
Subject: Re: what am I doing wrong?

Posted by [thompson](#) on Tue, 06 Oct 1998 07:00:00 GMT

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

Bruce Bowler <bbowler@bigelow.org> writes:

> William Thompson wrote:

>>

>> Bruce Bowler <bbowler@bigelow.org> writes:

>>

> (stuff deleted)

>>> if I run it as follows...

>>

>>> # idl

>>> idl> .run 6953

>>

>>> it works fine...

>>

>>> if I run it like

>>

>>> # idl 6953

>>

>>> the results are

>>

>> (stuff deleted)

>>

>>> unfortunately, because of the way I need to interact with some other

>>> stuff, I really need to run it the second way. What can I do to make it

>>> work like the first way? I don't think precompiling will work either as

>>> I generate the file "on the fly".

>>

>> If you're using Unix, then what you need to do is to create a file like the

>> following:

>>

>> #!/bin/csh

>> setenv MY_IDL_SCRIPT \$argv

>> idl << done

>> .run \$MY_IDL_SCRIPT

>> done

>>

>> Use "chmod +x" to make the file executable, and then you can use this to

>> execute whatever IDL script you want.

> Unfortunately, as I said before, that won't work in my case. I am

> passing this to a third party package that invokes it as

> idl xxx

> I have no control over that.

There are a couple of ways you can accomplish this. One way, as already pointed out by Stein Vidar, is to end each line within a loop with the characters "& \$", so that IDL treats it all as one line. As I recall, this works unless the loop is very large.

Another way is to use two procedure files. One file will contain the main-level program that you wish to run. For example, let's say you save it in a file called myprogram.idl. The second file will then contain the single line

```
.run myprogram
```

You can then pass the name of this second file to IDL. For example, if the second file is named run_myprogram.pro, then the statement

```
idl run_myprogram
```

will execute the main-level procedure myprogram. It would be just the same as if you had typed ".run myprogram" at the command prompt.

William Thompson
