JUST ASK!

Question: We are moving our system to another platform. We have found a problem with our device handling. On our current system, we use a string to identify our device that looks like `internal^external`, where `internal` is the `$IO` value for the device and `external` is our own identifier. Anytime we try to `OPEN` or `USE` a device using this format under the new system we get an error.

Editors: What you see is the consequence of two mistakes. Your current vendor made the first mistake; you (or whoever wrote your software) made the second. The standard (ANSI X11.1 - 1990) states that the arguments of the `OPEN`, `USE`, and `CLOSE` commands are expr (expressions that evaluate to a string). During the 1970s and early 1980s, the most common implementations of M used device identifiers (the value that you saw in `$IO`) that were numeric. The standard certainly allowed this—any numeric is also a string.

What you have with your current system, however, is something a bit different. Your implementation is actually doing a type coercion on the arguments of the `OPEN`, `USE`, and `CLOSE` commands. The system is automatically converting any string to an integer before proceeding with the actual command. In effect, the implementor is treating the argument of these commands as an intexpr rather than an expr.

Here is another way to look at the problem. The standard requires that the following code fragment will make `$TEST=1`:

```
OPEN DEV USE DEV IF $IO =DEV
```

In your case, the above code fragment would fail on the `OPEN` command using your `internal^external` format because the device is trying to use the whole string as the device identifier. In your current implementation, the code fragment could make the value of `$TEST` false if you used your `internal^external` value for `DEV`. (In both cases we are assuming that the actual device can be `OPEN`ed.)

This is the long way of saying that your current implementation does not follow the standard. Even though the implementation is nonstandard, you may still not have run into a problem if your application developers had not made a mistake: They also did not follow the standard. Your code depends on the implementation doing a type coercion to integer for the argument of the `OPEN`, `USE`, and `CLOSE` commands.

If your application code followed the standard, the above code fragment would always produce a `$TEST=1`, even on your current implementation, because you would not use your `internal^external` format and assume type coercion. The requirements described here were in all previous versions of the ANSI X11.1 standard and are also in the proposed new standard, currently in canvass. M

Please address your *Just Ask!* questions or requests to the managing editor at *M Computing*.

An Invitation for Authors and Would-Be Authors

M Computing editors welcome articles and news items from our readers. Tell us about M application areas, new systems and installations, interfacing, techniques, technology challenges, or something else related to the world of M. Send book reviews, product announcements, press releases, other materials, or a brief description of your proposed article to Marsha Ogden, Managing Editor, MTA, Suite 205, 1738 Elton Road, Silver Spring, Maryland 20903-1725. Phone: 301-431-4070; Fax: 301-431-0017; or use FORUM.

JUST TWO WRITING OPPORTUNITIES LEFT IN 1994!
BOOT UP YOUR WORD PROCESSOR!
September Focus: *Interoperability*
Deadline: July 1, 1994
November Focus: *Client/Server Technologies*
Deadline: August 17, 1994