
Subject: Re: Troubleshooting - Error Messages.
Posted by [Steve Cox](#) on Wed, 01 Dec 1999 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Vicky A wrote:

- > Is that what it boils down to, IDL tells you Where the error is, and
- > you work out What the error is, by recognition or something?

As others have said, that pretty much IS what it boils down to. The good thing is, knowing the location of the error is all you need most of the time. And putting a comma after 'print' will become second nature to such an extent that it will screw you up when you use 'print' in some other language that doesn't use the comma.

A couple of tips for avoiding common errors in the first place:

Indent your loops and blocks, and leave enough white space so that the code is readable. If you've got statements with tons of parentheses, break them up onto separate lines (using the \$ continuation marker). Then count parentheses a couple of times anyway, and check the statement from the inside out to see if it makes sense. DON'T do what is very tempting, and just throw in or take out parentheses until it compiles. It's very easy to have a statement that compiles, but doesn't do what you want.

When you start a loop or block, write the terminator (ENDFOR, ENDIF, etc.) immediately, and then fill in the middle. Similarly if you're modifying existing code and removing a loop, remove the FOR and ENDFOR at the same time. The 'End of file encountered before end of program' message is one of the harder ones to debug.

If you've got some sequence of commands that you find yourself typing or copying into every procedure you write, consider making a routine out of that sequence. Your code will be shorter and more readable, and you won't have to worry about screwing that section up.

- > I find it really frustrating that I have to keep going back to the
- > book/webpages for every little typing mistake....

You'll quickly stop making most of the mistakes you're making now, and will recognize the other ones without having to go to the manuals.

I keep making this mistake though: when I use a keyword, I sometimes nonsensically put a slash in front of it, then assign it a value anyway, as in:

FOO,/BAR=bar

This is wrong, I know why it's wrong, the compiler won't allow it, and it's hardwired into my fingers anyway. All I can do is correct it each time.

-Steve Cox
