MEMDUMP.MAC

A Command Language Macro Application Note

OVERVIEW

SourcePoint's powerful command language is extremely useful for automating tasks. It uses an interpreted scripting language that is very similar to the common C programming language. MemDump.MAC is an example of a SourcePoint™ command language macro. Its purpose is save a 64k memory dump to a file using the flist and nolist commands. It also demonstrates use of the argcount[ ] and argvector[ ] functions.

THEORY OF OPERATION

MemDump contains a single procedure called mdump( ) that can be called with two optional parameters to dump a portion of memory to a file named 'MemDump.txt'.

Argcount[ ] and argvector[ ] can be used when it is necessary for a procedure to have one or more optional parameters. A procedure can test argcount[ ] to determine the number of parameters it was called with. After testing the argcount[ ] function to determine if parameters are present, the procedure can call the argvector[ ] function to retrieve each of the parameters. An example of this can be seen by examining MemDump.MAC with a text editor.

The macro is loaded from within SourcePoint by clicking on: **File|Macro|Load Macro**

It is then invoked from the command window by entering the following at the command prompt:

mdump<cr>
mdump(Segment)<cr>
mdump(Segment, Length)<cr>

Where:

Segment is the real mode segment where the memory dump is to start (default = 0000h)
Length is the number of bytes that are to be dumped starting at offset 0h (default = 0100h)