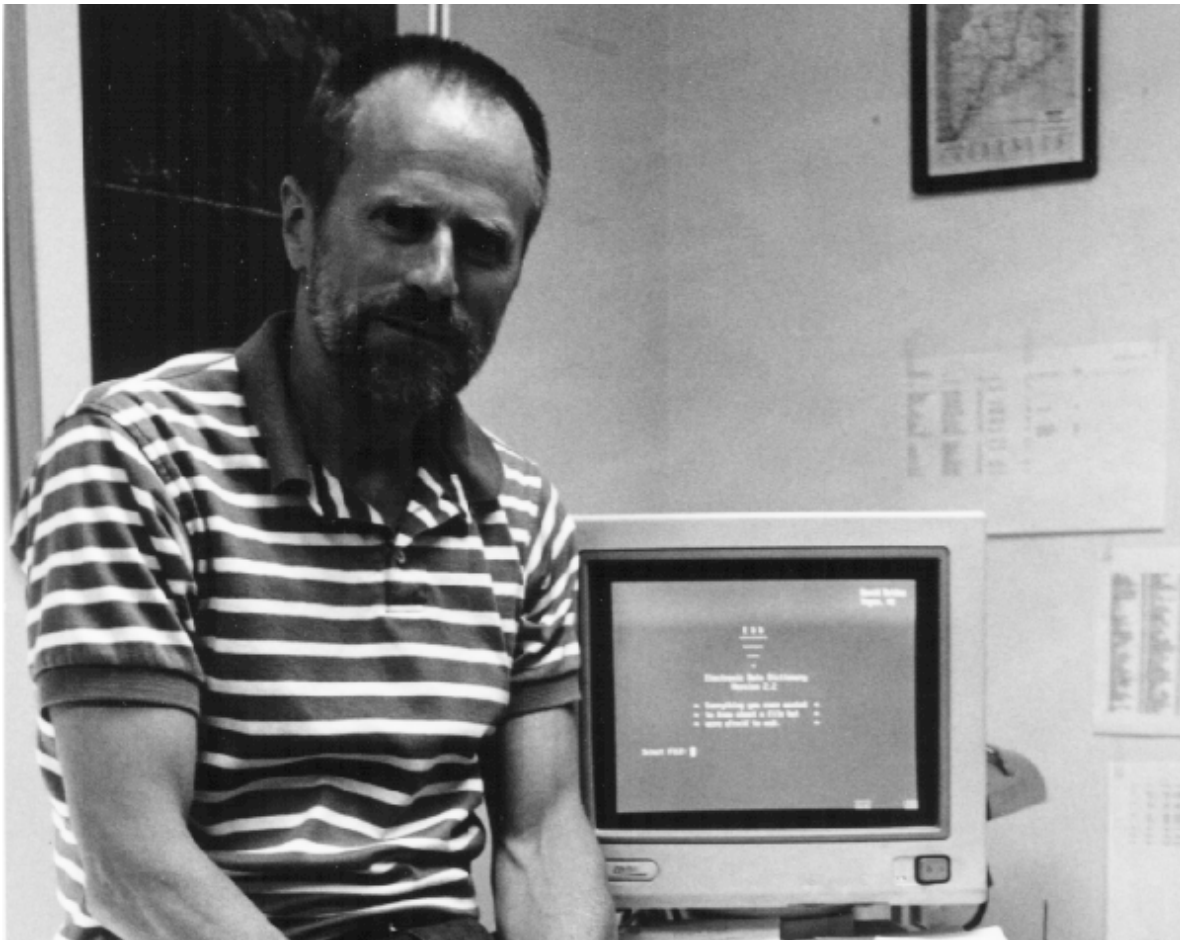
## TABLE OF CONTENTS

---

4.4	COMMON ROUTINE READER AND EDITOR FEATURES.....	30
<b>5</b>	<b>FILEMAN TOOLS.....</b>	<b>31</b>
5.1	VEDD - ELECTRONIC DATA DICTIONARY.....	31
5.2	FIELD GLOBAL LOCATIONS.....	33
5.3	FILEMAN UTILITIES - HELP TEXT.....	35
5.4	TEMPLATE DISPLAYS.....	35
5.4.1	SORT	35
5.4.2	PRINT	36
5.4.3	INPUT	36
<b>6</b>	<b>KERNEL TOOLS.....</b>	<b>37</b>
6.1	HELP TEXT FOR KERNEL OPTIONS.....	37
6.2	RUNNING VPE AS A KERNEL OPTION.....	37
<b>7</b>	<b>MISCELLANEOUS TOOLS.....</b>	<b>38</b>
7.1	CALENDAR.....	38
7.2	IDENTIFYING KEY SEQUENCES.....	38
7.3	ASCII TABLE.....	39
<b>8</b>	<b>SUMMARY.....</b>	<b>40</b>

## *Special Thanks*

*The MUMPS community owes a special thanks to David Bolduc who authored the VPE software. Dave is a dedicated, hard-working programmer who has graciously created and released this powerful software in the hopes that it is useful to others.*



*You may contact Dave at his electronic mail address:*

**BOLDUC. DAVID@FORUM.VA.GOV**

*If you like the software and use it, send Dave a postcard! All programmer's like to hear their efforts are appreciated.*





## 1 INTRODUCTION

Welcome to a *VPE USer Guide*. You will be learning about tools that will make your job easier and more enjoyable. You will also learn how to extend these tools to make a very powerful M programming environment custom tuned to your needs.

### 1.1 CLASS GOALS AND OBJECTIVES

- Provide background of the M Programmer environment
- Introduce the Victory Programmer Environment (VPE)
- Reveal the benefits of the VPE in overcoming typical programmer mode limitations
- Review some of the typical uses for the VPE software

### 1.2 PREREQUISITES

Students should have fundamental knowledge of MUMPS and programmer mode to receive maximum benefit from the class. Portions of the class will mention VA File Manager, so knowledge of FileMan will be beneficial, but not essential.

### 1.3 CONVENTIONS

Terminal sessions and computer listings of globals and routines are found throughout this handout. To make them easier to distinguish, they are displayed enclosed in a box in the Courier font, as shown below. Sometimes, in the interest of saving space on a page, the dialogues will be shortened. These changes will be cosmetic only, leaving the content intact.

```
PUT(REC)           ;deposit line in carrier message
  I $G(REC)'="" S LCNT=LCNT+1,^XMB(3.9,XMZ,2,LCNT,0)=REC
  Q
```

Terminal sessions distinguish the user input by double underlining as shown below.

```
Select OPTION: ENTER OR EDIT FILE ENTRIES
```

Some of the dialogue boxes contain global listings. They were created by one of the tools you will be studying. The form of the listings, shown below, is followed by an actual listing.

```
ref.#) global node = data value
```

Here is a portion of a global in the format mentioned above. Notice how the reference numbers, the far left, make it easier to refer to a node in the listing. When a node is too long to be displayed on one line it wraps around and the equal sign is repeated.

```
1) ^VECJ(19050.1,0) = INDIVIDUAL^19050.1I^24^23
2) ^VECJ(19050.1,1,0) = NASIUM,JIM^4
3) ^VECJ(19050.1,1,1) = 123 WESTOVER RD^APT A^MAIN^13^34567
4) ^VECJ(19050.1,1,2,0) = ^19050.16A^2^2
5) ^VECJ(19050.1,1,2,1,0) = 555-1234
6) ^VECJ(19050.1,1,2,2,0) = 555-1235 MODEM
7) ^VECJ(19050.1,1,2,"B","555-1234",1) =
8) ^VECJ(19050.1,1,2,"B","555-1235 MODEM",2) =
9) ^VECJ(19050.1,1,3,0) = ^^1^1^2900802^^^
10) ^VECJ(19050.1,1,3,1,0) = Loves to stay after school and play any ball game
    = .
```

To make it easy to see non-displayable keys that must be typed, the following symbols will be used in the materials to represent them.

SYMBOL	KEY
<PF1>	F1 function key
<PF2>	F2 function key
<PF3>	F3function key
<PF4>	F4 function key
<TAB>	TAB
<AU>	Arrow Up
<AD>	Arrow Down
<AL>	Arrow Left
<AR>	Arrow Right
<SP>	Space bar
<ESC>	Escape

This page intentionally left blank

## 2 WHAT IS THE VICTORY PROGRAMMER ENVIRONMENT?

### 2.1 OVERVIEW

Before you can understand the reason for the Victory Programmer Environment (VPE), you need to consider a little MUMPS history. The M standard defines the commands, functions, special variables and other characteristics of the language. It does not, however, define a broad, standard programmer's environment (this situation is so loose that the commands to load, edit and save a routine were never even standardized). As a result, over the years, this part of M has been very open to interpretation. While some M vendors provide much richer programming environments than others, what is missing in every case is a standard.

For example, in some implementations, you invoke a global lister by entering %G, in others, %GL, etc. Some implementations offer full screen editors, while others provide only the most basic of routine editors. Nearly every vendor provides their own way of displaying system status, routine and global directories, etc.

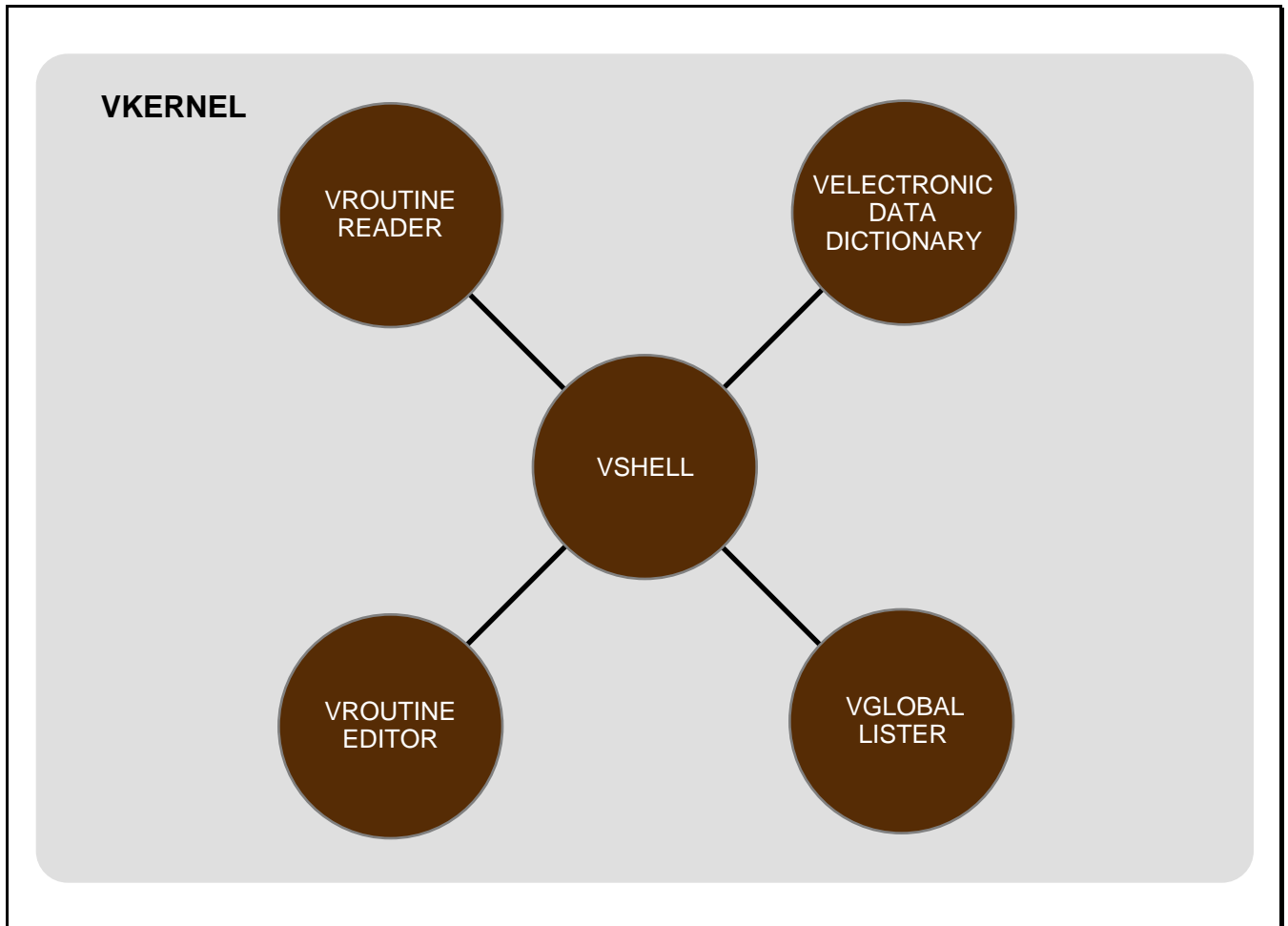
The VPE is much more than traditional programmer mode. While adding features it attempts to standardize the programmer environment by creating it with standard MUMPS code, whenever possible. When vendor specific features are needed, they are hidden and every attempt is made to duplicate the functionality in each implementation.

### 2.2 ORGANIZATION

The VPE provides a rich environment for those who work in programmer mode (programmers, verifiers, support staff, etc). The modules of the VPE are:

MODULE	NAMESPACE	CONJURING
VGlobal Lister/Editor	^%ZVEMG*	DO ^%ZVEMG
VRoutine Reader	^%ZVEMR*	DO ^%ZVEMR
VRoutine Editor		X ^%ZVEMS ( "E" )
VElectronic Data Dictionary	^%ZVEMD*	DO ^%ZVEMD
VShell	^%ZVEMS*	X ^%ZVEMS
VKernel	^%ZVEMK*	

The following drawing shows how these modules are related to one another. By taking the time to study and memorize this diagram, you will be better able to understand the entire environment provided by the VPE software.



As the above diagram shows, the VShell is the common way to access all of the other VPE modules. (VKernel is not shown on the drawing because it is common to all VPE modules.)

### 2.3 WHO SHOULD USE VPE?

Contrary to its name, VPE is not only for programmers. Many people perform technical tasks from programmer mode because it is so powerful. Think of VPE as *enhanced* programmer mode and then you will see that it is ideal for anyone who regularly operates in programmer mode.

VPE is effective if you seldom use programmer mode or if you live in programmer mode! How can this be true? Because the person who occasionally uses programmer mode has a hard time remembering the names of the various system routines. VPE makes them readily available and consistent across implementations. The person who lives in programmer mode will appreciate the keystroke savings and the ability to customize the Shell to their liking. So, wherever you fall in this spectrum, VPE can be useful to you.

### 2.4 ADAPTABLE TO THE ENVIRONMENT

The VPE routines are designed to function in the presence or absence of either the VA FileManager or the VA Kernel software. In some situations, the presence of the Kernel or FileManager makes the VPE software easier to use.

---

### 3 VShell - IMPROVED PROGRAMMER MODE

#### 3.1 CONCEPT

The VShell is a series of MUMPS routines and globals nodes that are used to simulate advanced programmer mode. By creating this environment using MUMPS code, it is much easier to add new features.

#### 3.2 COMMON FEATURES: THE VKERNEL

To make the VPE more robust and easy to learn, the VPE modules share a common set of features.

##### 3.2.1 CLH: COMMAND LINE HISTORY

The command line history remembers the lines you recently typed. This makes it very easy to pull up a previous entry and reuse it without having to type it in again. This is a tremendous time saver. When you realize that you can pull up and edit previous lines you have discovered one of the most beneficial uses of the VPE software.

The command line history is supported in many places, such as:

- VShell prompt - >> prompt
- VGlobal lister prompt - Global ^ prompt
- VRoutine Reader and Editor - Select ROUTINE: prompt
- VElectronic Data Dictionary - FILE: prompt

The command line history supports

##### 3.2.2 STANDARD KEYS

The cursor keys are designed to allow moving back and forth in a line, as well as up or down in lines that wrap around the screen. This is very useful for editing and is a feature supported in many places in the VPE software.

VPE uses the <ESC> key, pressed twice, to terminate the current activity. If you prefer, you may enter the ^ character (the caret) to terminate.

##### 3.2.3 SCROLLING LISTS

In many places, VPE supports the use of the arrow keys to scroll in a list. Pressing return when in these lists will take you to the next page, but the ability to scroll forward and backward gives you much finer control.

Learn these keys and their actions and your efforts will be rewarded

### 3.2.4 USER ID

In order to control access to applications and to allow each person to customize the VPE environment to their personal tastes, each user must have an ID.

When entering the Shell for the first time, you will be asked to create a user ID. When you leave the Shell and come back into it, you will be asked for your User ID each time, unless you are running in a VA Kernel environment. The VA Kernel software keeps track of a user ID for the Kernel and the VShell piggybacks on this by establishing a table of Kernel User IDs that are tied to VShell User IDs.

U S E R   L I S T   ( B Y   T Y P E )				
TYPE	ID	NAME	UCI	DUZ
-----	-----	-----	-----	-----
1. VA KERNEL:	124	KREIS,GREG	PRD,AAA	124
2. QWIK:	124			
3. PARAM:	124			



### 3.3 QWIKS

#### 3.3.1 CONCEPT

A QWIK is the VShell name for a single line of M code. Think of a QWIK as a macro. These macros, which come in two varieties, reduce the number of keystrokes required to activate the code they contain. Those created by you are called User QWIKs and those that come with the VShell are called System QWIKS.

- A User QWIK is invoked by preceding the name with a single period
- A System QWIK. is called with two periods in front of the name.

#### 3.3.2 USER QWIKS

You create your own QWIKs to customize your VPE environment. User QWIKs are stored under each user's ID to avoid naming conflicts. This makes it very important for VPE to remember and recognize your User ID. Without proper identification, you will not be able to access your QWIKS.

##### 3.3.2.1 CREATING QWIKS

In order for a QWIK to be of maximum benefit, it must be easily created, changed and deleted. To this end, creating a QWIK is easy. Just press <TAB> to be asked for a QWIK name. If you type in a new name it will create a new QWIK, otherwise it will find the existing QWIK and present it for editing. Review the following example.

```
PRD,AAA>><TAB>

*** Add/Edit QWIK Command ***

Enter QWIK: EXAMPLE

ADD/EDIT USER QWIK
-----
1) NAME:  EXAMPLE
2) CODE:
3) DESC:
4) PARAM:
5) BOX:
-----
<ESC><ESC>=Quit  <TAB>=Restart  <ESC>H=Help  <ESC>1-5=Field  <ESC>U=UNsav

Edit CODE: W !,"The example code is being executed."
Edit DESCRIPTION: Sample of a simple QWIK
Edit PARAM NOTES:
Edit BOX:
```

### 3.3.2.2 EDITING QWIKS

In the following dialogue, the EXAMPLE QWIK is edited. First, the <TAB> key was pressed and the name of the desired QWIK was entered. Five fields were then presented for editing. Using the <AL> key, the cursor was moved into the CODE value and edited to include the word 'now'.

```
PRD,AAA>><TAB>

*** Add/Edit QWIK Command ***

Enter QWIK: EXAMPLE

ADD/EDIT USER QWIK
-----
1) NAME:  EXAMPLE
2) CODE:  W !,"The example code is being executed."
3) DESC:  Sample of a simple QWIK
4) PARAM:
5) BOX:
-----
<ESC><ESC>=Quit  <TAB>=Restart  <ESC>H=Help  <ESC>1-5=Field  <ESC>U=UNsav

Edit NAME: EXAMPLE
Edit CODE: W !,"The example code is being executed now."
Edit DESCRIPTION: Sample of a simple QWIK
Edit PARAM NOTES:
Edit BOX:
```

### 3.3.2.3 DELETING QWIKS

Finally, since QWIKs are so easy to create, why not make them easy to delete so you can readily use QWIKs to meet long or short-term needs. To delete our sample QWIK we take the following steps.

```
>>..QD

*** Delete QWIK Command ***

Enter QWIK: EXAMPLE
Are you sure you want to delete this QWIK? YES NO
Deleted..
```

## 3.3.2.4 LISTING QWIKS

When you have created several QWIKs you may forget some of their names or the MUMPS code that you placed in them. The <PF1>1 and <PF1>2 key combinations display the current user's QWIKs. <PF1>1 lists the QWIKs with name and description, as shown below.

```
PRD,AAA>> <PF1>1
U S E R   Q W I K S   ( .QWIK)   ID: 124   BOX:
=====
1) DOS           True DOS mode
2) E             Old E editor
3) FM           Fileman
4) G             MSM CUA Style Editor
5) IND          %INDEX
6) MACRTN       Read CD-ROM text files from Mac and White Knight
7) MRTN         Mass load of routines over serial lines
8) PDS          Pioneer FileMan Options
```

The <PF1>2 key generates the list of QWIKs, but substitutes the MUMPS code for descriptions. See below.

```
PRD,AAA>> <PF2>
U S E R   Q W I K S   ( .QWIK)   ID: 124   BOX:
=====
1) DOS           !COMMAND
2) E             X ^%ZVEMS("EOLD")
3) FM           D P^DI
4) G             W:$I'=1 !,"Won't work here... " I $I=1 ZL @%1 X ^%G
5) IND          D ^%INDEX
6) MACRTN       D ^ZZINPUT
7) MRTN         D ^ZZINPUT
8) PDS          D ^VECUCF
```

We will be studying System QWIKs shortly, but it seems appropriate to mention here that <PF1>3 functions like <PF1>1 only the listing is the list of System QWIKs. As you have probably guessed, <PF1>4 lists the System QWIKs with their MUMPS code.

### 3.3.2.5 CONVERTING COMMAND LINE HISTORY TO A QWIK

Have you ever entered a line of code that you wanted to save somehow so you could repeat it at will, even after having logged off and back on? If you have, you probably found that creating a routine works, but it is inefficient to create an entire routine to store a single line of code. You may also find the process of naming, creating and saving the routine to be tedious. One solution is to put many lines of code in a single routine and reference them using line labels. This works, but it requires even more keystrokes to call the code.

Fortunately the VShell provides a solution. When you pull up a line in the command line history, you can simply press <ESC>Q and it will prompt you for the information required to make the code into a user QWIK. The following example should help.

First, we create a line of code that we realize we will need several more times before our problem is resolved.

```
PRD,AAA>>
R !,"REC#",REC S LOOP=0 F  S LOOP=$O(^VE CJ(19050.1,REC,2,LOOP)) Q:'LOOP  W !,$
G(^ (LOOP,X))

REC#1
555-1234
555-1235 MODEM
PRD,AAA>>
```

Next, we call the line back from the Command Line History (CLH) and then press <ESC>Q to create the QWIK using the presently displayed code.

```
PRD,AAA>><AU>
R !,"REC#",REC S LOOP=0 F  S LOOP=$O(^VE CJ(19050.1,REC,2,LOOP)) Q:'LOOP  W !,$
G(^ (LOOP,X))<ESC>Q

Enter QWIK: FIXVE CJ
Do you want to add as a new QWIK? YES// Y
Edit DESCRIPTION: Temporary code to diagnose a bug with phone numbers
Edit PARAM NOTES:
Edit BOX:
PRD,AAA>>
```

### 3.3.3 SYSTEM QWIKS

A system QWIK is one that comes pre-defined with the VShell. You may not create or edit them, but, unlike User QWIKs, you can count on them being present wherever you are using the VShell.

#### 3.3.3.1 PRE-DEFINED

The current list of System QWIKs (v 7.13) is shown in the table below. (The table lists the QWIK name followed by its description.)

ASCII	ASCII Table
CAL	Calendar Display -> %1=Number of Starting Month
CLH	Resequene Command Line History
DIC	Fileman DIC Look-up Template
DOS	DOS Interface
DTMVT	Reset VT-100 in DataTree
E	Routine Editor -> %1=Rtn Name
FMC	Fileman Calls
FMTI	Fileman Input Template Display
FMTF	Fileman Print Template Display
FMTS	Fileman Sort Template Display
G	Global List
GCOPY	Global Copy
GD	Global Directory
GDE	Extended Global Directory
GEDIT	Global Edit
GLB	Global Screen Capture -> %1=Global Reference %2=How many lines before pause
GR	Global Restore
GS	Global Save
GSE	Global Search
GSEL	Global Select
KEY	Display Escape Sequence for any Key
LF	VA KERNEL Library Functions
LOCKTAB	Lock Table
NOTES	VPE Programmer Notes
PARAM	System Parameters
PUR	Purge VShell Temp Storage Area - %ZVEMS("%") -> %1=Number of days to preserve
PURVEDD	Purge Command Line History (VEDD)
PURVGL	Purge Command Line History (VGL)
PURVRR	Purge Command Line History (VRR)
PURVSHL	Purge Command Line History (VShell)
QB	Assign QWIK to Display Box
QC	Copy a QWIK

---

---

QD	Delete a QWIK
QE	Add/Edit a QWIK
QL1	List User QWIKs & Description
QL2	List User QWIKs & Code
QL3	List System QWIKs & Description
QL4	List System QWIKs & Code
QSAVE	Save/Restore User QWIKs
QV	Add Vendor Specific Code
QVL	List Vendor Specific Code
RCHANGE	Routine Change
RCMP	Routine Compare
RCOPY	Routine Copy to Another UCI
RD	Routine Directory
RDEL	Routine Delete
RR	Routine Restore
RS	Routine Save
RSEARCH	Routine Search
RSEL	Routine Select
RSIZE	Routine Size
RTN	Make NEW Routine
UCI	Switch UCI
UL	List Users DUZ/ID
UTIL	Utilities Menu
VEDD	%VElectronic Data Dictionary
VER	VShell Version Information
VGL	%VGlobaL Lister
VRR	%VRoutine Reader
XQH	Help Text for Kernel Menu Options -> %1=Kernel Menu Option
ZD	KILL all local variables whose names start with these let -> %1=letters %2=letters ...
ZP	ZPrint a Routine -> %1=Rtn Name %2=Page Length
ZR	ZRemove 1 to 9 Routines -> %1=Rtn Name %2=Rtn Name ...
ZW	ZWrite Symbol Table -> %1=Starting letter

---

## 3.3.3.2 INVOKING SYSTEM QWIKS

Remember, to invoke a System QWIK, enter two periods and the QWIK name or a part of the name. Look at the following examples.

Below, we enter a portion of a QWIK's name and it presents a list of choices. If a QWIK called V exists, then it will be selected. Avoid creating very short named User QWIKs if you would like to use this feature with your QWIKs.

```
PRD,AAA>>..V

  1. VEDD      %VElectronic Data Dictionary
  2. VER       VShell Version Information
  3. VGL       %VGlobal Lister
  4. VRR       %VRoutine Reader
CHOOSE 1-4: 3

VGL . . . Victory Global Lister . . . . . David Bolduc
<SPACE>=File Name/Number   Global*=List   ?=Help

Session 1...Global ^
```

Below is a quick list of all System QWIKs.

```
PRD,AAA>>..

  S Y S T E M   Q W I K S
-----
ASCII          CAL          CLH          DIC          DOS          DTMVT          E          FMC
FMTI           FMTP         FMTS         G             GCOPY          GD           GDE          GEDIT
GLB            GR           GS           GSE           GSEL          KEY          LF           LOCKTAB
NOTES          PARAM        PUR          PURVEDD       PURVGL        PURVRR       PURVSHL      QB
QC             QD            QE           QL1           QL2           QL3          QL4          QSAVE
QV             QVL          RCHANGE      RCMP          RCOPY         RD           RDEL         RR
RS             RSEARCH      RSEL         RSIZE         RTN           UCI          UL           UTIL
VEDD           VER          VGL          VRR           XQH           ZD           ZP           ZR
ZW
```

Here we enter the entire QWIK's name.

```
PRD,AAA>>..VER

VPE version 7.1.3
```

### 3.3.4 VENDOR SPECIFIC CODE

System QWIKs, as well as User QWIKs, can be setup to adapt to each MUMPS implementation. This is very useful when the code that a QWIK performs must vary from implementation to implementation.

You may use `..QVL` to view the vendor specific code associated with a QWIK. The example below shows the diverse ways that various MUMPS utility menus are called. The beauty of the UTIL QWIK is that it will determine the proper code to execute, so all you have to remember is UTIL.

If no code is specified for an implementation that you are using, the QWIK will use the default code (shown above the vendor specific code).

```
PRD,AAA>>..QVL

*** List Vendor Specific Code ***

Enter QWIK: UTIL

      D I S P L A Y   V E N D O R   S P E C I F I C   C O D E

QWIK NAME..... UTIL
TYPE..... System
DESCRIPTION.... Utilities Menu

Default Code... W $C(7),!?2,"Not available for this M Vendor.",!

Vendor DSM..... D ^%LIB

Vendor MSM..... D ^%UTL

Vendor DTM..... ZZUTIL

Vendor VAXDSM.. D ^%LIB

Enter QWIK:
PRD,AAA>>
```

You can create your own Vendor Specific User QWIKs by using the 'QV - Add Vendor Specific Code' System QWIK (remember, type `..QV`).



### 3.3.5 ADVANCED QWIK TOPICS

Once you have learned how to create a User QWIK, you are ready to move onto more advanced topics. These include ways to pass information to a QWIK, grouping of related QWIKs and preparing QWIKs for shipment to other locations running the VPE software.

#### 3.3.5.1 PARAMETER PASSING

QWIKs can be called with values in an attempt to avoid further prompting. Parameters are additional values placed on the line with the QWIK call. The values must be separated by a space. If a value contains a space, then it must be entered in quotes.

The values passed to the QWIK are stored in the variables %1, %2, %3 up to %9. These variables always exist (they are set to null if no value is passed) so the QWIK code does not need to use \$G or \$D when sampling the parameters.

```
ZW          ZWrite Symbol Table
           -> %1=Starting letter
```

Here is an example of a parameterized call to the ZW QWIK shown above. Notice the documentation of the %1 variable (parameter).

```
PRD,AAA>>..ZW IO

=====
                        S Y M B O L   T A B L E
=====
 1. IO.....: 64
 2. IO(0)....: 64
 3. IO(1,64)..:
 4. IOBS.....: $C(8)
 5. IOF.....: #,$C(27,91,50,74,27,91,72)
 6. IOHG.....:
.....
.....
17. POP.....: 0
18. U.....: ^
19. X.....: T
20. Y.....: 2950603
=====
<>  <ESC>H=ScrollHelp  F=Find  L=Locate
```

Now, let's look at the ZW code that references the parameters.

```
D WRITE^%ZVEMSPS(%1)
```

## 3.3.5.2 GROUPING QWIKS IN BOXES

When your list of User QWIKs gets very long, it may be worth the time to group related QWIKs in 'Boxes'. You may create as many boxes as you like. The only restriction is that the box name must be a whole number.

Boxes can be displayed by entering a period followed by the box number for User QWIKs and two periods to view System QWIKs. See the dialogue box below.

```
PRD,AAA>>..1
```

```
PF3   S Y S T E M   Q W I K S   (..QWIK)   ID: 124   BOX: 1
-----
1) QB      Assign QWIK to Display Box
2) QC      Copy a QWIK
3) QD      Delete a QWIK
4) QE      Add/Edit a QWIK
5) QL1     List User QWIKs & Description
6) QL2     List User QWIKs & Code
7) QL3     List System QWIKs & Description
8) QL4     List System QWIKs & Code
9) QL5     List User QWIKs & Box Location
10) QSAVE  Save/Restore User QWIKs
11) QV     Add Vendor Specific Code
12) QVL    List Vendor Specific Code
```

```
PRD,AAA>>
```

To put a QWIK in a box, press <TAB> at the >> prompt and then edit the BOX field after selecting the desired QWIK.

```
PRD,AAA>>
```

```
*** Add/Edit QWIK Command ***
```

```
Enter QWIK: G
```

```
ADD/EDIT USER QWIK
```

```
-----
1) NAME:   G
2) CODE:   W:$I'=1 !,"Won't work here... " I $I=1 ZL @%1 X ^%G
3) DESC:   MSM CUA Style Editor
4) PARAM:
5) BOX:
```

```
-----
<ESC><ESC>=Quit  <TAB>=Restart  <ESC>H=Help  <ESC>1-5=Field  <ESC>U=UNsav
```

```
Edit NAME: G
```

```
Edit BOX: 4
```

## 3.3.5.3 I NEED THAT QWIK, QUICK

User QWIKs are the result of your handiwork and you will probably want to have them available wherever you are using programmer mode. The VShell supports the creation of a MUMPS routine that holds your QWIKs so they may be sent and installed at another location.

The QSAVE System QWIK, as shown below, can either copy QWIKs into a selected routine or restore QWIKs from a saved routine.

```
PRD,AAA>>..QSAVE

*** Save/Restore User QWIKs ***

1. Save QWIKs
2. Restore QWIKs

Select NUMBER: 1
Enter ROUTINE: ZZQWIK//

I will save your QWIKs to routine ^ZZQWIK.  Ok? YES//      Please wait..
PRD,AAA>>
```

When restoring QWIKs, you will be asked for a Box and User ID . See the example below.

```
PRD,AAA>>..QSAVE

*** Save/Restore User QWIKs ***

1. Save QWIKs
2. Restore QWIKs

Select NUMBER: 2
Enter ROUTINE: ZZQWIK//
Enter BOX:
Enter ID: 124

ABC CALC DOS EDIT FIXFILE FIXVECJ FM GD GE GLOAD GSAVE IND MM NOTES
PACK PACKREP PFM QWIKQWIK RCOPY RD RLOAD RPRINT RS RSAVE SS TE TMON
TO VENDOR VETL VIEW ZE ZP ZTPP ZUCI ZW

This routine contains the above QWIK(s) which already exist on your
system. If I continue I will overwrite them. You may stop the load
here and edit ^ZZQWIK to change the names.

Should I continue? YES//
```

### 3.4 MISCELLANEOUS FEATURES

#### 3.4.1 MONITORING FOR GLOBAL KILLS

The VShell scans each line of code entered at the >> prompt to see if it contains code to kill a global. If it finds such, it asks if you really wish to proceed since killing certain globals or global nodes could be a costly mistake. The following dialogue illustrates this very useful feature.

```
PRD,AAA>>S GREG=45 K ^GREG W GREG

WARNING: Your code may be killing a global.
Should I execute your code: NO// Y
45
```

The default answer to the question to proceed is set by using the PARAM QWIK.

#### 3.4.2 USER PARAMETERS

VPE allows each programmer to customize some of VPE features. These features are listed in the example below.

```
PRD,AAA>>..PARAM

                S Y S T E M   P A R A M E T E R S
=====
1. Global Kill..... NO                Default answer to global kill warning
2. Prompt..... ACTIVE                 ACTIVE Prompt includes UCI,VOL>>
3. Time-out..... 999999                Shell Time-out length in seconds
4. SAVE Routine..... ZZQWIK            Routine that holds your saved QWIKs
5. <DEL> & <BS> Keys... SAME           <DELETE> different from <BACKSPACE>
=====

Select NUMBER:
```

### 3.4.3 TIME OUT PROCESSING

If you want code to be performed when you time out at the VShell >> prompt, create a User QWIK called TO. This is a special name that VShell looks for when a timeout occurs.

(See the online VPE documentation under Miscellaneous.)

### 3.4.4 ONLINE HELP

VPE has extensive online help text only a ? away. The topics below each present a scrollable list of help text.

```
PRD,AAA>>?  
=
```

#### V S H E L L   H E L P   M E N U

Introduction	Boxes
Protection	Vendor Configurations
Keyboard	Command Line History
VA KERNEL	Programmer Tools
QWIK Commands	Parameter Passing
System QWIKs	Miscellaneous
User QWIKs	Quit

SELECT:

<ESC><ESC>, '^'=Quit      HIGHLIGHT<RET>,Enter Name=Select Menu Option

## 4 MUMPS TOOLS

### 4.1 VGL - GLOBAL LISTER

One of the most common tools used in programmer mode is the global lister. Each MUMPS vendor seems to have their own twist on how to design this tool which makes it hard to master them all. As MUMPS systems are attached to larger and larger networks, the chances of coming across different global listers increase.

The VGlobal Lister (VGL) is an enhanced global lister based on ANSI MUMPS code, so it is available on all MUMPS implementations. The following dialogue illustrates a typical session with the VGL.

```
PRD,AAA>>..VGL

The VGlobal Lister . . . . . David Bolduc
<RETURN>=Quit   <SPACE>=File Name or Number   ?=Help

Session 1...Global ^VECJ(19050.1

=====[Session 1]=====
1) ^VECJ(19050.1,0) = INDIVIDUAL^19050.1I^46^43
2) ^VECJ(19050.1,1,0) = NASIUM,JIM^11
3) ^VECJ(19050.1,1,1) = 3783 UPLAND DR^APT A^MARIETTA^1^34567
4) ^VECJ(19050.1,1,2,0) = ^19050.16A^2^2
5) ^VECJ(19050.1,1,2,1,0) = 555-1234
6) ^VECJ(19050.1,1,2,2,0) = 555-1235 MODEM
7) ^VECJ(19050.1,1,2,"B","555-1234",1) =
8) ^VECJ(19050.1,1,2,"B","555-1235 MODEM",2) =
9) ^VECJ(19050.1,1,3,0) = ^^3^3^2950601^^
10) ^VECJ(19050.1,1,3,1,0) = Loves to stay after school and play any ball gam
    = e.
11) ^VECJ(19050.1,1,3,2,0) =
12) ^VECJ(19050.1,1,3,3,0) = It allows me to edit more than one line?
13) ^VECJ(19050.1,1,4) = 2891220
14) ^VECJ(19050.1,2,0) = SKYWALKER,LUKE^3
15) ^VECJ(19050.1,2,1) = 12 MILKY WAY^^TWILIGHT ZONE
16) ^VECJ(19050.1,2,4) = 2900106
17) ^VECJ(19050.1,3,0) = WILSON,JAMES^3
18) ^VECJ(19050.1,3,1) = 1821^^FAIRBANKS^2^12314
19) ^VECJ(19050.1,3,2,0) = ^19050.16A^1^1
=====
<> 'n'=Pieces  A=Alt  G=Goto  S'n'=Skip  C=CdSrch  ?=Help  M=More...
Select: 3
```

Notice how the above display stops, one screen at a time. To the far left of each node is a reference number to make it easier to select a specific, as was done in this example.

### 4.1.1 SHORTCUTS

A Nice feature of the VGL is command line history. If you wish to pull up an earlier global listing, simply recall the earlier one and edit it. The example below shows pressing the up-arrow cursor key three times to get to the desired value.

```
PRD,AAA>>..VGL

The VGlobal Lister . . . . . David Bolduc
<RETURN>=Quit   <SPACE>=File Name or Number   ?=Help

  Session 1...Global ^<u>AU</u>
^DIC(200<u>AU</u>
^VECJ(19050.1<u>AU</u>
^%ZVEMS
```

There is a convenient way to list the top level notes in a global when using VGL. Just enter the global name followed by an \* when asked for the Global.

```
Session 1...Global ^<u>XMB*

  1. ^XMB(1 . . . . .: KERNEL SITE PARAMETERS
  2. ^XMB(3.51 . . . . .: SPOOL DOCUMENT
  3. ^XMB(3.6 . . . . .: BULLETIN
  4. ^XMB(3.7 . . . . .: MAIL BOX
. . . . .
 13. ^XMB("NUM" . . . . .: 236
 14. ^XMB("PARENT" . . . . .: 48
 15. ^XMB("POST" . . . . .: 2780
 16. ^XMB("TIMEZONE" . . . . .: EDT
```

The global display window is a scrollable list. By pressing the <AD> or <AU> keys, you can move forward or backward in the list.

```
=====[Session 1]=====
 3) ^VECJ(19050.1,0) = INDIVIDUAL^19050.1I^46^43
 4) ^VECJ(19050.1,1,0) = NASIUM,JIM^11
 5) ^VECJ(19050.1,1,1) = 3783 UPLAND DR^APT A^MARIETTA^1^34567
. . . . .
19) ^VECJ(19050.1,3,0) = WILSON,JAMES^3
20) ^VECJ(19050.1,3,1) = 1821^^FAIRBANKS^2^12314
21) ^VECJ(19050.1,3,2,0) = ^19050.16A^1^1
=====
<> 'n'=Pieces A=Alt G=Goto S'n'=Skip C=CdSrch ?=Help M=More...
```

## 4.1.2 NODE AND SUBSCRIPT EDITING

VGL supports the editing of global nodes as well as subscripts. To edit a global node, you enter the command EV (edit value) and to edit the subscript, you enter ES (edit subscript). When editing, the cursor keys are fully functional.

```

=====[Session 1]=====
 3) ^TMP("DDS",5,14,2,37) = 0,
 4) ^TMP("DDS",5,14,2,37,"0," ) = 1
 5) ^TMP("DDS",5,14,2,37,"0","GL") =
 6) ^TMP("DDS",5,14,"F0","0","1,37","D") =
.....
.....
20) ^TMP("DDS",11,14,1,36,"3","GL") = ^VECJ(19050.2,
21) ^TMP("DDS",11,14,2,37) = 0,
22) ^TMP("DDS",11,14,2,37,"0," ) = 1
=====
<> 'n'=Pieces  A=Alt  G=Goto  S'n'=Skip  C=CdSrch  ?=Help  M=More...
Select: EV
Enter REF NUMBER: 20

```

You are presented with the following display and can cursor into the node's value and edit it.

```

                                EDIT GLOBAL VALUE

^TMP("DDS",11,14,1,36,"3","GL") = ^VECJ(19050.2,

```

Entering ES instead of EV and selecting a node, you are able to edit its subscripts. (At present, you are not allowed to edit the subscripts of a node that has descendants.)

```

Select: ES
Enter REF NUMBER: 5

                                EDIT GLOBAL SUBSCRIPT

^TMP("DDS",5,14,2,37,"0","GL")
  "DDS",5,14,2,37,"0","GL"

```



## 4.1.3 FILEMAN AWARE

The VGL program is aware of File Manager so it can recognize when it is listing a global that is really a File Manager file. Based on this knowledge, it reverse engineers the global to track back to the relevant DD to provide you with the information you need.

First, select a node from a FM file by using its number displayed to the left of the node.

```
=====[Session 1]=====
 3) ^VECJ(19050.1,0) = INDIVIDUAL^19050.1I^46^43
 4) ^VECJ(19050.1,1,0) = NASIUM,JIM^11
 5) ^VECJ(19050.1,1,1) = 3783 UPLAND DR^APT A^MARIETTA^1^34567
.....
<> 'n'=Pieces A=Alt G=Goto S'n'=Skip C=CdSrch ?=Help M=More...
Select: 5
```

The initial screen display will show data in its internal format, as shown below. Note the STATE field's value.

```
3) ^VECJ(19050.1,1,1) [INTERNAL VALUE]
=====
STREET ADDRESS 1..... 1.) 3783 UPLAND DR
STREET ADDRESS 2..... 2.) APT A
CITY..... 3.) MARIETTA
STATE..... 4p) 1
ZIP..... 5.) 34567
<> <> <>
.....
<> 'n'=FldDD I=IntVal X=ExtVal ?=Help <ESCH>=ScrollHelp
Select: X
```

By selectin g X, as shown above, the display changes to the one shown below. Note that the state pointer has been evaluated.

```
3) ^VECJ(19050.1,1,1) [EXTERNAL VALUE]
=====
STREET ADDRESS 1..... 1.) 3783 UPLAND DR
STREET ADDRESS 2..... 2.) APT A
CITY..... 3.) MARIETTA
STATE..... 4p) ALABAMA
ZIP..... 5.) 34567
<> <> <>
.....
.....
<> 'n'=FldDD I=IntVal X=ExtVal ?=Help <ESCH>=ScrollHelp
```

In the following dialogue box, you see how FileManager influences the VGL global selection process. If you enter a global name of a space, then VGL asks you to look up a file in the File Manager's File of Files. With this information, it can derive the file's global reference. Going a step further, you can enter another space at the Global ^%ZIS(1, // prompt and be allowed to select a batch of records from the selected file, one at a time.

```
PRD,AAA>>..VGL

The VGlobal Lister . . . . . David Bolduc
<RETURN>=Quit   <SPACE>=File Name or Number   ?=Help

  Session 1...Global ^ <SP>
Select FILE: DEVICE
              Global ^%ZIS(1, // <SP>
Select DEVICE NAME: CONSOLE      PC CONSOLE      1      AAA
Select DEVICE NAME:

=====[Session 1]=====
  1) ^%ZIS(1,1,0) = CONSOLE^1^1^1^^^^^AAA^^
  2) ^%ZIS(1,1,1) = PC CONSOLE^^122
  3) ^%ZIS(1,1,91) = 80^#,$C(27,91,50,74,27,91,72)^24^$C(8)
  4) ^%ZIS(1,1,"IOUPAR") =
  5) ^%ZIS(1,1,"SUBTYPE") = 9
  6) ^%ZIS(1,1,"TYPE") = TRM
  7) ^%ZIS(1,1,"XUS") = ^^^^1
  <> <> <>
  .....
  .....
=====
  <> 'n'=Pieces  A=Alt  G=Goto  S'n'=Skip  C=CdSrch  ?=Help  M=More...
  Select:
```

These are just a few of the MANY features of the VGL program. Take the time to explore this module's commands and online documentation by entering ? for help.

## 4.2 VRR - ROUTINE READER

The VPE Routine Reader allows you to open up to four routines at a time. This is beneficial when trying to track the flow of control as one program calls another, etc. You may navigate within the routines using the cursor keys.

Another benefit of the routine reader is access to online FileMan documentation and a table of ASCII characters and their codes. Extensive online help for Routine Reader commands is available by entering a ?.

You can access the VGL Global Lister and the VEDD Electronic Data Dictionary from within the Routine Reader so you do not have to leave the routine listing to view global or DD information. [We will be discussing the VEDD module in the next section.]

```
PRD,AAA>>..VRR DIO2

|=====|=====^[^DIO2      ]=====[1 of 4]===== [Lines: 52 ]=====|
DIO2 ;SFISC/GFT,TKW-PRINT ;9/20/94 14:11
2    ;;21.0;VA FileMan;;Dec 28, 1994
3    ;Per VHA Directive 10-93-142, this routine should not be modified.
4    S (DISTP,DILCT)=0
XDY I $D(DIBTPGM) D @("EN"_DIBTPGM),ENRLS^DIOZ(+P(DIBTPGM,"^DISZ",2)) Q
6    X DY(DN) G XDY:DN
7    Q
8    ;
SEARCH S DIO=1
SCR S DIO("SCR")=1,DE=0 I '$D(DIS(0)) G OR
11   X DIS(0) Q: '$T G PASS: '$D(DIS(1))
OR S DE=DE+1 I '$D(DIS(DE)) Q
13   X DIS(DE) E G OR
PASS S: '$D(DPQ) DIPASS=1
O F DLP=0:1:DX Q: 'DN X $$($D(DPQ):DX(DLP),1:^UTILITY($J,99,DLP))
16   Q
17   ;
NW !
T I $X,IOT'="MT" W !
20   I '$D(DIOT(2)),DN,$D(IOSL),$$('$D(DIWF):1,$P(DIWF,"B",2):$P(DIWF,"B",
2),1:1)+$Y'<IOSL,$D(^UTILITY($J,1))#2,^(1)?1U1P1E.E X ^(1)
|=====|=====
<> <TAB>=MenuBar <F3>=Block <RET>=Insert <ESC>K=Keybrd <ESC><ESC>=Quit
```

Pressing the <TAB> key will take you to a menu bar at the bottom of the screen providing access to online help (?) and other commands.

### 4.3 E - ROUTINE EDITOR

VPE contains a new, powerful full screen editor that works on any ANSI standard device. This means you don't have to be at a PC console to use this editor. It is available over a dial up connection as well as at the console!

The E Routine Editor is very similar to the VRR Routine Reader in appearance and function. It provides a command line history of last edited routines, as shown below or it can be called with the routine to be edited as a parameter.

```
PRD,AAA>>..E
```

```
Victory Routine Editor . . . . . David Bolduc
^=Quit   <RETURN>=DefaultRtn   ?=Help   ??=RtnList
```

```
Select ROUTINE: DIO2//
DIO2 <AU>
VECJFD <AU>
```

```
|=====|=====^[^DIO2      ]=====[1 of 4]=====[Lines: 52 ]=====|
DIO2 ;SFISC/GFT,TKW-PRINT ;9/20/94 14:11
2      ;;21.0;VA FileMan;;Dec 28, 1994
3      ;Per VHA Directive 10-93-142, this routine should not be modified.
4      S (DISTP,DILCT)=0
      XDY I $D(DIBTPGM) D @("EN"_DIBTPGM),ENRLS^DIOZ(+SP(DIBTPGM,"^DISZ",2)) Q
6      X DY(DN) G XDY:DN
7      Q
.....
.....
      N W !
      T I $X,IOT'="MT" W !
20     I '$D(DIOT(2)),DN,$D(IOSL),$S('$D(DIWF):1,$P(DIWF,"B",2):$P(DIWF,"B",
      2),1:1)+$Y'<IOSL,$D(^UTILITY($J,1))#2,^(1)?1U1P1E.E X ^(1)
|=====|=====|
<> <TAB>=MenuBar <F3>=Block <RET>=Insert <ESC>K=Keybrd <ESC><ESC>=Quit
```

The Routine Reader and Editor share a unique screen display. Line labels are right-justified and unlabeled lines are numbered.

The Routine Editor gives the programmer the chance to customize the meaning of pressing the RETURN key when in the editor.

To get to the menu of choices, press <TAB> when in the editor and enter P at the Select prompt (shown below). The menu dialog explains the two options.

```
|=====|=====|
Select: [P ] <RET>=Quit R=Rtn F=FndTg L=LctStrg G=Goto ?=Help M=More...
```

When you want to add a new line of code to the routine you are editing, you hit <RETURN> to open a new line. By setting the following parameter, you can control how this process works.

1. Open a new line BELOW the current line, regardless of where on the current line the cursor is located.
2. Open a new line ABOVE the current line if the cursor is at the start of the line. Open a new line BELOW the current line if the cursor is at the end of the line. If the cursor is located anywhere else, BREAK the line.

Enter number of your choice: 2//

The Routine Editor supports cut and paste operations within the current routine or by navigating to another routine to get the cutting to be pasted into the edited routine (see the VPE online documentation below).

#### B L O C K M O D E:

In EDIT mode, type <F3> to change to BLOCK mode. The word BLOCK will appear at the upper right of the screen. Use the Up/Down Arrow keys to highlight routine lines. To act on the highlighted lines do:

```
<ESC>C .....Copy lines to the clipboard
<ESC>X .....Cut lines to the clipboard
<DEL> .....Delete lines
<F3> .....Return to EDIT mode without taking action
```

While in BLOCK mode, you may position the cursor and then hit:

```
<F1><AL> ...Highlight all lines from cursor to top of routine
<F1><AR> ...Highlight all lines from cursor to bottom of routine
```

When you've returned to EDIT mode, position the cursor and hit <ESC>V to paste saved lines into the current routine. The new lines will be inserted below the cursor. In MENU BAR mode you can branch to other routines save code to the clipboard, return to the current routine, and paste the code in.

#### 4.4 COMMON ROUTINE READER AND EDITOR FEATURES

The Routine Reader (VRR) and Routine Editor (E) share many features. Two very useful ones are the interactive Routine Lister and Global Lister.

##### ROUTINE LISTER

When reading a routine, you may place the cursor on the ^ in the routine reference and press <ESC>R to see the routine. The screen will clear and show the selected routine. This can be done up to four levels deep. Leaving this routine display (using <ESC><ESC> or ^) returns to the previous routine display.

##### GLOBAL LISTER

A global reference in a routine can be used to specify a global listing. Place the cursor on the ^ of the global reference and press <ESC>G. Nothing will seem to have happened, but be patient. Now, cursor into the global reference to include all literal subscripts (\$J counts as literal) that you want to see and press <ESC>G again. A global listing will appear and when you leave it, it will return to the previous program listing.

## 5 FILEMAN TOOLS

The VPE software provides several tools to make life easier for those who use File Manager.

### 5.1 VEDD - ELECTRONIC DATA DICTIONARY

The File Manager maintains detailed Data Dictionaries, but it only provides access to the extensive information via reports of all fields. Because these listings waste paper and are time consuming to generate, VEDD was conceived and programmed to provide rapid, online access.

The following dialogue shows a brief session with VEDD.

```
PRD,AAA>>..VEDD

Select FILE: OPTION

A.) FILE NAME:----- OPTION
B.) FILE NUMBER:----- 19
C.) NUM OF FLDS:----- 89
D.) DATA GLOBAL:----- ^DIC(19,
E.) TOTAL GLOBAL ENTRIES:-- 928
F.) FILE ACCESS:
    DD_____
    Read____
    Write____
    Delete___
    Laygo____
G.) PRINTING STATUS:-- Off
=====
                        M A I N   M E N U
=====
X  Cross References      G  Fld Global Location      VGL  VGlobal Lister
PI Pointers IN          T  Templates                PR  Printing-On/Off
PO Pointers OUT         D  File Description          H  Help
GR Groups              A  Globals In ASCII Order
TR Trace a Field       C  File Characteristics
I  Indiv Fld Summary   R  Required Fields

SELECT OPTION: I
```

Notice that after selecting the file you are given a large menu of options that cover most aspects of a Data Dictionary. For example, see the next listing which describes a field in great detail: input transform, cross-references, etc.

Select FIELD: 4 TYPE [0;4] [RS]

FIELD NAME: TYPE

FLD NUMBER: 4 FLD TITLE: THE TYPE OF OPTION  
 NODE;PIECE: 0;4 HELP FRAME: TYPE

ACCESS: RD: DEL: WR:

DATA TYPE: Set of Codes  
 Required field  
 A:action  
 E:edit  
 I:inquire  
 ....  
 ....  
 L:limited  
 C:ScreenMan

INPUT TRANSFORM: Q

CROSS REF NAME: AOR  
 TYPE: MUMPS

Node: 1 I X="O"!(X="Q") S ^DIC(19,"AOR",\$P(^DIC(19,DA,0),U,2),DA)  
 =" "

Node: 2 K ^DIC(19,"AOR",\$P(^DIC(19,DA,0),U,2),DA)

This sets the "AOR" cross-reference on the file if the option type is either a protocol or a protocol menu.

CROSS REF NAME: AOH  
 TYPE: MUMPS

Node: 1 D REDO^XQ7

Node: 2 D REDO^XQ7

This cross-reference causes the option to be flagged for updating in the menu trees so that the changes effecting display and/or access to the option are correctly updated.

DESCRIPTION: This field indicates the 'category' or type of option. A print-type option, for instance, will call FileMan to print something.

.....

There are many other options of the VEDD module that we will test in class and that you should test when you get the software running.



## 5.2 FIELD GLOBAL LOCATIONS

Another very useful feature of the VEDD is the display of fields, in field number order, showing their location in the file's data global. From the field display, you can track down the fields on a specific node or use pointer field to navigate to another file's DD display. (The following displays reveal these features).

```
>>..VEDD
Select FILE: 19050.1  INDIVIDUAL
```

```
.....
=====
                                M A I N   M E N U
                                =====
      X  Cross References      G  Fld Global Location      VGL  VGlobal Lister
.....
Select OPTION: G
ALL_FIELDS  STARTING_FIELD  QUIT  EXIT
```

The following display offers several options to explore the DD fields. In the example below, we select a specific global node for the next display.

```
File: INDIVIDUAL                                     Branch: 1
REF  NODE;PIECE      FLD NUM  FIELD NAME
=====
  1  0;1              .01  NAME
  2  1;1              1  STREET ADDRESS 1
  3  1;2              2  STREET ADDRESS 2
  4  1;3              3  CITY
  5  1;4              4  STATE                                     <-Pntr 5
  6  1;5              5  ZIP
  7  2;0              6  PHONE                                     <-Mult 19050.16
  8  -0;1             .01  -PHONE
  9  3;0              7  COMMENTS                                     <-WP 19050.17
 10  -0;1             .01  -COMMENTS
 11  0;2              8  EMPLOYER                                     <-Pntr 19050.2
 12  4;1              9  DATE ENTERED
 13  Computed        50  FIRST LAST NAME
 14  Computed        51  FIRST LAST NAME II
 15  Computed        52  CSZ
  <>  <>  <>
=====
<>  'n',I=FldDD  DA=Data  F=Find  G=Goto  N=Node  P=Pointer  VGL=VGL  ?=Help
Select: N
Select NODE: 1
```

Here we see the fields for the selected node and we have the option to explore any of the fields in great detail.

NODE ; PIECE	FLD NUM	FIELD NAME
1;1	1	STREET ADDRESS 1
1;2	2	STREET ADDRESS 2
1;3	3	CITY
1;4	4	STATE
1;5	5	ZIP

You may now do an 'INDIVIDUAL FIELD DD'  
on the field(s) listed above..

Select FIELD:

The following two displays show how to use a field to navigate to another file's DD using pointer information. (Notice how the Branch information in the upper right hand corner of the display, changes to indicate the new location.)

```
File: INDIVIDUAL                                     Branch: 1
REF  NODE;PIECE      FLD NUM  FIELD NAME
=====
....
  5  1;4              4  STATE                                <-Pntr 5
.....
<>  'n',I=FldDD  DA=Data  F=Find  G=Goto  N=Node  P=Pointer  VGL=VGL  ?=Help
Select: P
Enter REF NUMBER: 5
```

```
File: STATE                                           Branch: 2
REF  NODE;PIECE      FLD NUM  FIELD NAME
=====
  1              .001  NUMBER
  2  0;1          .01  NAME
  3  0;2           1  ABBREVIATION
  4  0;3           2  VA STATE CODE
....
<> <> <>
=====
<>  'n',I=FldDD  DA=Data  F=Find  G=Goto  N=Node  P=Pointer  VGL=VGL  ?=Help
Select:
```

### 5.3 FILEMAN UTILITIES - HELP TEXT

The ..FMC QWIK provides online access to documentation of FileMan utility calls. This will not obsolete the FileMan manuals, but it can help jog the memory of the experienced programmer. (The screen displays for each topic are scrollable lists.)

```
PRD,AAA>>..FMC
                                FILEMAN 20 - CALLABLE ROUTINES

DDS...ScreenMan                DIO2..Intern to extern DATE
DDIOL.Writer                   DIP...Print
DIAC..File access              DIPT..Print/Sort Temp
DIAXU.Extract data             DIPZ..Print Temp Compile
DIB...User controlled edit     DIQ...Data display.DATE conver
DIC...Look-up.Add              DIQ1..Data retrieve
DIC1..Custom look-up.File info *DIR..Reader
DICD..Wait msg                 DIS...Search
DICN..New entry.YES/NO         DIU2..DD delete
DICQ..Look-up display          DIWE..Text edit
DICRW.Required variables       DIWF..Form document
DID...DD list                  DIWP..Word process.
DIE...Edit                     DIWW..Output last line
DIEZ..Input temp compile       %DT...DATE/Time input/convert
*DIFG.Filegrams                %DTC..DATE/Time manipulate
DIK...Delete.Reindex           COMMA.%DTC Number format
DIKZ..Xref compile             %RCR..Array move
DIM...Code validate            Quit

SELECT:
<ESC><ESC>,'^'=Quit           HIGHLIGHT<RET>,Enter Name=Select Menu Option
```

### 5.4 TEMPLATE DISPLAYS

The VPE software comes with QWIKs that make documented calls to the File Manager to display templates online. This is much more convenient than having to use or edit the template in order to see its specifications.

#### 5.4.1 SORT

Here is the online display of a sort template.

```
PRD,AAA>>..FMTS
Select SORT TEMPLATE: VECJ EMPLOYEE ROSTER
                                (Jan 29, 1990@01:41)                File #19050.1
Template: VECJ EMPLOYEE ROSTER                                     Printed: 06/03/95
File: INDIVIDUAL (#19050.1)
Last Used: 04/21/95
-----
SORT BY: ]EMPLOYER;S1// (EMPLOYER not null)
        WITHIN EMPLOYER, SORT BY: @NAME// (NAME not null)
```

### 5.4.2 PRINT

Here is the online display of a print template.

```
PRD,AAA>>..FMTP

Select PRINT TEMPLATE: VECJ EMPLOYEE ROSTER
                        (Jan 29, 1990@09:21)           File #19050.1
Template: VECJ EMPLOYEE ROSTER                        Printed: 06/03/95
File: INDIVIDUAL (#19050.1)
Last Used: 04/21/95
-----
FIRST PRINT FIELD: NAME;S1//
THEN PRINT FIELD: STREET ADDRESS 1;C1;"//
THEN PRINT FIELD: STREET ADDRESS 2;C1;"//
THEN PRINT FIELD: CITY;C1;"//
THEN PRINT FIELD: ", ";X//
THEN PRINT FIELD: STATE:ABBREVIATION;X//
THEN PRINT FIELD: " ";X//
THEN PRINT FIELD: ZIP;X//
THEN PRINT FIELD: COMMENTS;C3;W70//
```

### 5.4.3 INPUT

Here is the online display of a input template. (NOTE: this QWIK calls a FM routine that may not properly display some templates that use navigation to other files).

```
PRD,AAA>>..FMTI

Select INPUT TEMPLATE: VETV BETTER BRANCH      (APR 17, 1992@17:12) FILE #19056.1

Template: VECJ BETTER BRANCH                        Printed: 06/03/95
File: INDIVIDUAL (#19050.1)
Last Used:
-----
FIRST EDIT FIELD: NAME//
THEN EDIT FIELD: STREET ADDRESS 1//
THEN EDIT FIELD: S:X=" "&($S($D(^VECJ(19050.1,D0,1)):$P(^1),U,1,5),1:"")?.") Y
="@1"//
THEN EDIT FIELD: STREET ADDRESS 2//
THEN EDIT FIELD: CITY//
THEN EDIT FIELD: STATE//
THEN EDIT FIELD: ZIP//
THEN EDIT FIELD: @1//
THEN EDIT FIELD: PHONE//
      THEN EDIT FIELD: PHONE//
```

## 6 KERNEL TOOLS

### 6.1 HELP TEXT FOR KERNEL OPTIONS

The `..XQH QWIK` snoops in an option in the `OPTION` file to see if it has help text. If it finds text, it is displayed and then control is returned to the Shell.

```
PRD,AAA>>..XQH
```

```
Select OPTION NAME: XM-NEW-FEATURES
```

```
Features New in Version 7 MailMan
```

```
New Features in MailMan v7
```

```
MailMan 7 is released with Kernel 7 & includes the following enhancements:
```

1. The 'QUERY' command, given at the 'Message Action:' prompt, now has 3 forms: 'Q'=addressed as, 'QD'=full detail (like previous versions) & 'QN' to see network routing.
2. There is a SHARED,MAIL user for public posting of messages.
3. An X.500 directory exists to help address remote recipients.
4. There are utilities for managing LATER'D Mail.
5. MAIL GROUPS may include remote members and Distribution Lists.
6. Copying messages has been simplified.
7. You may assign an AUTOMATIC DELETION DATE to a message in a basket.
8. You may override the Message Action Default.
9. There is a field in the user options to control display of titles.
10. You may assign yourself a text editor from your site's approved list.
11. The WRITE command sends a new message at the message action prompt.
12. You may assign a VAPORIZATION DATE to a message in a mail basket.

```
Select HELP SYSTEM action or <return>:
```

```
PRD,AAA>>
```

### 6.2 RUNNING VPE AS A KERNEL OPTION

You may wish to make VPE a command menu item, so you can easily fall into VPE, from any Kernel menu, and then halt out of VPE back to the Kernel menus. This means you may never go into 'raw' programmer mode ever again! (In actuality, VPE's interaction with vendor debugging and error trapping is not perfected and is one of the main reasons some VPEers still go into programmer mode.)

The name of the option you create **MUST** include the word `VSHELL` in order for this work. A sample Option to run VPE is shown below.

```
NAME: ZZ VSHELL
```

```
MENU TEXT: Vshell
```

```
TYPE: action
```

```
CREATOR: KREIS,GREG
```

```
E ACTION PRESENT: YES
```

```
ENTRY ACTION: X "N XQY0 X ^%ZVEMS"
```

```
TIMESTAMP OF PRIMARY MENU: 55990,1443 UPPERCASE MENU TEXT: VSHELL
```

## 7 MISCELLANEOUS TOOLS

Listed below are just a few of the many tools that come with the VPE software. You are encouraged to explore them all! Some of these tools were written by other programmers (their identities are made known if permitted).

### 7.1 CALENDAR

A desk top calendar can be very handy. This QWIK can only display a calendar, scheduling is not included.

```
PRD,AAA>>..CAL
                                S I X   M O N T H   P L A N N E R                Roger Ackerman
=====
```

MAY 1995							JUN 1995							JUL 1995						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6					1	2	3							1
7	8	9	10	11	12	13	4	5	6	7	8	9	10	2	3	4	5	6	7	8
14	15	16	17	18	19	20	11	12	13	14	15	16	17	9	10	11	12	13	14	15
21	22	23	24	25	26	27	18	19	20	21	22	23	24	16	17	18	19	20	21	22
28	29	30	31				25	26	27	28	29	30		23	24	25	26	27	28	29
														30	31					

AUG 1995							SEP 1995							OCT 1995							
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	
			1	2	3	4	5						1	2	1	2	3	4	5	6	7
6	7	8	9	10	11	12	3	4	5	6	7	8	9	8	9	10	11	12	13	14	
13	14	15	16	17	18	19	10	11	12	13	14	15	16	15	16	17	18	19	20	21	
20	21	22	23	24	25	26	17	18	19	20	21	22	23	22	23	24	25	26	27	28	
27	28	29	30	31			24	25	26	27	28	29	30	29	30	31					

```
=====
```

### 7.2 IDENTIFYING KEY SEQUENCES

When debugging some programs, it is can be helpful to know what characters are being sent by a terminal. Simply summon the ..KEY QWIK and then press different keys to see what they generate.

```
PRD,AAA>>..KEY
K E Y B O A R D   I N T E R P R E T E R

Hit any key:

ASCII:   27
CHAR:

CONTINUE  QUIT
```

## 7.3 ASCII TABLE

Have you ever been frustrated because you could not remember the ASCII code for the letter T? Well, sweat no more. The ..ASCII QWIK very quickly displays a listing of the ASCII character set.

```
PRD,AAA>>..ASCII
```

## A S C I I C H A R A C T E R S E T

Dec	Chr	Dec	Chr	Dec	Chr	Dec	Chr	Dec	Chr	Dec	Chr	Dec	Chr	Dec	Chr
0	NUL	16	DLE	32	SP	48	0	64	@	80	P	96	`	112	p
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	\$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(	56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41	)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[	107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	
13	CR	29	GS	45	-	61	=	77	M	93	]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

## 8 SUMMARY

VPE is a powerful way to standardize your working environment while offering flexible options to the programmer to adapt to their unique needs.

VPE is constantly evolving and needs your error reports as well as your suggestions. In order to share tricks among VPE Users and communicate needs to the developer, join the VPE USERS mail group on FORUM. When a new release is available, notification will be sent to this group with instructions on how to download it.

Happy environments to you!