

# Chapter 1

## Fundamentals of OpenVMS

### 1.1 Hardware

Covering all the hardware produced by DEC and third-party companies over the years would be a book of massive size and provide you with little information needed by an application developer. What an application developer really needs to be aware of is the terminals they will log in with. Occasionally, you will need to know how to hit the OFF-LINE button on a system printer, hit FORM FEED a few times, then hit OFF-LINE again to put the printer back on-line, but those buttons are clearly marked on all printers, so I think you can figure it out.

Many types of terminals exist. The DEC series of terminals are referred to as VT terminals for Video Terminal. (Some say it stands for Video Tube as all such devices were called “tubes”; others say it stands for Video Text but many of these terminals handle graphics.) Today, all standard terminal models have 3-digit model numbers. The 4-digit numbers are generally VXT terminals for multi-session X window type interfaces. The granddaddy of them all is the VT-52. This was a massive black and white (monochrome) terminal normally communicating at a whopping 150 or 300 baud working in uppercase all of its life. Yes, they could communicate faster, but the cable length had to be very short. There were older terminals and paper consoles (yes, a printer with a keyboard built into it), but the VT-52 seems to be when things were finally catching on. The VT-52 definition for the most part is the ANSI TTY definition with added formatting controls for cursor movement. Its replacement, the VT-100, became the ANSI standard terminal and is supported by a plethora of non-DEC operating systems. My first exposure to it was on a PDP-11/70 running RSTS/E. This was also my first exposure to DEC equipment.

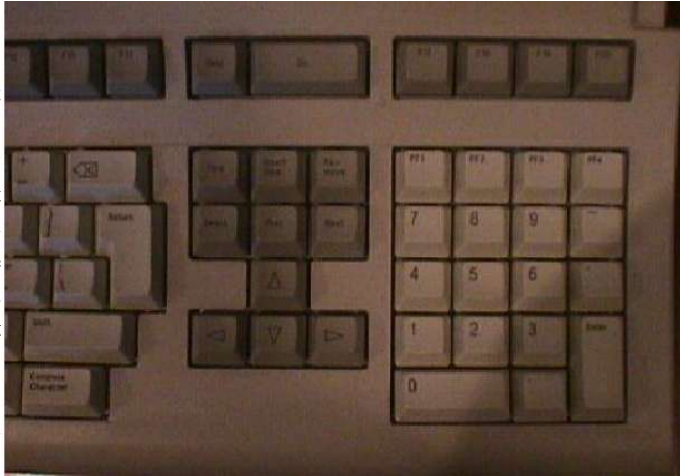
Today, regardless of the VT model, it will generally support VT-100 escape sequences and functionality. Some newer models only go back as far as VT-220, which was one of the first models supporting the enhanced keyboard used by later DEC terminals. Below is a picture of the DEC style keyboard attached to my Alpha machine.

Of special interest to us is the numeric keypad on the right hand side. When we get to the section on using the various editors available on OpenVMS, you will see that they are the primary interface.



Terminals are going the way of the abacus though. Most places today purchase a VT terminal emulation package for a personal computer. I use one almost exclusively on my notebook computer when traveling to client sites. The advantages of the emulator are many. Most notably you can communicate over the office network at full network speed. You can have multiple sessions opened on multiple machines and use your mouse to paste between terminal windows. Emulators also give us the ability to paste to/from PC-based applications into the terminal. Above all, you can customize the colors almost anyway you wish to make your computing experience enjoyable.

Notice the keyboard layout. The numeric keypad on the right has 4 PF keys across the top and 3 keys down the right side below them. PC-based keyboards are short one key in this area which can make for some interesting problems. The numeric keypad is how you navigate through the different text editors available on OpenVMS. When we get to the section on using the EDT version of the editor we will cover this in significant detail.



Another point worthy of note are the [HELP] and [DO] keys at the top. In the TPU version of the editor (and many OpenVMS applications), these two keys are used heavily. The keys [F1], [F2], [F3] and [F5] are quite significant. [F1] is the Hold key. This stops the terminal scroll. You hit it again to turn scrolling back on. It does not stop an application that is running or turn off output. This key is also mimicked by <ctrl><S> (hold) and <ctrl><Q> (end hold) combinations. [F2] is the screen print key. If your terminal has a printer attached to the back of it (or a network printer setup via the SETUP key), when you hit this key it causes the contents of the screen to be dumped to the printer. You can turn on a scrolling print by pressing <ctrl>[F2] and turn it off by hitting the same key combination again. [F3] is known as the SETUP key. When you hit it, one or more menus will pop up that you can navigate through setting all the possible configuration parameters for your terminal. There are too many to go into here and they vary terminal to terminal. Basically, all aspects of how the terminal communicates with the outside world and appears to you are controlled here.

[F5] is a key of special importance. Unless your terminal is hooked up to a LAT or is a multi-session enabled terminal, you should NEVER hit this key. When the terminal is hardwired as the console for many systems (central terminal used by operators on a special port on the machine) hitting the BREAK key can halt the entire system for all users. You will know you have done this when you see the following prompt: >>>. If you do not type RES and hit <return> before an internal time-out occurs on the box, you will have crashed the entire system for all users. Don' play with this key unless you know what you are doing. On old paper terminals hooked up as system consoles, the combination of <ctrl><p> used to duplicate this function. More than one operator went to hit <ctrl><o> to turn output off for something and accidentally hit <ctrl><p>.

The combination of <ctrl><o> is used to turn output off. This is not like “hold,” which simply stops the scrolling so you can read it. This combination routes the output to the bit bucket until the command interpreter regains control.

Most of the remaining function keys across the top are available for application usage. Their function will vary based upon what program you are running.

## 1.2 Logging In

The first terms you need to become familiar with are “Logging On” and its counterpart “Logging Off.” (Also called “Logging In” and Logging Out.) Logging In is the process of getting Logged In. (Sorry, I couldn' resist doing a Webster on you.) We will start with the easy scenario assuming you have a single session terminal hooked up via a serial connection. We will also assume that you have been assigned a user ID of FRED with a password of MY\_PASSWORD. (Neither user names nor passwords are case sensitive in the OpenVMS environment.) Your password will not be displayed while it is typed.

```
<return>
Username: FRED
Password:

                               Welcome to .yada yada yada
$
```

The “\$” is the default prompt on a OpenVMS system. Most system managers will have changed the prompt to be the name of the node you are logged into followed by “->.” So you see something like:

```
Kirk->
```

instead of the \$ prompt. To change the prompt so it shows your node name instead of the default \$, requires the use of a lexical function that we will talk about later on. If you are exploring on your own for a while, you can type HELP LEX at your command prompt to display the lexical functions available to you.

```
$ set prompt='f$getsysi("nodename")->
LGKL1->
```