
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

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

Vicky A (vicky_a@my-deja.com) writes:

- > Is there some standard procedure for dealing with compilation errors?
- > (I mean errors that prevent a file compiling)
- >
- > I know an experienced user can just look at
- > print n
- > and say/think 'There should be a comma there.'
- >
- > 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?

I'm afraid, Vicky, that is what it boils down to:
experience. :-)

But I'm here to tell you that after you have fixed
a thousand no-comma problems, you do tend to recognize them
fairly quickly. :-)

But the fact you are asking questions is incredibly
encouraging. I see all types in my IDL programming
classes, but I especially remember one person who
HATED to get help. He wanted to do EVERYTHING himself,
including make his own mistakes. I try to read what
people want, so although I saw him struggling with
his programming exercise and I offered to help a
couple of times, he wasn't having it. He struggled
for about an hour and a half, refusing to give up.
The class and I moved on to another topic, which
he completely missed. Finally, he admitted defeat
and asked for my help.

"You're missing a comma here," I said, smiling.

That was it. One comma that I noticed was missing

one and a half hours ago and was begging to tell him,
if only he had allowed me to.

I'm all for struggling. It is really the only way
to gain experience. I try to make sure everyone in my
classes struggles a bit. But one and a half hours over
a comma is WAY too much. (Although I'll bet it's the
first thing he looks for if we wasn't so discouraged
that he gave up IDL programming completely.) Ask for help.

I solve about half the problems people have in classes
just by coming over and looking over their shoulders.
The problems just go away. No kidding. :-)

After five minutes, go find somebody to look over
your shoulder. That's my best advice. :-)

Good programming!

David

--

David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Toll-Free IDL Book Orders: 1-888-461-0155

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

I think that's a good assessment of what programmers do. That's why we
get paid the big bucks ;^}

One (simple) thing that may help is to make sure to indent correctly.
This helps keep track of FOR's without ENDFOR's, etc. I use xemacs in
idlwave mode
that automatically highlights and indents for me.

Vicky A wrote:

> snip.. snip...
>
> 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?
>
