New Technical Notes

Macintosh

®

Developer Support

MacWorkStation Q&As
Networking

Revised by: Developer Support Center	October 1992
Written by: Developer Support Center	October 1990

This Technical Note contains a collection of Q&As relating to a specific topic—questions you've sent the Developer Support Center (DSC) along with answers from the DSC engineers. While DSC engineers have checked the Q&A content for accuracy, the Q&A Technical Notes don't have the editing and organization of other Technical Notes. The Q&A function is to get new technical information and updates to you quickly, saving the polish for when the information migrates into reference manuals.

Q&As are now included with Technical Notes to make access to technical updates easier for you. If you have comments or suggestions about Q&A content or distribution, please let us know by sending an AppleLink to DEVFEEDBACK. Apple Partners may send technical questions about Q&A content to DEVSUPPORT for resolution.

Customizing MacWorkstation documents
Date Written: 4/25/89
Last reviewed: 11/8/90

Where should I put resources—in the MacWorkstation document or the MacWorkstation application file?

___

To maintain the generic nature of MacWorkstation, it is important that you do all customization is on MWS documents and not on the MWS application. Any resource added to an MWS application may produce incompatibility with other applications that use MWS. Also, future versions of MWS may support multiple sessions. Here are some of the things that go into MWS documents:

• Communication module code resources
• Host application's menu and dialog resources
• Any other host-specific resources such as icons, pictures, and data-strings
• Exec modules are stored in MWS documents as resources of type 'MWSX', along with a resource ID number for each module. The host program can load the module using either its resource ID number or its name.

How to make CCL "Debug" & "Matchstr" commands work correctly

Date Written: 4/25/89
Last reviewed: 11/8/90

The "Debug" and "Matchstr" commands in the Communication Command Language (CCL) don't work as documented.

___

MacWorkstation does not attempt to read from the communications module at any time other than during the "Wait" command, so if a script does not have the "Wait" command, neither the "Debug" nor the "Matchstr" command will work. Add the "Wait" command and it all works.

Restricting access permissions for the Macintosh host

Date Written: 4/25/89
Last reviewed: 11/8/90

Is there some way to restrict the access permissions for the host?

___

Currently there is no easy way to restrict access permissions for the host. The host can access any file on the Macintosh and create resources in the MacWorkstation documents. It would be possible, however, to write an EXEC which would prohibit certain commands from working under MWS, such as all file transfer commands.

Filtering out non-printing characters on the MacWorkstation end

Date Written: 4/25/89
Last reviewed: 11/8/90

How can I filter out non-printing characters on the Macintosh end of the connection?

___

No checking is done on the MacWorkStation (MWS) side to filter out non-printing characters. If you want to filter them out, you must write your own communication module. This could also be a function of the host application.

Using TextEdit for MacWorkstation "WYSIWYG" Exec

Date Written: 4/25/89
Last reviewed: 11/8/90

I am developing a MacWorkstation EXEC that will support a WYSIWYG display of a mainframe word processing document. Can I use TextEdit for display and editing?

___

You can use TextEdit, but you should know about some of its limitations. The latest TextEdit supports styles like bold, italic, and underlining, but does not support strike-thru (which can be done with QuickDraw).

The TextEdit package may give the impression that there is a full featured word processing system built in the ROM, but unfortunately, TextEdit is not up to the job of being a word processor, for several reasons. First, TextEdit is limited to 32,767 characters per record (the TELength is defined as an integer). A second more subtle limit is the drawing limit of the rectangles surrounding the text. The destRect and viewRect both surround the complete TextEdit record. Using some rather rough approximations, there is an upper limit of about 40 pages of text that can be supported in the QuickDraw rectangle. This is quite a lot for some applications, but not very many considering the job typically required of a word processor.

Another programmatic limitation is performance. TextEdit will become quite sluggish with large blocks of data. After 2,000-4,000 characters are stored in a TextEdit record, performance slows to an unacceptable level. It is notable that the lineStarts array is a linear array of offsets into the edit record. If the data towards the end of the data record (high in the record) changes, the offsets must be changed. This can involve updating thousands of integer offsets for every character typed. If the different font, size and style information is tacked on top of all that, the performance can be expected to suffer with large blocks of text. This is especially true in a MacWorkStation Exec. TextEdit was not designed to handle large documents. It was designed as a simple field editor for the Dialog Manager, and extended from there. It was never intended to handle the large jobs expected of a word processor.

In order to perform the operations required of a word processor it is necessary to use QuickDraw extensively. The expected Macintosh selection approach with autoscrolling, typing over selected text, cut/copy/paste, and so on are best implemented using QuickDraw directly. How the text is stored internally is the primary determining factor on how the word processor will perform.