JUST ASK!

Question: What's a little space between commands?

Editor: One of the many powerful features of the M language is that it allows for commands to be shortened into a single letter, which speeds up the writing of M code. While some commands have no arguments, others may consist of the command name followed by a space and one or more arguments, separated mainly by commas. These same features, however, may cause even experienced M programmers some confusion.[1] One command that clearly exemplifies this is the QUIT command. You will often see this written Q for short.

QUIT is used to signal the end of execution of a given code segment. The introduction of the extremely powerful extrinsic functions to the language in the late 1980s has added some confusion. Previous to this, no arguments were allowed with the QUIT command. A reference to an extrinsic function starts with two dollar signs followed by the name of the function and a list of parameters enclosed in parentheses, such as `$$FUNCTION(PARAMETERS)`.[2] Functions return a single value that is appended as an argument following the QUIT command exiting the function. This is the only case in which the QUIT command has an argument. The syntax in this case is QUIT ARGUMENT.

The following example helps to illustrate what may happen by simply missing a single space after a QUIT command. What happens when the following lines of M code are executed?

```
Q:'OK D
.W!,"OK to execute the
structured block"
```

The intent is to execute the block-structured DO if OK is true. At first glance one might think when the above lines of code are executed, they will yield an error since a blank is missing after the conditional QUIT command. Upon executing the above two lines, however, M will not error out. Instead, if OK is true, M will skip the block-structured DO with the WRITE (W) command. On a closer look, it becomes apparent that M interprets the conditional QUIT followed by a single space and a variable as a function QUIT with the variable "D" as its argument. Since the structured block is not preceded by a DO command, it will not be executed. Hence, had the structured block consisted of a more crucial set of commands than a single WRITE command, such as a series of SETs and KILLs to elements of a database, as simple a mistake as a missing blank may yield a series of unexplained events and a major discrepancy in the overall database without ever giving any signs, warnings, or errors. Indeed, such a problem will go undetected by M and the error-checker utility since it is valid syntactically.

The above case stands true whether the lines of code were part of a subroutine or a function structure. On the other hand, if OK is false, the outcome will differ in the case of a subroutine versus a function. In a subroutine context, an error will occur since the code is attempting to quit with an argument. In a function structure, however, the value of the variable "D" will be returned if "D" is defined, or the code will yield an error if "D" is undefined.

The M language provides many features that make it unique as a programming language. Some of these features add to the strength of the language, but at the same time may introduce an element of confusion. Eliminating such confusion relies heavily on the skill and expertise of the programmers and requires thorough testing procedures.

Diana Mohsene, Antrim Corp., contributed the substance of this *Just Ask!* column.

[1] R.F. Walters, *ABCs of MUMPS* (Marlboro, Massachusetts: Digital Equipment Corporation), 1989.

[2] J.M. Lewkowicz, *The Complete MUMPS* (Englewood Cliffs, New Jersey: Prentice-Hall, Inc.), 1989.