Block-Mode Terminal Capability With VAX/VMS

Revised: 5/18/89
Security: Everyone

Block-Mode Terminal Capability With VAX/VMS

==========

This article last reviewed: 10 April 1989

This article describes why trying to do block-mode terminal emulation on a VAX running VMS is not a good idea. Essentially, it is a very complex, time-consuming process to convert software.

Having block-mode terminal capabilities on a VAX running VMS requires more than writing a terminal emulator. It also requires rewriting all of the programs that run on the VAX so that they support block-mode terminals. At the lowest level, VMS does support block-mode terminals. However, the lowest level only lets these features be implemented by an application. This requires the VMS application, itself, to implement all of the block-mode input and output routines that are usually accessible from the standard I/O routines on machines that run block-mode terminals.

Block-mode transfer is available from a specific model of DEC terminal in the VT100 family, the VT131. Currently, no terminal emulation package is available for the Macintosh that implements the block-mode features of the VT131. The alternative, then, is to write a terminal emulator with the necessary features.

Unfortunately, writing a terminal emulator that has block-mode features does not solve the problem on a VMS machine. The following quote from Digital's "Terminal and Printers Handbook" explains this:

"Digital currently supports the VT131 block mode under VAX/VMS version 3.0 and later versions. Support is at the operating system level, and this requires the use of third-party application programs to take full advantage of the VT131 features. This allows Digital software OEMs that use block mode terminals to use their software on Digital hardware with minimal modification. Digital has no plans to provide application software for the VT131 in other than VT102 (conversational) mode."

What this means is that although you could write a terminal emulator that has block-mode features, none of the standard VMS application software

would take advantage of those features. They would read the terminal in character-by-character (conversational) mode, anyway. Thus, to implement block mode, you would have to write the terminal emulator and rewrite every application on the system to support block mode. For example, the VMS editor would need to be completely rewritten.

The use of AppleTalk VMS does not solve this problem for the same reason that writing the terminal emulator will not. Standard VMS applications are still written for conversational mode. It is possible that performance might degrade in a situation where each character is transmitted in its own AppleTalk packet with the echoed character also returned in its own packet.

Tech Info Library Article Number:3763