Chapter 1

Viking Data Entry Overview

Introduction

The Viking Data Entry System, which we call VDE for short, is a powerful application generator for fast-response, error-free keyboard input applications. VDE transforms your computer into an efficient work station to key-enter data rapidly and accurately, without additional special hardware.

VDE offers a complete, versatile set of 4GL tools for use in your most demanding key entry applications. It combines the best features needed for both 'heads up' and 'heads down' data in an integrated data entry system. With VDE, you can distribute the data entry function to the information sources without sacrificing speed and accuracy of traditional 'heads down' systems. Its full-screen approach makes VDE equally effective for 'heads-up' applications.

Installation Instructions

The information you need to install VDE is contained in Appendix B.

Organization of this Manual

This manual is organized into 20 chapters and 2 appendices. Each chapter covers a particular subject, such as painting forms or function keys.

Chapter titles and page numbers appear in the upper outside corner of each page. Page numbers are prefixed with the chapter number. Notice that the next page is numbered 1-2 (chapter 1, page 2).

This overview begins with a definition of some important terms, provides an overview of each module, and then suggests which portions of the manual you should read next. Please read the remainder of this chapter in order to gain an overview of VDE and then proceed to read the sections that are pertinent to your particular needs.

Notes and Cautions

Important notes, cautions and warnings appear beside the pointing hand symbol, ☞, to draw your attention to the information in the box. Be sure you read and understand these messages before moving to the next step.

Important Definitions

The following terms have a very specific meaning in the Viking Data Entry System.

Form The word Form has a special meaning in the context of VDE. It refers to all the information that appears on a terminal's screen. Some synonyms are: template, format and panel. The Viking Forms Development Facility is used to create the Forms. There are two kinds of Forms, Record Forms and Menu Forms. The elements of a Form are Text, Lines and Data Fields.

Text The information that appears on the screen for the benefit of the user is called Text. Text is not part of the output Data Record and it is not necessarily associated with any particular Data Field. End users never enter or change Text.

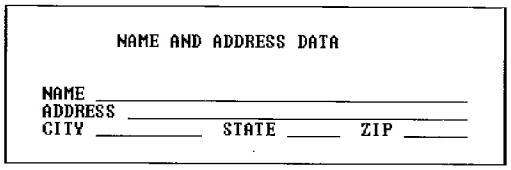
Line Lines are a special kind of Text that use the special graphic features of the monitor or terminal. Lines and boxes are easy to draw and they make the Form more attractive and easier to read.

Data Fields Data Fields are the most important parts of the Form. The end user enters data into the fields. The data keyed into the fields is put in the output area, which may ultimately be a disk Data Record. The user can navigate the cursor from one field to another using the Special Function Keys. The cursor will always be in the Data Fields and will never be in the Text area of the screen. Data Fields have many attributes that control the information that can be keyed into them.

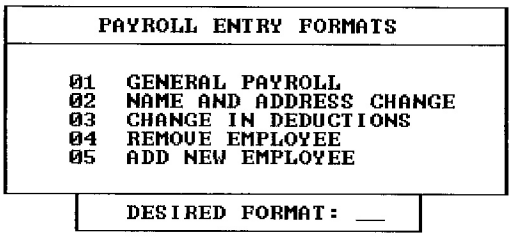
Record Form A Record Form is the most common Form. It is generally, but not necessarily, used to enter data corresponding to a Data Record on a disk or a data structure in a program. A **Record Form Example** appears below.

Scrolling Form A special type of Record Form is a Scrolling Form which is capable of building variable length Data Records when a group of fields is repeated. An example is an invoice with a variable number of line items.

Menu Form A Menu Form looks just like a Record Form, but it has some simplifications. Its primary use is for communication between the user and the program. A series of menus can be used to provide help screens and control selections. A **Menu Form Example** appears below.



Record Form Example



Menu Form Example

Form Set A collection of Menu and Record Forms used for a particular application is called a Form Set. It is produced by the Viking Forms Painter (VFP) and must be given a name consisting of from 1 to 8 characters, alpha or numeric.

Form Set Files The information in a Form Set is actually contained in a special data file called the Form Set File. There are three variations of the Form Set File:

(1) the "Source" Form Set File
(2) a "Loadable" Form Set File, or
(3) an "Object Module" Form Set File.

Character Sets or Sieves Character sets (sometimes called character sieves) are used to determine which characters can be entered into a Data Field.

How Forms Relate to Applications

Forms and Form Sets are the key elements in creating applications using the Viking products. For example, creation of a Form Set is all that is required to generate an application using VDE. The Form Set not only describes how the CRT screens will look and behave, but will also define the data files that will be produced by the end user of the application.

The three simple steps to generate and use a VDE application are as follows.

1. Create a Form Set Source File, using the Viking Forms Painter (VFP)
   a. Create one Record Form for each type of Data Record
   b. Output a Loadable Form Set
2. Execute VDE with the Form Set just created
3. Enter Data Records

When a VDE application is executed, the user sees a Form displayed on the screen and can fill in the blank fields with data. When a field is filled the cursor automatically advances to the next field. All character validations and edit checks are done immediately after a key is pressed.

When an error is detected, VDE displays an error message in the first 24 positions of the reserved area, which is normally the top line. This immediate checking greatly increases the accuracy of the data that is being input and simplifies error correction.

In summary, the Form Set is the key to easy applications using the Viking products. The process of creating the Form Set is very simple because of the Viking Forms Development Facility, which is described below.

Viking Forms Development

The Viking Forms Development Facility consists of two parts; the Viking Forms Painter and the Viking Forms Compiler. The Form Set Source File is created by the Viking Forms Painter (VFP). VFP is analogous to a Text Editor or Word Processor that is used to produce Source Code for programs. Of course, it is much easier to use.

VFP is used to define what your screen Forms will look like on the screen and to specify the attributes of the data fields. VFP is interactive and easy to use.

With VFP you can:

- Create a new Form Set Source File.
- Modify an existing Form Set Source File.
- Combine existing Forms from one or more Form Set Source Files to create a new Form Set Source File.

The second part of Forms development is handled by the Viking Forms Compiler (VFC). VFC takes the Source Code that is generated by VFP and produces a Form Set Object Module, a Loadable Form Set File, a printable Source File, or any combination of these three files. VFC can be automatically invoked when exiting VFP.

Viking Data Entry Program

The Viking Data Entry Program, VDE, is an application generator used to key data into sequential data files on the disk. It is frequently used to replace key-to-disk or keypunch machines. It is also used to create transaction files of data to be loaded into a data base or for processing by other programs.

No programming is required to use VDE. The necessary Forms are created using the Viking Forms Development Facility. VDE does all the rest. When required, customized versions of VDE can easily be created by programmers using the optional Viking product Portal.

Viking Control System

The Viking Control System, VCS, provides a menu interface for non-technical computer users instead of the standard control language used by the computer's operating system. Once a user has logged on, menus with selections of the jobs that the user can run are presented. A job is invoked by simply keying the letter code that identifies the job on the menu. After the job is completed the menu will return to the screen for the next selection.

Jobs may be programs, command files or additional selection menus. VCS builds menus dynamically by comparing the user's privilege profile with the job's requirements profile. This augments security and control provided by the operating system. Audit trails and usage statistics can be accumulated and reported. VCS can be fully integrated with VDE or other programs.

Error Messages

VDE has many error messages that are displayed in the reserved area of the screen to inform the user of errors and communicate other information. Normally, the end-user must clear each message by pressing the [**Reset**] Key before the next character can be keyed. Optionally, the next valid keystroke can clear the error message.

Modes of Use

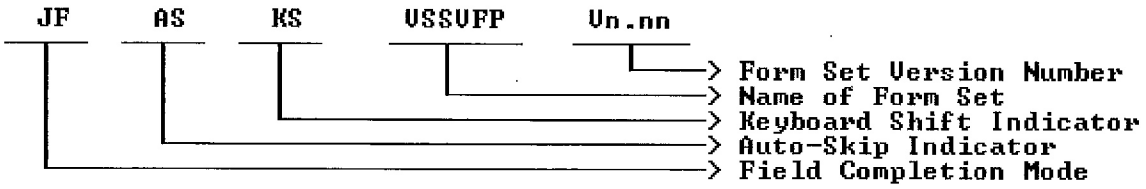
There are various Entry modes for using VDE, Create, Update, Verify and Display Mode. The most common mode is called Create mode which allows you to enter new data for the first time. In Update mode you can modify existing data. In Verify mode existing data is re-keyed and compared to the original data to assure extreme accuracy. Display mode is used for read-only inquiry applications.

The VDE user can select the field completion mode with the [**Mode**] Key at any time. Two modes of field completion are available that affect the way the [**Enter**] key works. In **Justify/Fill** mode, **JF**, when you press the [**Enter**] key, the field is justified and filled if necessary and then the cursor advances to the next field. Characters to the right of the cursor in a field will be lost. Conversely, in the **Edit** modes, pressing [**Enter**] merely results in the cursor advancing to the next field, no justification is performed, and no data is lost as a result.

In Edit mode, there are two sub-modes for replacing characters in existing data. In **Edit/Replace** mode, **ER**, as you key over a character it is simply replaced. If you are in **Edit/Insert** mode, **EI**, the additional characters are inserted and existing characters are moved to the right to make room for the new character. The right-most characters in the field will be lost when they are moved past the end of the field, except in word wrap fields.

The Reserved Area of the Screen

The Form Set name, status information and error messages are displayed in the reserved area on the screen. An example of the reserved area is shown below.



Reserved Area of Screen - Example

The default location for the reserved area is the first 24 positions of line one. This location was chosen for reasons of efficiency and because screens of every size have a top line. However, the reserved area can be anywhere on the screen. Its location can be changed by programmers with the use of the **VFMOPT** subroutine which is part of Portal.

How to Read this Manual

Every new user of any of the VDE products should read Chapters 2, 3 and 4.

People who will be setting up data entry jobs should read Chapters 5-15. Chapter 15, which is about VCS, describes a menu interface for computer users that can insulate them from the complicated control language used by the computer's operating system.

Those responsible for day to day operations will want to read Chapters 16-20.

Appendix A is the Error Messages and Codes chapter. Appendix B is for the system installer and contains information about the features that are unique to a particular computer and its operating system.

Chapter 2
Special Function Keys Overview

Introduction

The Viking Special Function Keys perform rapid movement across characters, fields, and records, either forward or backward. There are also specialized functions such as **Duplicate** (duplicate a field) and **Menu Request**. The user presses one or two keys to activate any function.

Before you begin to actually create your Form Sets, you need to be familiar with the Viking Special Function Keys. Viking provides 50 Special Functions which are a simple means of communicating with a program. The first 30 keys are Viking Supplied Functions, and the last 20 may be User Defined Functions.

We describe the 30 predefined Function Keys in generic terms, without referring to a specific physical key. The Function Key's action will be described in terms of the behavior as seen by the user. Default assignments for these Function Keys are illustrated in the keyboard diagrams at the end of the chapter.

VDE restricts the use of the Special Functions based on the type of Form (Record or Menu), the Entry mode (Create, Update, Verify or Examine) and the position in a field. The chart shown, in the section titled **Function Keys Allowed in Different Modes**, at the end of this chapter shows which functions are allowed with each Form and Mode. While a field is being modified, many of the Function Keys are disabled until the field is completed. That is, some of the Special Functions can only be used prior to keying data in the field, thus insuring data accuracy. Two Function Keys **Expert Mode** and **Delete Immediate** are used only in the Viking Forms Painter (VFP).

Chapter 13, **TERMKEYS - Defining Function Keys**, describes how specific physical keys are assigned to each of the Viking Special Functions by a set of assignment tables that are created and modified by running the Interactive Terminal Definition routine, **TERMKEYS.** This chapter also tells how User Defined Functions 31-50 may be set for programmers to develop their own Special Functions.

Function Key Definitions

Auto This function acts as a toggle switch to turn ***Auto Skip*** on and off. When ***Auto Skip*** is on the cursor will skip over all fields which have been automatically duplicated, or which belong to an active Skip Set. When ***Auto Skip*** is turned off, **AS** will not appear in the reserved area of the screen and the cursor will stop at the beginning of every field that is not a Protected or Locked field.

Character Back This function is a non-destructive backspace. The cursor is moved to the preceding position in the current field. When the cursor is already in the first position of the field, the request is ignored.

Character Forward This function moves the cursor forward one character in the current field, provided the current character is valid for that field. The movement is non-destructive. If the cursor was already on the last character of a field it is moved to the beginning of the next field. If the cursor is positioned on the last character of the last field the Form is completed.

Delete This function is a destructive backspace. The character to the left of the cursor will be replaced with a blank and the cursor will move back to that character. If the cursor is already on the first character of the field, the request is ignored. In **Edit/Insert** mode the characters to the right of the cursor will be moved to the left, replacing the deleted character.

Delete Immediate This function is used in the Viking Forms Painter (VFP) to delete the character under the cursor while painting a Form in Normal Text mode. This function is not available in Line Drawing mode.

Duplicate Data is copied from a Dup Buffer into the field and displayed on the screen. An error message will be displayed if the field does not have a Dup Buffer assigned to it or if the Dup Buffer does not contain any data.

Enter This function is used to signal field completion. A field may be completed, either by keying all required characters or by using this function key. When **Enter** is pressed in **Justify/Fill** mode the field is justified and the assigned fill character is used to complete the field.

When **Enter** is pressed in **Edit** mode the cursor merely advances to the next field, no justification is performed and no data is lost as a result. If a field has been assigned the 'return required' completion parameter, an **Enter** is required even when the field has already been filled with data. When painting a Form the **Enter** function will advance the cursor to the first character of the next line.

Exit Scroll This function is used only in Scrolling Forms and Word Wrap Fields. In Scrolling Forms, the entire Scrolling Form is completed unless you are in the Fixed Area and there are 'must enter' fields that have not been completed. In Word Wrap Fields, this function completes the field and moves forward to the next available field.

Expert Mode This function is used in the Viking Forms Painter to by-pass an intermediate menu while painting a Form.

Field Back This function moves the cursor to the first position of the current field. If the cursor is already in the first position, it will be moved to the first position of the previous field. The request is ignored if no previous field exists. If in a Scrolling Item, the cursor will move to the last field of the previous item or the last Fixed Field if no previous item exists. In a Word Wrap Field, the cursor will move up one line retaining the current column position.

Field Correct This function temporarily disables Update or Verify mode record protection of the current field. When the field is completed record protection will be restored. In Verify mode the user then must verify the new data.

Field Forward This function moves the cursor forward to the beginning of the next field to be entered. When the current field is the last field to be entered the Form will be completed. You can not use this function to move out of a 'must enter' field. In a Word Wrap Field, the cursor will move down one line retaining the current column position.

Flag Field This function invokes the routine **VFMFLG** which places the # character into the first position of the field and blank fills the remainder of the field. The field will then bypass Field Edits and it will not be stored in a Dup Buffer. Using Portal, programmers may modify **VFMFLG** to change the field contents in some other fashion.

Force Character This function is used to enter a character in a field when that character is not a member of the field's character sets. The next character keyed will be forced through the character sieves.

Forms Change This indicates a request to change to a different Record Form. The user will be prompted for the desired Record Form ID. If the Form ID is valid that Form will be used in place of the current Form. This function is normally valid only in Create mode.

Help This function asks the system to look in the Help File for a message to display. If VDE finds a Help message for the field, it displays the message on the screen. Field Help is described in Chapter 8.

Keyboard Shift The Reserved Area will display **KS** if shifting is active and blanks when not active. If you are not using the 029 keyboard translation, this function will shift lower case alpha's to upper case when it is active. When 029 translation is in use, this function will auto shift the numeric cluster of keys.

Last Scroll This function applies only to Scrolling Forms and Word Wrap Fields. If the cursor is in the Scrolling Fixed Area, this function is illegal. If it is in a Scroll Item, the cursor will advance to the last Scroll Item. The screen will be scrolled forward if necessary. In a Word Wrap Field, the cursor will be moved to the last line of the field.

Location Return This function requests a return to a previously accessed record. Commonly, the desired record is the last record keyed prior to executing one or more **Record Back** functions. In Create or Update mode, VDE takes you to the last record of the file. In Verify mode, VDE takes you to the next record to be verified. Inside a Scrolling Region of a Form it always moves you to the first Scroll Item.

Menu Request

This function causes a Record Form to be interrupted and requests a Menu Form. The Menu ID associated with the current Record Form will usually be used to select the Menu to be processed.

Mode

This function is used to change the field completion mode. The modes are **Justify/Fill, Edit/Replace,** and **Edit/Insert.** Each time you press the **Mode** key VDE changes from one mode to the next. The default mode is **JF.** Pressing the **Mode** key will change it to **ER** mode, pressing it again will change it to **EI,** and pressing it another time will take you back to **JF** mode. VDE displays the mode in the reserved area.

Justify/Fill mode is the default and is normally used for data entry applications. When **Enter** is keyed the field will be justified and the remaining character positions will be filled with the specified fill character. In this mode **Enter** destroys all of the characters from the cursor to the end of the field. Some of the Special Functions are prohibited in fields that have already been entered.

When **Edit/Replace** or **Edit/Insert** modes are active, the cursor navigation functions are allowed from anyplace in the field. Justification is never performed, the data is left as it is. **Enter** is treated just like **Field Forward**. **ER** and **EI** modes are used in word processing-like applications and in Update mode with existing data.

If **Edit/Replace** is active, a character keyed over another character replaces it. In **Edit/Insert** mode the new character is inserted into the field. The character under the cursor, and all characters in the field to its right, are moved to the right. The right most character in the field is lost.

Page Back

This function applies only to Scrolling Forms and Word Wrap Fields. If the cursor is in the Scrolling Fixed Area, it will do a **Record Back**. If it is in a Scroll Item the screen will be scrolled backward to refill the Scroll Region. If the cursor is already in the first Scroll Item, it will go back to the Fixed Area. If the cursor is in a Word Wrap Field, this function will Scroll the field back one full window.

Page Forward This function applies only to Scrolling Forms and Word Wrap Fields. If the cursor is in the Scrolling Fixed Area, it will advance you to the first Scroll Item. If it is in a Scroll Item, the screen will be scrolled down to refill the Scroll Region. If it is in a Word Wrap Field, it will Scroll the window down to refill the window.

Record Back This function returns the cursor to the first field to be entered in the current record. When the cursor is already positioned in that field, the Form is completed. VDE will then present the previous data record and its Form.

Record Correct This function disables Update/Protect or Verify mode to allow the entire record to be modified. This function automatically changes the Entry mode to Update/Unprotected until the record is completed. In Verify mode the user must then re-verify the corrected record.

After using this function in **Justify/Fill** mode, be careful to use the **Field Forward** function to skip over fields that contain good information. When you press the **Enter** key to complete a field the data will be replaced with the specified fill character. e.g., a zero fill field will be changed to zeroes, destroying the previous data.

Record Delete This function is used to request that the current Data Record be deleted. If you are positioned on a Scrolling Item, just that item will be deleted. If you are in a Word Wrap Field, the function will delete the entire line under the cursor.

Record Forward This function moves the cursor forward to the last field to be entered on the current Form. If the cursor is already positioned in the last field the Form will be immediately completed. VDE will then present the next Form. In Create mode this function will not pass over 'must enter' fields that have not yet been entered.

Record Insert This function enables the user to request that a Data Record be inserted immediately in front of the current Data Record. If you are positioned on a Scrolling Item, you will see a new Item inserted immediately ahead of the current Item and you will be in Create mode for entry of that Item. If you are in a Word Wrap Field, this function will insert a new line, moving your text from the current line to the end of the field down one full line.

Reset This function is usually required to clear an error message. Until **Reset** is pressed, any other characters will be ignored, the bell will ring, and the error message will continue to be displayed.

Show Fields This function is valid only in the Verify mode. It displays the contents of the field currently being verified. The end user must still verify the field. In a Word Wrap Field, this function will only display the current line being verified.

Tab This function moves the cursor forward one or more fields to the next field to be entered that has the 'Tab Stop' attribute. If such a field is not found the Form will be completed. **Tab** will also stop the cursor on a 'must enter' field that has not been entered, even though it does not have the 'Tab Stop' attribute. In a Word Wrap Field, this function will move the cursor to the next tab stop in the text line. The default tab stops are every fifth character.

Function Key Considerations

Some of the Special Function Keys have slightly different behavior when used with Scrolling Record Forms or in Word Wrap Fields. Also, they may have different results when keyed from the Fixed Area than when keyed from the Scrolling Region. In general they behave the same way, or analogous to the way, that they work with regular Record Forms. The following table shows the different responses.

FUNCTION		
	(F)	Fixed Part
	(S)	Scrolling Region
	(W)	Word Wrap Fields
EXIT SCROLL	(F)	Advances to next Form.
	(S)	Advances to next Form.
	(W)	Advances to next field.
FIELD BACK	(W)	Moves up to same position in previous line.
FIELD FORWARD	(W)	Moves down to same position in next line.
LAST SCROLL	(F)	Invalid function.
	(S)	Advances to the last Scroll Item in the Scroll Region.
	(W)	Advances to last line in field.
LOCATION RETURN	(F)	Returns to last active data record.
	(S)	Returns to the first Item in the Scroll Region.
PAGE UP	(F)	Same as Record Back.
	(S)	Scrolls backward to refill the Scroll Region. (Backs up to the Fixed Area when on the first Item.)
	(W)	Scrolls back one window.
PAGE DOWN	(F)	Advances to the first Scroll Item.
	(S)	Scrolls forward to refill the Scroll Region.
	(W)	Scrolls forward one window.
RECORD BACK	(F)	Backs up one record.
	(S)	Backs up to the previous Scroll Item.
RECORD DELETE	(F)	Deletes entire record.
	(S)	Deletes current Scroll Item.
	(W)	Deletes the line under the cursor.
RECORD FORWARD	(F)	Advances one record.
	(S)	Advances to the next Scroll Item.
RECORD INSERT	(F)	Inserts a new record.
	(S)	Inserts new Scroll Item before current Item.
	(W)	Inserts a new line.
TAB	(F)	Advances to next Tab stop or to first Scroll Item.
	(S)	Advances to next Scroll Item
	(W)	Advances to next Tab stop (every 5th character) on line.

NOTE: In Scrolling Forms "Item" refers to one occurrence of the scrolling part of the Data Record. A Scroll Item can occupy more than one screen line.

Function Keys Allowed in Different Modes

SPECIAL FUNCTION	MENU	RECORD FORMS				FIELD CHANGED
		CREATE	UPDATE	VERIFY	DISPLAY	J/F MODE
AUTO SKIP	YES	YES	YES	YES	---	YES
CHARACTER BACK	YES	YES	---**	YES	---	YES
CHARACTER DLTE	YES	YES	---**	---	---	YES
CHARACTER FWD	YES	YES	---**	---	---	YES
DUPLICATE FIELD	YES	YES	---**	YES	---	YES
EXIT SCROLL *	---	YES	YES	---	YES	---
FIELD BACK	YES	YES	YES	YES	---	---
FIELD CORRECT	---	---	YES	YES	---	---
FIELD FORWARD	YES	YES	YES	---	---	---
FLAG FIELD DATA	YES	YES	---**	---	---	YES
FORCE CHARACTER	YES	YES	---**	---	---	YES
FORMS CHANGE	---	YES	---**	---	---	---
FUNCTIONS 31-50	YES	YES	YES	YES	YES	---
HELP	YES	YES	YES	YES	YES	YES
KEYBOARD SHIFT	YES	YES	YES	YES	---	YES
LAST SCROLL *	---	YES	YES	---	YES	---
LOCATION RETURN	---	YES	YES	---	YES	---
MENU REQUEST	---	YES	YES	YES	YES	---
MODE	YES	YES	YES	---	YES	YES
PAGE BACKWARD *	---	YES	YES	---	YES	---
PAGE FORWARD *	---	YES	YES	---	YES	---
RECORD BACK	---	YES	YES	YES	YES	---
RECORD CORRECT	---	---	YES	YES	---	---
RECORD DELETE	---	YES	---**	---**	---	---
RECORD FORWARD	YES	YES	YES	---	YES	---
RECORD INSERT	---	YES	YES	YES	---	---
RESET	YES	YES	YES	YES	YES	YES
RETURN	YES	YES	---**	YES	---	YES
SHOW FIELDS	---	---	---	YES	---	---
TAB	YES	YES	YES	---	---	---

** N O T E **

The Field Changed column applies only to **JF** mode. In **EI** or **ER** mode all the functions are allowed and will not change the field.

* The Scrolling Functions only work with Scrolling Forms or in Word Wrap Fields.

** Update and Verify support these functions while either **FIELD CORRECT** or **RECORD CORRECT** is active.

Considerations While Painting Forms

The [Up], [Down], [Left], and [Right] [Arrow Keys] will be used to position the cursor while painting a Form with the Viking Forms Painter regardless of any Special Function assignment to these keys in other Viking applications. [Delete Immediate] and [Expert Mode] are only available as Special Function Keys in the Viking Forms Painter.

Function Keys in Word Wrap Fields

Several of the Function Keys take on a different behavior upon entering a Word Wrap Field. This is done to provide ease of movement and editing since Word Wrap Fields can contain many lines of text.

[Enter]	becomes **Line Completion.** If text follows the current cursor location, that text will be moved to the next line when the Entry Mode is **Justify/Fill**.
[Exit Scroll]	allows you to exit the Word Wrap Field at any time. If you are verifying the field, it will complete the verification of the field if all of the remaining characters are blanks.
[Field Back]	becomes **Line Back.**
[Field Forward]	becomes **Line Forward.**
[Last Scroll]	will move you to the last line of the Word Wrap Field, changing the page displayed when necessary.
[Record Delete]	becomes **Line Delete.**
[Record Insert]	becomes **Line Insert.**
[Scroll Page Forward]	this function moves the next page into the display window, when the Word Wrap Field consist of more lines than can be displayed in its window.
[Scroll Page Back]	This function will move back a page of text in the window.

Tab becomes a convenient way to move across a line five columns (default) at a time.

Viking Terminal Support

The Viking system supports many different terminals, monitors and keyboards. The determination as to which terminal you are currently using is made by requesting your systems's identification. It is imperative that your terminal is properly identified before executing any Viking tasks, this is explained in detail in Appendix B.

The **TERMKEYS** program is used to assign the Viking ID to a terminal name and to change Function Key Assignments. It must be executed on the terminal that is to be affected by the change. For example, if you want to alter the Function Key Assignments for the VT-100 terminal, you must be using a VT-100 terminal when you execute **TERMKEYS**. The default assignments are contained in the file **TERMINAL.KBD**.

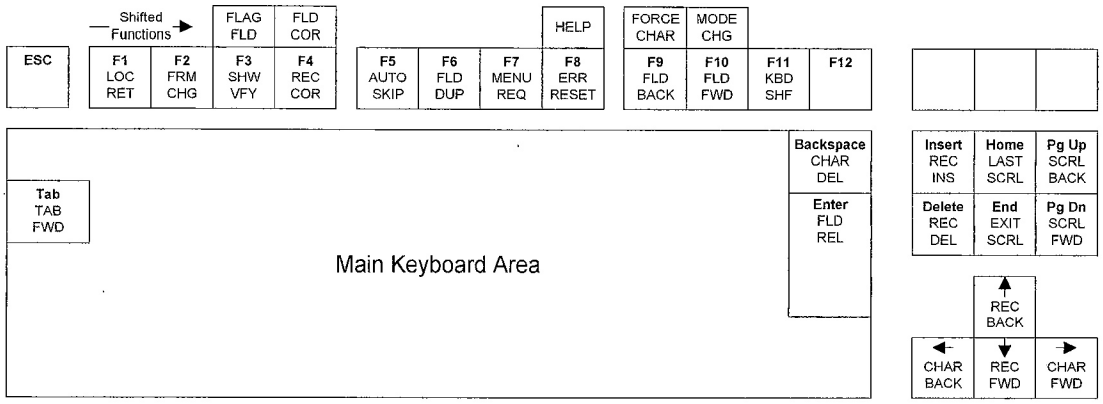
Once you have set the Viking ID's and Function Keys, you may save your new definitions in the file, **TERMINAL.VSS.** This file must then be moved to the directory pointed to by the Environmental Variable, **VFMKBD** (VMS & Unix Systems) or **VSSVDE** (DOS Systems). Optionally, you may create an Object File which can be Linked into your tasks.

Terminal Description	UNIX TERM =	VAX/VMS Device Name	VIKING ID	Table Name
DEC VT100 Series	vt100, 101,... 132	vt100, 101,... 132	100	vt100tbl
DEC VT200 Series	vt200, 220	vt200	200	vt200tbl
DEC VT300 Series	vt300, 320	vt300	300	vt200tbl
DEC VT400 Series	vt400	vt400	400	vt200tbl
IBM 3151, 3161 & 3163	ibm3151, 3161, 3163	ft2	061	ibm316x
IBM 3162 & 3164 Color	ibm3162, 3164	ft3	062	ibm316x
Hewlett Packard No/Atr	hp	ft1	001	helwpack
Hewlett Packard w/Atr	hp2621	ft4	004	hewpack4
Wyse 50	wy50	ft5	050	wyse50
Wyse 60	wy60	ft6	060	wyse50
Bull VIP 7201	hw72	ft7	072	vip720lt
BULL BDS 3	-not assigned-	vt200	200	vt200tbl
BULL BDS 5 & 7	hw78	-not assigned-	075	bds5and7
BULL BDS 5 & 7 (rev7)	hw78	-not assigned-	077	bds5and7
X-Windows xterm	xterm	-not assigned-	081	xtrmtbl
X-Window sun	sun, sun-cmd	-not assigned-	083	suntable
AT&T 605 & 610	610	ft8	043	att610
MS/DOS Monochrome	not applicable	not applicable	021	DOS
MS/DOS Color Monitor	not applicable	not applicable	022	DOS
XENIX console	ansi	not applicable	023	consxenx
XENIX color console	ansic	not applicable	024	consxenx
IBM AIX console	ibm6155	not applicable	025	aixcons
IBM AIX color console	ibm5081	not applicable	026	aixcons
UNIX 386 console	ansi, AT386-M	not applicable	027	consunix
UNIX 386 color console	ansic, AT386	not applicable	028	consunix

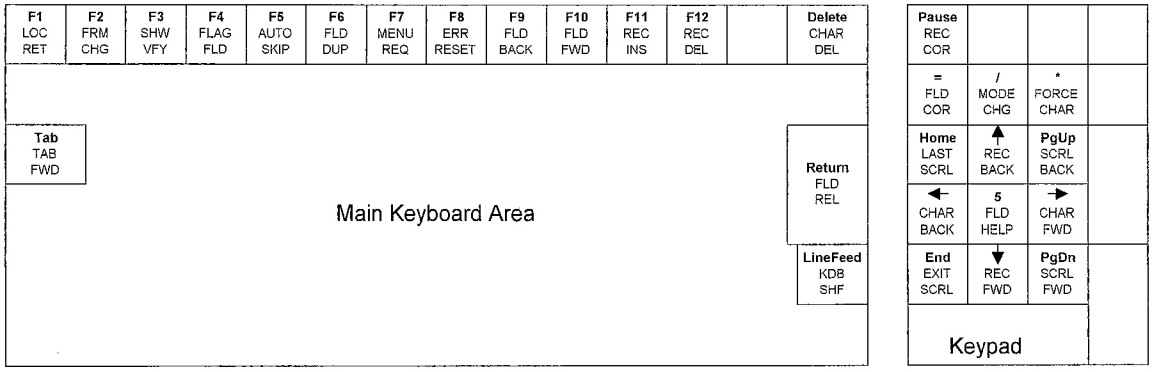
Function Key Assignments

Each terminal may have its own set of Function Key Assignments which can be customized with the routine named **TERMKEYS.** Our default function assignments have assumed the terminal is running in its native mode and using a standard keyboard. If you have an optional keyboard, you will have to use **TERMKEYS** to establish the Function Key Assignments.

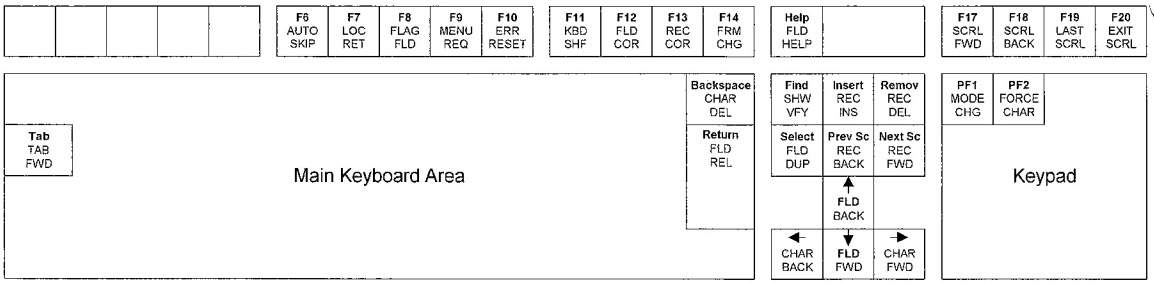
IBM PC Keyboard, XENIX and UNIX 386 Consols, XTERM, IBM 3151, etc.



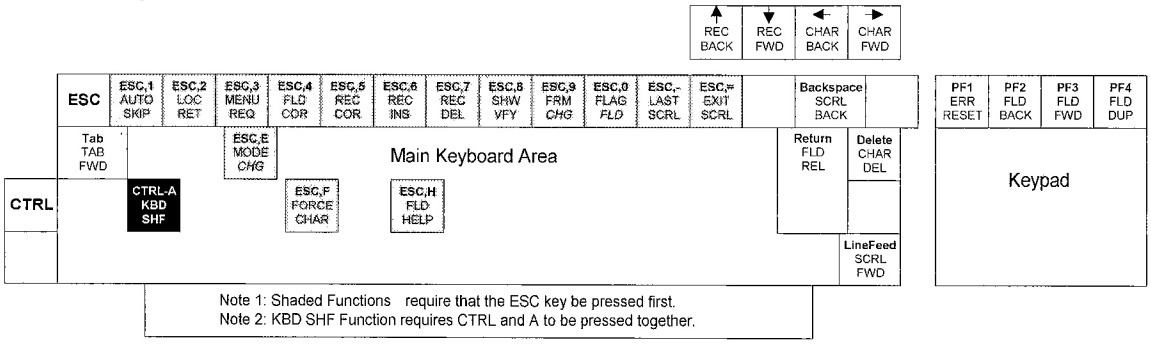
SUN CONSOLE



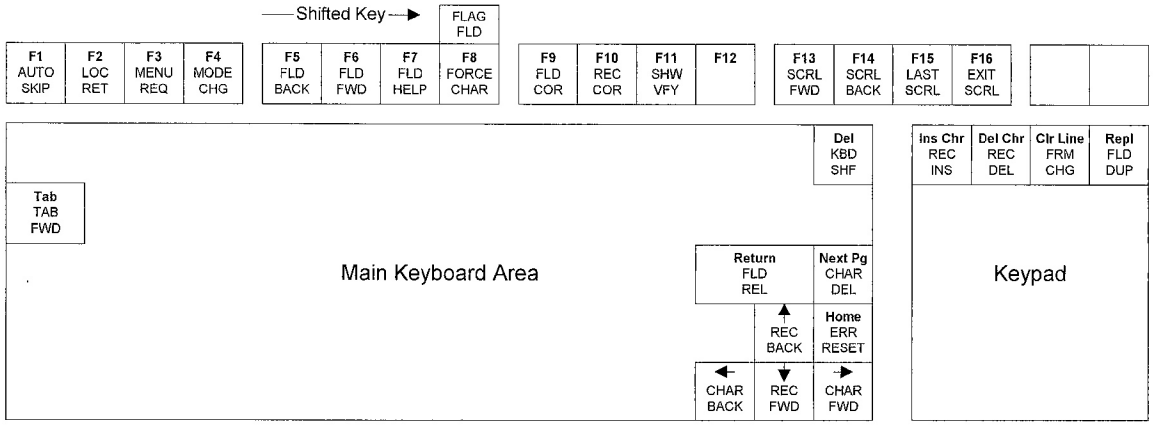
VT-200/300/400 Keyboard



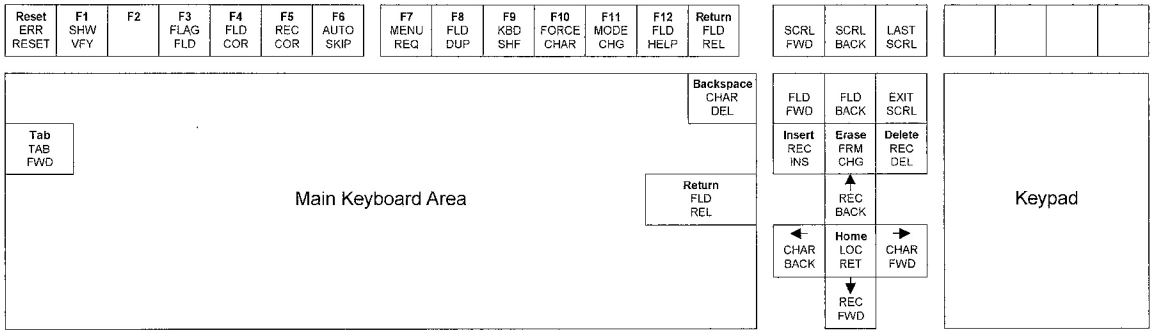
VT-100 Keyboard



WYSE-50



BDS5and7



Chapter 3

Introduction to the VDE Program

Overview

The VDE program is used for data entry into standard ASCII sequential files. Although it has many other uses, the most common use of VDE is to perform production data entry of the type often done on specialized key entry machines, such as key-to-disk and keypunch machines. VDE makes it possible to do this with general purpose computers and standard keyboards and monitors.

** N O T E **

There are some slight variations in the versions of VDE for different computer systems. These differences are described in Appendix B. Be sure to read it very carefully before you begin to use VDE.

Features and Capabilities

VDE has many features and capabilities to make it easy to quickly and accurately key enter data for a variety of situations and circumstances.

Data Entry Applications Without Programming

You do not have to be a programmer to use VDE. In fact, most of the people who create VDE applications are not programmers. No programming is necessary to generate applications that will Create, Update, Verify, or Search data files. Complete data entry applications can be generated in minutes.

VDE Can Be Customized

Portal is an option to VDE that allows programmers to develop customized User Exit routines and Field Edit routines for complex applications. If this optional product is licensed to your organization, detailed information will be contained in a separate manual.

Fill in the Blanks

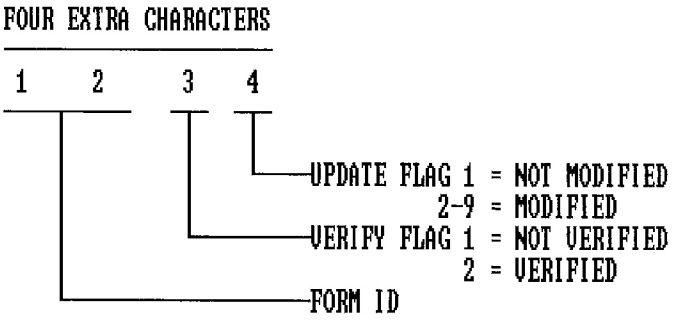
VDE uses the efficient, human-engineered fill-in-the-blanks Record Forms created using the interactive Viking Forms Painter, which is described in Chapter 5. These Forms use the data validation capability and Special Function Keys described in Chapter 2 to enable data to be entered rapidly and accurately.

Standard Sequential Files

VDE works with standard sequential files that are compatible with most computers and programs. The data files created by VDE are ready to be processed by other programs or transmitted to another computer.

**** N O T E ****

VDE appends four additional characters onto the end of each Data Record. The first two characters are the Form ID. The third character is the Verify Flag. The fourth character is the Update Flag. You may remove these Form and Flag characters easily by executing the program **CONFILE**, described in Chapter 10. However, after the removal of these characters, the resulting new data file will no longer be accessible by VDE.



Four Extra Characters

Data Accuracy

Data accuracy is the cornerstone of VDE. In addition to the standard Field Edit routines, VDE implements the concept of re-keying sensitive data to verify that it is correct. This is a tried and proven technique that has been shown to produce nearly error-free data. Improved data accuracy results in fewer re-runs and substantial labor savings associated with checking the results of the programs that depend on the data.

Customized Option File

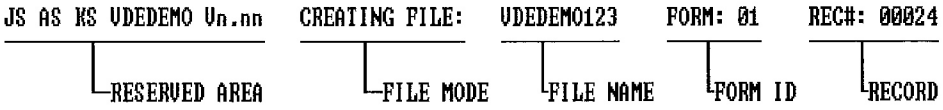
The VDE Option File allows for each job to be customized and take advantage of the advanced features of VDE. Setting up an Option File is easily done using the **VDEOPT** Program described in Chapter 9.

Reformatting and Conversion

Data files can be reformatted and converted using the VDE File Conversion Utilities which are described in Chapter 10. Again, no programming is required.

Status Line

VDE uses the top line of the screen as a status line. The first 24 characters comprise the VFM reserved area which shows the current keying mode, the Auto Skip/Dup status, the Keyboard Shift status the Form Set name (job) and its version number,. The remainder of the line is used by VDE to display the file mode, name and record information.



Status Line Illustration

Table of Features

The **Viking Data Entry Features Table** shows which VDE features are an inherent part of the Form created by the Viking Forms Painter and which are provided by the VDE program. Features provided by VDE will be described in this section. Features that are a part of VFP are described in Chapter 5.

Viking Data Entry Features Table

VDE Features

- **Six Entry modes**
  - Create
  - Verify
  - Sample Verify
  - Update
  - Examine
  - Search
- **File Processing Features**
  - Record Insert, Delete, Modify, and Append
  - Variable length ASCII records
  - 99 record types (or levels)
  - Transaction Logging
  - Scrolling
  - Shared Files (Trail Verify)
- **Operator Statistics**
  - By operator, by job, or by batch
  - Keystrokes
  - Number of Records
  - Inserts, Deletes, Updates
- **Interrupt and restart**
- **Balancing**
  - Record Crossfooting
  - Batch Total Balancing
  - Batch Sub-Total Balancing
- **Auto Dup the System Date and Time**
- **Data File Conversion Utilities**
- **Fully Menu Driven**
- **Re-key Verify Mode**
  - Normal Verify
  - Trail Verify
  - Sample Verify
  - Conditional Verify

Basic Form Functions

- **Three Field Completion Modes**
  - Justify/Fill
  - Edit/Replace
  - Edit/Insert
- **Three types of Duplication**
  - Automatic, Manual & Plugging
- **Re-key Verify Mode**
  - Optional by Field
  - Field Correct
  - Character Correct
  - Record Correct
- **Character Validation**
  - Eight Character Sieves
  - Character Sieve Override
- **Table Look-ups**
  - Internal and External Tables
  - Optional Substitution
- **Error Messages**
  - System and User
  - Optional Reset Required
- **Simulated 029 Keyboard - Optional**
  - Embedded Numeric Cluster
- **Forms Painter to create applications**
- **Field Attributes**
  - Size
  - Must Enter
  - Must Complete
  - Verify
  - Non-display
  - Display Attributes
  - Negative (or credit) signs Leading, or Overpunched
  - Character Sets
  - Left or Right Justify
  - Zero or Blank Fill
  - Field Release Required
  - Protected fields
  - Location in Data Record
  - Must Verify Empty Fields
  - Keyboard Shift On

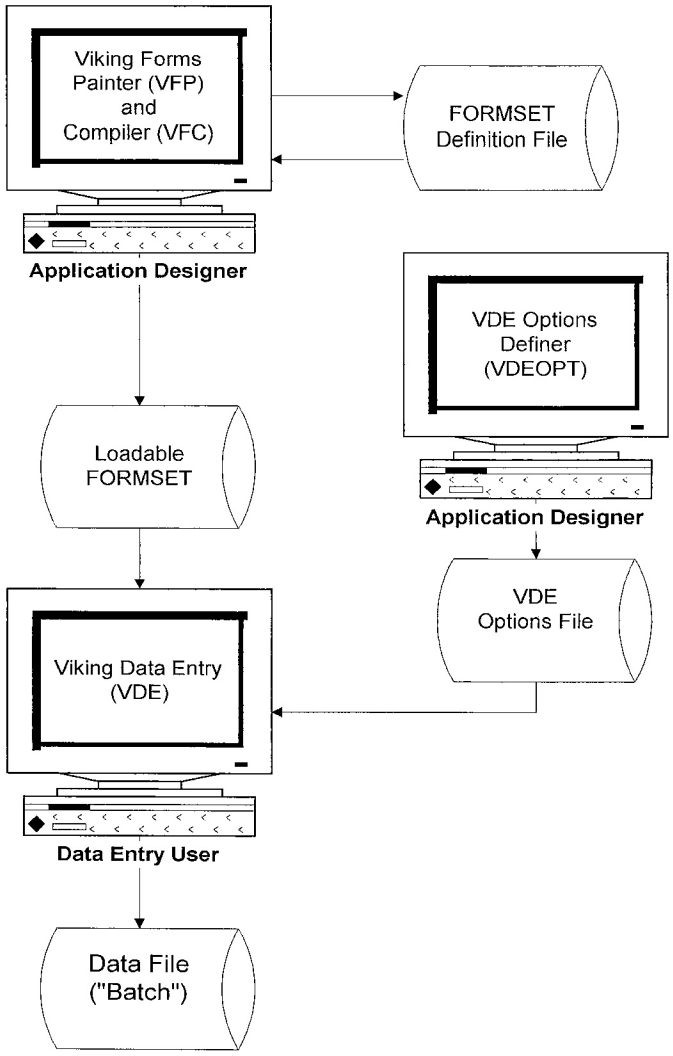
Creating Viking Data Entry Applications

The five easy steps necessary to generate a VDE application are:

1. Design the record layouts of the Data Records in the file.
2. Create the necessary Record Forms using the Viking Forms Painter
3. Output a Loadable Form Set from the Viking Forms Compiler, VFC.
4. Define the VDE options.
5. Run VDE using the Loadable Form Set from step 3.

None of the five steps require any programming.

Schematic of Creating & Using a VDE Application with Loadable Form Sets



DOS Operating System

VDE and Portal are available for computers that use the DOS operating system. They will execute on almost any LAN configuration as well as on stand alone machines.

VDE and Portal support color, which allows you to paint and display your Forms using the 16 Foreground and 8 Background colors that are available on the DOS color monitors in text mode. A hardware check is used to determine if you are using a color monitor. This test determines if the Forms will execute with color or monochrome display attributes. The **TERMKEYS** program allows you to disable the color support should you desire to do so.

VDE and Portal use the Function Key assignments as shown in Chapter 2. The **TERMKEYS** program can be used to customize these defaults and create a new **TERMINAL.VSS** file with your customized assignments.

The VDE and Portal tasks will not modify your cursor display. You may use the routine **VCURSOR** to modify your cursor display. Execute **VCURSOR** without command line arguments to see suggested values for the monitor you are using. Example:

C:> vcursor 0 14 [Enter] {Block Style Cursor on VGA Monitor}

DOS Directories

The VDE and Portal systems are copied into one directory with six sub-directories named **CONTROL, DATA, DEVEL, HELP, TABLES,** and **VDEPROD.** The Portal system will be copied into a language specific sub-directory named **CLANG, COBLANG** or **FORLANG.** Initial default files will be moved into the **HELP** and **CONTROL** directories.

The directory identified as **DEVEL** is not restricted to a single directory. Any number of individuals in any number of directories can act as a developer of Viking jobs. The **DOS Flow Chart** below illustrates that new jobs can be developed without affecting the current production jobs.

Likewise, the directory identified as **DATA** is not required to be a single directory. Each user may log into and remain in a unique directory to use the production system. If this is your choice, then please note that VDE will place the data files into the current users working directory unless you use the VDE Option Files to specify where the data files are to reside. This is an important consideration if one user is to key a file and another user is going to verify the file.

DOS Environment Variables

VSSVDE and PATH

The location of the Viking Programs must be set using the variable **VSSVDE.** The variable **PATH** must also be set to this location. These variables allow the user to execute the programs with simple commands and to locate all of the demonstration files. Your license file, **VLICENSE.VSS,** and all sub-directories are also located with this variable.

VFMUSER

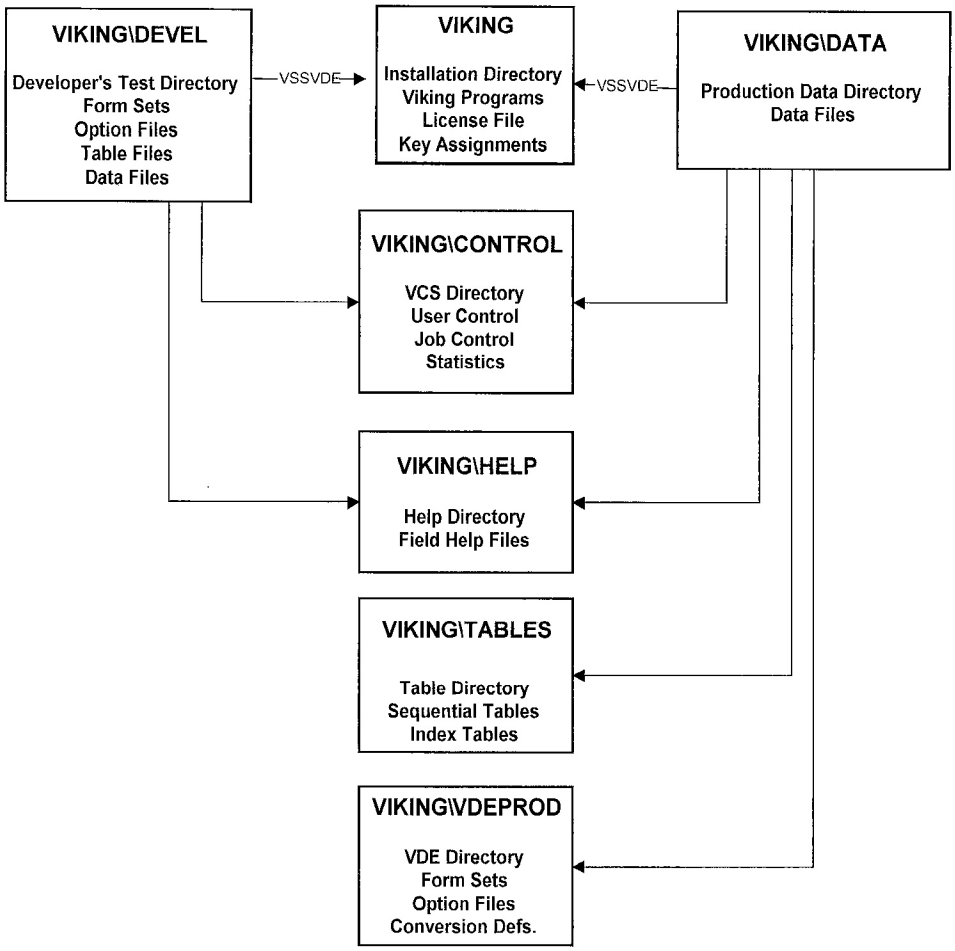
If you want VDE to produce operator statistics files and you are not running from the **Viking Control System, VCS**, set **VFMUSER** to an Operator ID (up to 4 characters). If you are running with a network license, you must either set **VFMUSER** or use **VCS**. Note that only one person at a time may use an Operator ID.

KEY029

If the VDE Option File specifies an optional 029 Keyboard translation, this variable may be set to **Y** or **N** to eliminate the need for the operator to answer the **"USE 029 KEYBOARD?"** question.

The DOS Environment Variable, **VSSVDE** is extremely useful at the time you install a new release of our software. Rather than installing the new release into your existing Viking directory, you can create a new directory. During the testing of a new release, you can point your version of the DOS Environment Variable to the new directory. Your production people can continue to use the previous release because their Environment Variable points to the old directory.

To switch your production users to the new release, you simply modify their Environment Variables to use the new directories. When you are comfortable that the new release is performing your tasks as it should, you can delete the old directories.



DOS Flow Chart

DOS Installation

The VDE and Portal systems are delivered on diskettes along with a pre-built license file or a License Installation Sheet which displays your license information which is required for the installation to succeed. You should have 4MB of space available on the target disk. The instructions below assume that the installation is being done on drive **"C:"**. If your installation of VDE or Portal is on a network, you must login as the **SUPERVISOR** and your Viking directory should be created on the server.

1. Boot your DOS system if necessary. If on a network, login as **SUPERVISOR.**

2. **A:INSTALL** [Enter]

   You will be able to modify the default installation directory. Follow the instructions on your screen. After the files have been installed, the License Installation Routine, **VINSTALL**, will be executed to initialize your License File and create a file named **VFMVARS.BAT** which can be used to set your Viking Environment Variable.

Novell Network Systems:

You must now set the search modes and grant the trustee rights to the appropriate users or groups. If all users belong to a common group, "groupname", the following statements will set the rights. If you have multiple users and no common group, you may wish to use FILER to grant these rights.

> SMODE . 2 [Enter] {set default search mode}

> GRANT R F TO "groupname" [Enter]

> GRANT ALL FOR CONTROL TO "groupname" [Enter]

> GRANT ALL FOR HELP TO "groupname" [Enter]

> GRANT ALL FOR TABLES TO "groupname" [Enter]

> GRANT ALL FOR VDEPROD TO "groupname" [Enter]

> GRANT ALL FOR DATA TO "groupname" [Enter]

> GRANT ALL FOR DEVEL TO "groupname" [Enter]

3. You are now ready to set the Environment Variables. This process differs if you are using a network or a stand alone DOS system.

Stand Alone Systems: Add **VFMVARS.BAT** to **AUTOEXEC.BAT**:

C:> COPY C:\AUTOEXEC.BAT+VFMVARS.BAT C:\AUTOEXEC.BAT [Enter]

Network Systems:

You must edit the default LOGIN script or each individual user LOGIN script and add the Environment Variable settings as found in the file **VFMVARS.BAT.** This process will differ somewhat depending upon your network. The work station users must have **VSSVDE** pointing to the new Viking directory. For example, if the following line was present in the **VFMVARS.BAT** file:

SET VSSVDE=**F:\VIKING**

Then the appropriate script in the Novell LOGIN would be:

DOS SET VSSVDE=**F:\\VIKING**

The Viking Directory must be added to your search list. For example, if the Main Directory is **F:\VIKING**, then the LOGIN Script should have:

MAP INS S16:=**F:\VIKING**

4. Testing Your Installation

C:> VFMVARS [Enter]	{This will happen automatically on boot}
C:> CD .. [Enter]	{Move out of the Viking Directory}
C:> VCS [Enter]	{Execute the Viking Control System}

** Type in **DEMO** and press [Enter] and then press [Enter] again for the Password, you should be able to execute any task, use [Tab] to Exit the Menus, respond **Y** to Exit.

If you are on a stand alone system, you should re-boot and verify that the variables are set properly by running VCS after the boot completes.

If you are on a network, logout and then login as a user and run VCS. If you have modified the login scripts properly, you should see the Viking Menus appear as above.

DOS Operating System Specifics

You should be running DOS 3.3 or above. Most of the Viking tasks will run under DOS 2.0, but file sharing and record locking are not generally supported.

Out of Environment Space

If you received the message **"OUT OF ENVIRONMENT SPACE",** you must modify your **CONFIG.SYS** file to allow more environment space.

In your **CONFIG.SYS** file, look for a command as follows:

SHELL=C:\DOS\COMMAND.COM ...

Edit this command, or if not present, add the command as follows:

SHELL=C:\DOS\COMMAND.COM C:\DOS /P /E:1024

Re-boot your system and the message should not appear.

Setting DOS Environment Variables

The DOS **SET** command is used to set your Environment Variables as follows:

C:> SET VSSVDE=C:\VIKING
C:> SET VFMUSER=xxxx {where "xxxx" is your Operator ID}

DOS Control System Options

The VCS Menus may be used to isolate your Network users from the operating system. For example, you may alter the user's Network login procedure to execute the VCS task, followed by the Network **"logout"** command. A user, upon booting or logging into your system, will be presented with the VCS login procedure and when they Exit from VCS, they will Exit from the Network. If you prefer to bypass the VCS login, you may set the VCS user's password to blanks

(no password required). You may also establish a time-out for VCS, which effectively would log out users that have not made a Menu selection in **"nn"** minutes.

Login Examples:

```
... set VSSVDE and PATH
...
VCS                     {require VCS login, no time-out}
-or-
VCS -15                 {require VCS login, time-out in 15 minutes}
-or-
VCS -20 MAG             {VCS login as MAG, time-out 20 minutes}

LOGOUT                  {log out of the Network}
```

A similar procedure can be used for stand alone DOS systems by modifying the **AUTOEXEC.BAT** file, but in this case you must put the user into a command procedure loop that in effect will never exit the **AUTOEXEC.BAT.** The user would simply have to shut down the DOS system with the Menu system active.

For example:

```
....
:LOOP_VCS
VCS
ECHO    "You may now power down your system"
PAUSE
GOTO LOOP_VCS
<eof>
```

If the user does not shut down the system they will simply be looped back to executing the VCS task. This can be very effective for DOS users that need to be isolated from the operating system.

DOS Differences and Restrictions

Viking Programs are subject to the constraints of the MS/DOS Operating System which includes:

- File names are limited to eight (8) characters.
- File name extensions are limited to three (3) characters.
- Form Sets cannot exceed 64K bytes.
- All tasks run in the standard 640K DOS memory.

You should attempt to make as much of the standard memory available as possible for VDE. This would include loading DOS and other memory resident programs into High memory.

DOS Naming Conventions

File Name	Description
form_set_name.VFD	Form Set definition file output by **VFP**.
form_set_name.VLF	Loadable Form Set output by **VFC**.
form_set_name.OBJ	Form Set Object output by **VFC**.
form_set_name.LIS	Form Set Print File output by **VFC**.
"name".TBL	Table files output by **VSSTABLE**.
form_set_name.HLP	Field Help file output by **VHELPMNT**.
form_set_name.HAS	Field Help ASCII file output by **VHELPASC**.
form_set_name.HLS	Field Help Print File output by **VHELPMNT**.
form_set_name.HBK	Previous version of Field Help renamed by **VHELPMNT**.
"name".KBD	Terminal Function Key Definitions output by **TERMKEYS**.
"name".LIS	Terminal Function Print File output by **TERMKEYS**.
form_set_name.VEO	VDE Options File output by **VDEOPT**.
form_set_name.BCK	Previous version of VDE Options File.
"name".CON	File Conversion Definition file output by **CONDEF**.
"batch".CUR	**VDE** opens this file for output. Indicates active batch.
"batch".DAT	**VDE** data file name if not using name controls option.
"batch".TRJ	**VDE** temporary transaction journal file.
"batch".DS1	**VDE** data file in Create stage.
"batch".DS2	**VDE** data file in Verify stage.
"batch".DS3	**VDE** data file is completed.
"batch".DS4	**CONFILE** default output file.
"batch".DS5	**VDE** data file after conversion by **CONFILE.**
"batch".BAK	**VDE** previous version of the data file.
"batch".SDF	**VDE** temporary file for Trail Verify version.
"name".VKD	**ISAMKEYS, ISAMLOAD** Key Definition files.
USERFILE.RPT	Listing of VCS users.
JOBFILE.RPT	Listing of VCS jobs.
JOBSTAT.RPT	**VDE** Job Statistics.
USERSTAT.RPT	**VDE** User Statistics.
BATCHSTS.RPT	**VDE** Batch Statistics.

UNIX Operating Systems

VDE and Portal are available for a large number of computers that use variations of the UNIX operating systems. The following instructions assume that you are using the Bourne shell.

VDE and Portal use the Function Key assignments as shown in Chapter 2. The **TERMKEYS** program can be used to customize the defaults contained in the file **TERMINAL.KBD** and create a new **TERMINAL.VSS** file with your customized assignments. You may create multiple **TERMINAL.VSS** files placed in separate directories and each user can set the Environment Variable, **VFMKBD**, to point to the file they wish to use.

Unix Directories

The VDE and Portal systems are copied into a single directory. The installation program will create seven (7) sub-directories named **control, data, devel, help, keys, tables**, and **vdeprod.** Initial default files will be moved into the **control, keys** and **help** directories. Viking makes use of Environment Variables to locate your directories, giving you the flexibility of re-naming them if you wish.

A production system should never have **VFMKBD, VCSDIR, VFMHLP, VFMTBL,** and **VDEPATH** pointing into the distribution directory. The directory pointed to by **VDEPATH** must never contain Loadable Form Sets unless they are to be used by the VDE task.

The directory identified as **devel** is not restricted to a single directory. Any number of individuals in any number of directories can act as a developer of Viking jobs. The **Unix Flow Chart** below illustrates that new jobs can be developed without affecting the current production jobs.

Likewise, the directory identified as **data** required to be a single directory. Each user may log into and remain in a unique directory to use the production system. If this is your choice, then please note that VDE will place the data files into the current users working directory unless you use the VDE Option Files to specify where the data files are to reside. This is an important consideration if one user is to key a file and another user is going to verify the file.

Unix Environment Variables

VFMEXE and PATH

The location of the Viking Programs must be set using the variable **VFMEXE.** UNIX systems must also set the variable **PATH** to this location. These variables allow the user to execute the programs with simple commands and to locate all of the demonstration files. Your license file, **vlicense.vss,** is also located with this variable.

VFMKBD

This variable locates the file **TERMINAL.VSS**. The Special Function Key assignments are carried in this file. All default Viking Programs find and load this file in order to execute. The default **TERMINAL.VSS** file is moved into this directory during installation.

VDEPATH

VDE will use the variable **VDEPATH** to locate the production Loadable Form Sets (**form set name.VLF**) and their Option Files (**form set name.VEO**) for an application. **CONFILE** also uses this variable to locate the Conversion Definition Files (**name.CON**).

VFMTBL

This variable is used by the Field Edit subroutines, **SEQTABLE** and **VIXTABLE,** to locate all Sequential and **ISAM** table files, see Chapter 7. **VIXTABLE** also uses this variable to locate the Loadable Form Sets used to maintain the **ISAM** files.

VFMHLP

This logical should point to the device and directory that contain the Field Help files (**form set name.HLP**). All Help files delivered with the system are moved into this directory during installation.

VCSDIR

This logical locates the VCS Control files, **VCSJOBS.DAT** and **VCSUSERS.DAT.** The VDE statistics file, **vdestats.dat,** is also placed in this directory. UNIX systems also place the individual user statistics file, **vstsuser.sts,** in this directory. Default **VCSUSERS.DAT** and **VCSJOBS.DAT** files are moved into this directory during installation.

VFMUSER

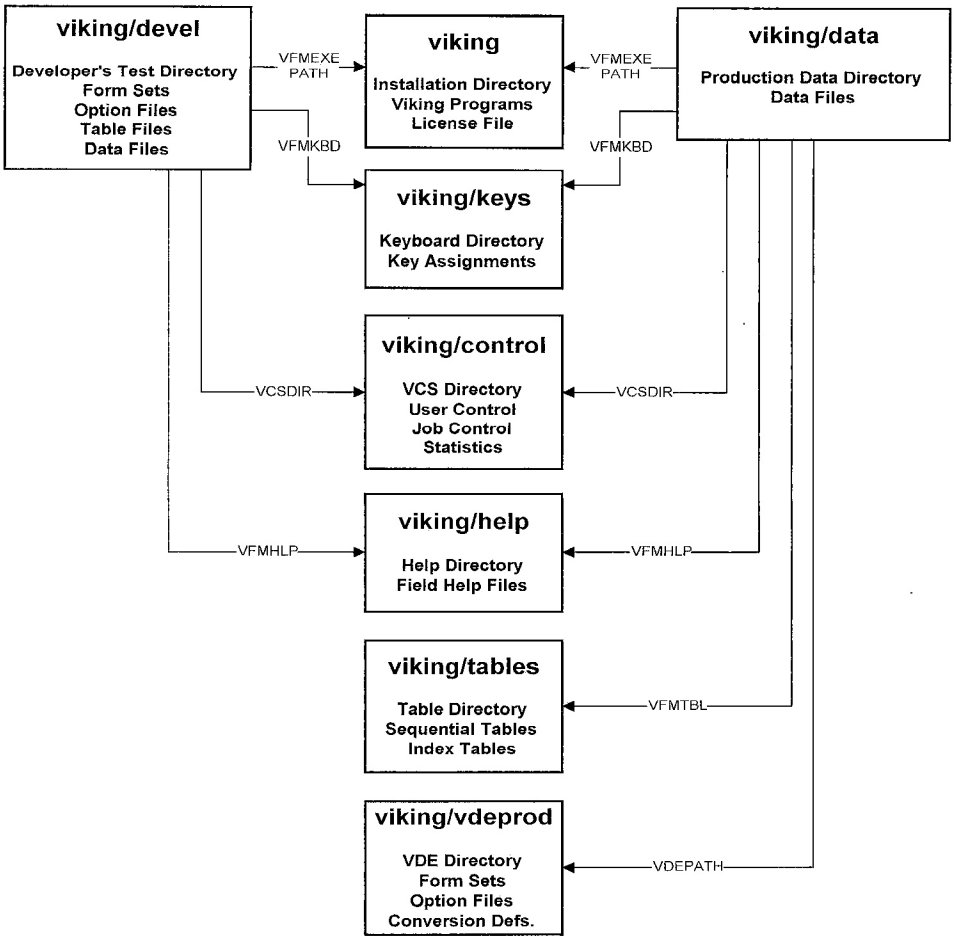
If you want VDE to produce operator statistics files and you are not running from the Viking Control System, **vcs**, set **VFMUSER** to an Operator ID (up to 4 characters).

KEY029

If the VDE Option File specifies an optional 029 Keyboard translation, this variable may be set to **Y** or **N** to eliminate the need for the operator to answer the **"USE 029 KEYBOARD?"** question.

The Unix Environment Variables are extremely useful at the time you install a new release of our software. Rather than installing the new release into your existing Viking directory, you can create a new directory. During the testing of a new release, you can point your version of the Unix Environment Variables to the new directories. Your production people can continue to use the previous release because their Environment Variables point to the old directories.

To switch your production users to the new release, you simply modify their Environment Variables to use the new directories. When you are comfortable that the new release is performing your tasks as it should, you can delete the old directories.



Unix Flow Chart

Unix Installation

The VDE and Portal systems are delivered on diskettes or tape along with a License Installation Sheet which displays your license information and your Password which is required for the installation to succeed. The VDE and Portal systems must be copied into a single directory on a hard disk. You should have 6MB of space available on the hard disk if you are installing the VDE or Portal systems. The instructions below assume that the installation is being done in the

directory **/usr/viking,** but you may create the base directory using any name you choose. You must have **root** privileges to install the Viking systems.

If you are installing Portal for the COBOL/XO language interface, you must not copy this interface into the same base directory. Each language interface consists of a set of libraries that are specifically built for that language. The languages C and COBOL do not have a common interface and must not be mixed. The language interface diskettes will be labeled and numbered separately from the Portal development tools diskettes.

The installation of VDE and Portal involves creating a new directory, copying the media into this directory, and executing the installation task, **vinstall.** You must have the License Installation Sheet available. The installation task, **vinstall,** will create your license file, **vlicense.vss,** in your default directory. The default directories will be created and the default files belonging in these new directories will be moved. Finally, a file named **vfmvars** will be created which sets the Environment Variables to the newly created directories.

1. Login to root.

2. Create a base directory for the VDE or Portal system, move into this directory and copy the media.

   # cd /usr [Enter]
   # mkdir viking [Enter]
   # cd viking [Enter]
   # tar xf /dev/dev name [Enter] {repeat if multiple diskettes}

3. Set file protections. These two commands must be entered in the order shown below:

   # chmod 755 * [Enter]
   # chmod 666 *.* [Enter]

4. The task, **vinstall,** will now allow you to enter the license information and set up the other directories. Before you can use this task, you must have your terminal identity, **TERM**, set to properly identify the terminal you are using. A list of supported terminal identities may be found in Chapter 2.

   # echo $TERM [Enter]
   "your identity"

** If the displayed identity is not correct then: **

 # TERM=new id [Enter]
 # export TERM [Enter]

5. Execute the **VINSTALL** task.

#./vinstall [Enter] ** Enter your license information as requested and modify the default directories if necessary. **

6. Assuming your installation completes successfully, you are now ready to set the Environment Variables. You must edit the default login **profile** script or each individual user's **profile** script and add the Environment Variable settings as found in the file **vfmvars.**

7. Testing Your Installation

 # exit [Enter] {exit from root}
login "your user id" [Enter]
password: "your password" [Enter]
** if you have not modified your profile

$ /usr/viking/vfmvars [Enter] {set Variables and execute vcs}

** if you modified your profile

$ vcs [Enter]

** Type in **DEMO** and press [Enter] and then press [Enter] again for the Password, you should be able to execute any task, use [Tab] to Exit the Menus, respond **Y** to Exit.

Setting Unix Environment Variables

Bourne Shell

 # VFMEXE=/usr/viking [Enter]
 # VFMUSER=xxxx [Enter] {where "xxxx" is your Operator ID}
 # export VFMEXE VFMUSER [Enter]

C Shell

 # setenv VFMEXE /usr/viking Enter
 # setenv VFMUSER MAG Enter

Unix Control System Options

The VCS Menus may be used to isolate your users from the operating system. For example, you may alter the user's profile by executing the VCS task, followed by the **exit** command. A user, upon logging into your system, will be presented with the VCS login procedure and when they Exit from VCS, they will Exit from Unix. If you prefer to bypass the VCS login, you may set the VCS user's password to blanks (no password required). You may also establish a time-out for VCS, which effectively would log out users that have not made a Menu selection in **"nn"** minutes.

Login Examples:

... set Environment Variables and export them
...

vcs	{require VCS login, no time-out}
-or-	
vcs -15	{require VCS login, time-out in 15 minutes}
-or-	
vcs -20 mag	{VCS login as MAG, time-out 20 minutes}
exit	{exit the login process}

Unix Differences and Restrictions

The **Viking Forms Compiler, VFC** command line options must be separated by commas instead of the slashes.

Unix Naming Conventions

File Name	Description
form_set_name.VFD	Form Set definition file output by **VFP**.
form_set_name.VLF	Loadable Form Set output by **VFC**.
form_set_name.o	Form Set Object output by **VFC.**
form_set_name.LIS	Form Set Print File output by **VFC**.
"name".TBL	Table files output by **VSSTABLE**.
form_set_name.HLP	Field Help file output by **VHELPMNT**.
form_set_name.HAS	Field Help ASCII file output by **VHELPASC**.
form_set_name.HLS	Field Help Print File output by **VHELPMNT**.
form_set_name.HBK	Previous version of Field Help renamed by **VHELPMNT**.
"name".KBD	Terminal Function Key Definitions output by **TERMKEYS**.
"name".LIS	Terminal Function Print File output by **TERMKEYS**.
form_set_name.VEO	VDE Options File output by **VDEOPT**.
form_set_name.BCK	Previous version of VDE Options File.
"name".con	File Conversion Definition file output by **CONDEF**.
"batch".cur	**VDE** opens this file for output. Indicates active batch.
"batch".dat	**VDE** data file name if not using name controls option.
"batch".trj	**VDE** temporary transaction journal file.
"batch".ds1	**VDE** data file in Create stage.
"batch".ds2	**VDE** data file in Verify stage.
"batch".ds3	**VDE** data file is completed.
"batch".ds4	**CONFILE** default output file.
"batch".ds5	**VDE** data file after conversion by **CONFILE.**
"batch".bak	**VDE** previous version of the data file.
"batch".sdf	**VDE** temporary file for Trail Verify version.
"name".vkd	**ISAMKEYS, ISAMLOAD** Key Definition files.
userfile.rpt	Listing of VCS users.
jobfile.rpt	Listing of VCS jobs.
jobstat.rpt	**VDE** Job Statistics.
userstat.rpt	**VDE** User Statistics.
batchsts.rpt	**VDE** Batch Statistics.

VAX and ALPHA VMS Operating System

VDE and Portal are available for the VAX and ALPHA platforms that use the VMS operating system. If you are not running VMS/V5.3 or greater, the Viking tasks may not execute. You must contact Viking's Customer Care group for instructions on rebuilding the system for your version of VMS.

VDE and Portal use the Function Key assignments as shown in Chapter 2. The **TERMKEYS** program can be used to customize the defaults contained in the file **TERMINAL.KBD** and create a new **TERMINAL.VSS** file with your customized assignments. You may create multiple **TERMINAL.VSS** files placed in separate directories and each user can set the Environment Variable, **VFMKBD**, to support the file they wish to use.

VMS Directories

The VDE and Portal systems are copied into a single directory. The installation program will create seven (7) sub-directories named **CONTROL, DATA, DEVEL, HELP, KEYS, TABLES,** and **VDEPROD.** Initial default files will be moved into the **CONTROL, KEYS** and **HELP** directories. Viking makes use of Environment Variables to locate your directories, giving you the flexibility of re-naming them if you wish.

A production system should never have **VFMKBD, VCSDIR, VFMHLP, VFMTBL,** and **VDEPATH** pointing into the distribution directory. The directory pointed to by **VDEPATH** must never contain Loadable Form Sets unless they are to be used by the VDE task.

The directory identified as **DEVEL** is not restricted to a single directory. Any number of individuals in any number of directories can act as a developer of Viking jobs. The **VMS Flow Chart** below illustrates that new jobs can be developed without affecting the current production jobs.

Likewise, the directory identified as **DATA** required to be a single directory. Each user may log into and remain in a unique directory to use the production system. If this is your choice, then please note that VDE will place the data files into the current users working directory unless you use the VDE Option Files to specify where the data files are to reside. This is an important consideration if one user is to key a file and another user is going to verify the file.

VMS Environment Variables

VFMEXE

The location of the Viking Programs must be set using the variable **VFMEXE.** These variables allow the user to execute the programs with simple commands and to locate all of the demonstration files. Your license file, **VLICENSE.VSS,** is also located with this variable.

VFMKBD

This variable locates the file **TERMINAL.VSS**. The Special Function Key assignments are carried in this file. All default Viking Programs find and load this file in order to execute. The default **TERMINAL.VSS** file is moved into this directory during installation.

VDEPATH

VDE will use the variable **VDEPATH** to locate the production Loadable Form Sets (**form set name.VLF**) and their Option Files (**form set name.VEO**) for an application. **CONFILE** also uses this variable to locate the Conversion Definition Files (**name.CON**).

VFMTBL

This variable is used by the Field Edit subroutines, **SEQTABLE** and **VIXTABLE,** to locate all Sequential and **ISAM** table files, see Chapter 7. **VIXTABLE** also uses this variable to locate the Loadable Form Sets used to maintain the **ISAM** files.

VFMHLP

This logical should point to the device and directory that contain the Field Help files (**form set name.HLP**). All Help files delivered with the system are moved into this directory during installation.

VCSDIR

This logical locates the VCS Control files, **VCSJOBS.DAT** and **VCSUSERS.DAT.** The VDE statistics file, **VDESTATS.DAT,** is also placed in this directory. Default **VCSUSERS.DAT** and **VCSJOBS.DAT** files are moved into this directory during installation.

VFMUSER

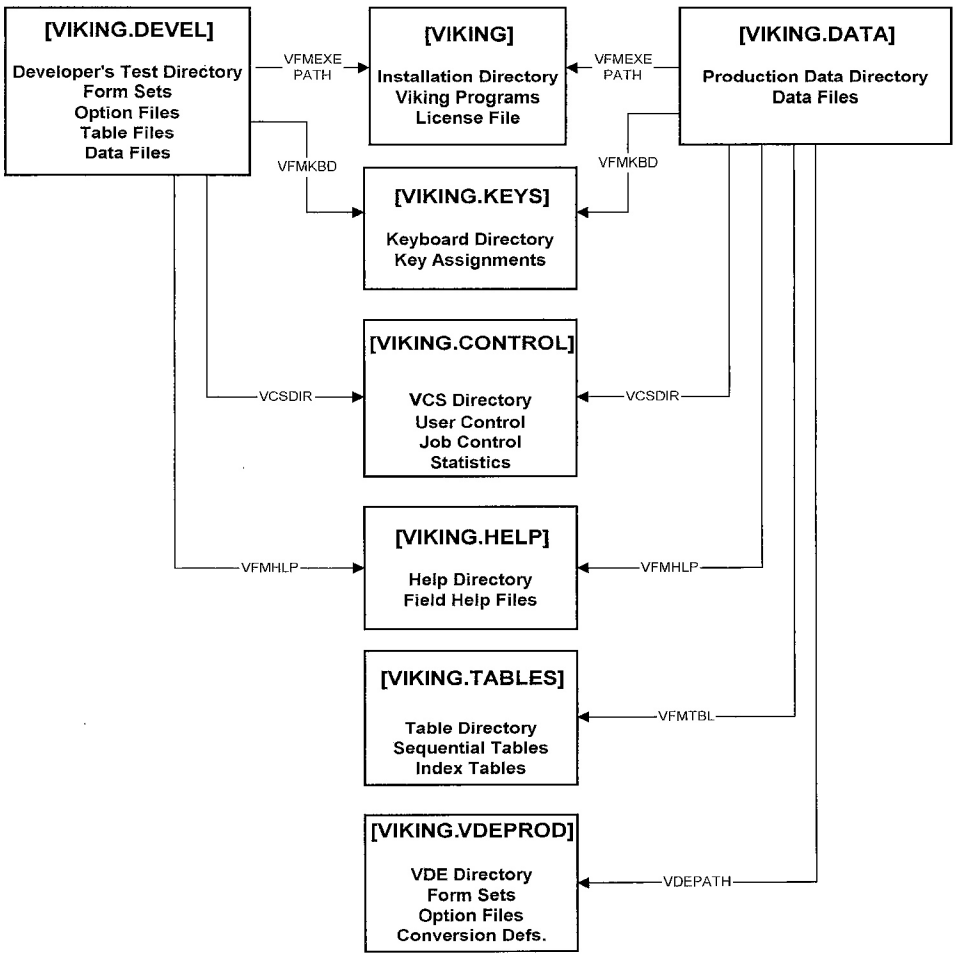
If you want VDE to produce operator statistics files and you are not running from the Viking Control System, **VCS**, set **VFMUSER** to an Operator ID (up to 4 characters).

KEY029

If the VDE Option File specifies an optional 029 Keyboard translation, this variable may be set to **Y** or **N** to eliminate the need for the operator to answer the **"USE 029 KEYBOARD?"** question.

The VMS Environment Variables are extremely useful at the time you install a new release of our software. Rather than installing the new release into your existing Viking directory, you can create a new directory. During the testing of a new release, you can point your version of the VMS Environment Variables to the new directories. Your production people can continue to use the previous release because their Environment Variables point to the old directories.

To switch your production users to the new release, you simply modify their Environment Variables to use the new directories. When you are comfortable that the new release is performing your tasks as it should, you can delete the old directories.



VMS Flow Chart

VMS Installation

The VDE and Portal systems are delivered on tape along with a License Installation Sheet which displays your license information and your Password, which is required for the installation to succeed. The VDE and Portal systems must be copied into a single directory on your hard disk. You should have 4MB of space available on the hard disk if you are installing the VDE or Portal systems. The instructions below assume that the installation is being done in a directory named **VIKING**, but you may create a base directory using any name you choose. You must have System privileges to install the Viking systems.

The Portal system on VMS will support all of the Digital languages that use their standard calling procedures. This includes C, Cobol, Fortran, Basic and Pascal. ADA does not use the standard calling procedures by default, but it can be directed to do so for the Portal calls.

The installation of VDE and Portal involves creating a new directory, copying the media into this directory, and executing the installation task, **VINSTALL**. You must have the License nstallation Sheet available. The installation task, **VINSTALL,** will create the license file, **VLICENSE.VSS,** in your default directory. The default directories will be created and the default files belonging in these new directories will be moved. Finally, a file named **VFMVARS.COM** will be created which sets the Viking Environment Variables to the newly created directories.

1. Log In with System Privileges.

2. Create a base directory for the VDE or Portal system, move into this directory and copy the media. You must have determined which disk to be used, **dev:**, and the device where the Viking media is to be copied from, **tape:.**

```
$ CREATE/DIR/PROT=(G:RE,W:RE) dev:[Viking] Enter
$ SET DEF dev:[Viking] Enter
$ MOUNT/FOREIGN tape: VFM Enter
$ BACKUP tape:VFM.BCK */NEW_VERSION Enter
$ DISMOUNT tape: Enter
```

3. Set file protections.

```
$ SET PROT=(G:RE,W:RE) *.* Enter
$ SET PROT=(G:RWE,W:RWE) *.DEF,*.HLP Enter
```

4. The task, **VINSTALL,** will now allow you to enter the license information and set up the other directories. Before you can use this task, you must have your terminal identity set to properly identify the type of terminal you are using. A list of supported terminal identities may be found in Chapter 2.

   $ SHOW TERM [Enter]
   -- if improperly identified --
   $ SET TERM/DEV=device_name [Enter]

5. Execute the **VINSTALL** task.

   $ RUN VINSTALL [Enter] ** Enter your license information as requested and modify the default directories if necessary.

6. Assuming your installation completes successfully, you are now ready to set the Environment Variables. You must edit the default login script or each individual user's **LOGIN.COM** script and add the Environment Variable settings as found in the file **VFMVARS.COM.**

7. Testing Your Installation

   $ LOGOUT [Enter] {exit from SYSTEM}
   Username: "your user id" [Enter]
   Password: "your password" [Enter]
   ** if you have not modified your **LOGIN.COM**
   $ @dev:[Viking]VFMVARS [Enter] (set Variables and VCS}
   ** if you modified your **LOGIN.COM**
   $ VCS [Enter]
   ** Type in **DEMO** and press [Enter] and then press [Enter] again for the Password, you should be able to execute any task, use [Tab] to Exit the Menus, respond **Y** to Exit

Setting VMS Environment Variables

VMS provides both the **ASSIGN** and **DEFINE** commands which can set the Viking Environment Variables. You may also set up Global symbols rather easily.

$ ASSIGN/NOLOG SYS$DISK:[Viking] VFMEXE
- or -
$ DEFINE/NOLOG VFMEXE SYS$DISK:[Viking]

$ VFMUSER :== xxxx {Define VDE operator symbol}
{where "xxxx" is the Operator ID}

VMS Control System Options

The VCS Menus may be used to isolate your users from the operating system. For example, you may alter the user's **LOGIN.COM** by executing the VCS task, followed by the **LOGOUT** command. A user, upon logging into your system, will be presented with the VCS login procedure and when they Exit from VCS, they will Exit from VMS. If you prefer to bypass the VCS login, you may set the VCS user's ID and/or Password on the VCS command line. You may also establish a time-out for VCS, which effectively would log out users that have not made a Menu selection in **"nn"** minutes. This must be done by altering the default **VCSCONTROL.COM** file. The variable **VCSTIMR** must be set to a non-zero number of minutes until time-out.

Login Examples:

... set Environment Variables
...
VCS {require VCS login, no time out}
-or-
VCS MAG {VCS user is MAG, ask for the Password}
-or-
VCS MAG SECRET {VCS user is MAG, Password is SECRET}

LOGOUT {exit the VMS system}

VMS Naming Conventions

File Name	Description
form_set_name.VFD	Form Set definition file output by **VFP**.
form_set_name.VLF	Loadable Form Set output by **VFC**.
form_set_name.OBJ	Form Set Object output by **VFC**.
form_set_name.LIS	Form Set Print File output by **VFC**.
"name".TBL	Table files output by **VSSTABLE**.
form_set_name.HLP	Field Help file output by **VHELPMNT**.
form_set_name.HAS	Field Help ASCII file output by **VHELPASC**.
form_set_name.HLS	Field Help Print File output by **VHELPMNT**.
form_set_name.HBK	Previous version of Field Help renamed by **VHELPMNT**.
"name".KBD	Terminal Function Key Definitions output by **TERMKEYS**.
"name".LIS	Terminal Function Print File output by **TERMKEYS**.
form_set_name.VEO	VDE Options File output by **VDEOPT**.
"name".CON	File Conversion Definition file output by **CONDEF**.
"batch".CUR	**VDE** opens this file for output. Indicates active batch.
"batch".DAT	**VDE** data file name if not using name controls option.
"batch".TRJ	**VDE** temporary transaction journal file.
"batch".DS1	**VDE** data file in Create stage.
"batch".DS2	**VDE** data file in Verify stage.
"batch".DS3	**VDE** data file is completed.
"batch".DS4	**CONFILE** default output file.
"batch".DS5	**VDE** data file after conversion by **CONFILE.**
"batch".SDF	**VDE** temporary file for Trail Verify version.
"name".VKD	**ISAMKEYS, ISAMLOAD** Key Definition files.
USERFILE.RPT	Listing of VCS users.
JOBFILE.RPT	Listing of VCS jobs.
JOBSTAT.RPT	**VDE** Job Statistics.
USERSTAT.RPT	**VDE** User Statistics.
BATCHSTS.RPT	**VDE** Batch Statistics.