
Subject: Re: basic IDL questions: please help!
Posted by [James Kuyper](#) on Mon, 21 Oct 2002 23:55:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

"B.G." wrote:

>
> Hi, Thanks for reading my question.
>
> I start a program with: `pro program_name`
> then I've got lots of stuff inbetween that and the "end" at the end of
> the program. After the program quits and returns me to the command
> prompt of IDL, it seems like it has lost all the variables from it's
> memory and I can't work with the variables anymore. I can fix this

That's correct. Variables created in that procedure are local to that procedure. They disappear when the procedure is complete. This is generally a good thing; it allows you to create temporary stuff and not have to worry about getting rid of it. If that's not what you want, you need to make the outputs arguments of the procedure. Thus:

In your procedure:

```
PRO procedure_name, output_array  
  output_array = indgen(10)  
END
```

At the IDL command line:

```
IDL> procedure_name, my_name
```

Inside your procedure, `output_array` becomes a reference to 'my_name'. If you don't already have a variable named `my_name`, it gets created when procedure name creates `output_array`. If `my_name` already exists, it gets replaced by the new value. Either way, when you're done, `my_name` contains the output from `indgen(10)`.

...

> Also, if I call a program while inside another program how do I get
> IDL to automatically recompile the inner program (or at least check to
> make sure it doesn't need to be recompiled) when IDL compiles the
> "outer" program?

When you call a routine that has not yet been compiled, IDL searches for a file with that same name and the extension '.pro'. If it finds that file, and finds an appropriate definition of the routine you've specified, it will compile everything in that file up to and including that routine, but nothing that appears after it.

Therefore, if you want to take advantage of this feature, every routine

you call should either appear in the same file, before the function or procedure that calls it, or should be in a separate file with the same name as the routine and an extension of '.pro'.

Subject: Re: basic IDL questions: please help!

Posted by [R.G. Stockwell](#) on Tue, 22 Oct 2002 12:59:12 GMT

[View Forum Message](#) <> [Reply to Message](#)

B.G. wrote:

- > Hi, Thanks for reading my question.
- >
- > I start a program with: `pro program_name`
- > then I've got lots of stuff inbetween that and the "end" at the end of
- > the program. After the program quits and returns me to the command
- > prompt of IDL, it seems like it has lost all the variables from it's
- > memory and I can't work with the variables anymore. I can fix this
- > problem with a "stop" before the "end", but Dave's `mpi_plot` program
- > won't seem to work correctly unless it sees an "end" and not a "stop".

You can "fix" this "problem" by running the code at the main level.
You do this by removing (comment out) the "pro program_name" line.
(keep the "end" at the end of the file).

To learn about "scope", search your idl help for "disappearing variables".
That doesn't acutally tell you much, but a quick google search gives us
this page that may help you.

<http://people.cs.vt.edu/~kafura/cs2704/scope.html>

- > I know this is probably pretty rudimentary, but it's always had me
- > confused and I'm hoping someone could help me out here by explaining
- > what's going on.
- >
- >
- > Also, if I call a program while inside another program how do I get
- > IDL to automatically recompile the inner program (or at least check to
- > make sure it doesn't need to be recompiled) when IDL compiles the
- > "outer" program?

If the program is already in memory, and you make changes to it, you will have to explicitly compile the code. If the program is not in memory (and idl can find it by seaching through the paths), it will load the program and compile it. IDL does this for speed reasons, since using code in memory is way faster than reading it from file every time it is called.

Cheers,
bob
