
Subject: Re: IDL, scripts and command line arguments?
Posted by [Michael Zingale](#) on Fri, 29 Mar 2002 18:19:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

The following procedure works for me. Create a shell script called test_script, containing:

```
#!/bin/csh -f

# simple wrapper for IDL to take 3 arguments (integer, real, and string)
# and pass them to the IDL function idl_pass

if ($#argv != 3) then
    echo usage: test_script int float \"string\"
endif

set tmpfile = tmp.$$

echo idl_pass, $1, $2, \"${3}\" >> ${tmpfile}

idl ${tmpfile}

rm -f ${tmpfile}

#end
```

also create an IDL procedure called idl_pass.pro containing:

```
pro idl_pass, int_pass, float_pass, string_pass

print, 'integer = ', int_pass
print, 'float = ', float_pass
print, 'string = ', string_pass

end
```

make test_script executable, and type

```
./test_script 1 2.4 "this is a test"
```

and IDL will be launched with the 3 arguments passed to the idl_pass procedure, and the following will be output to the screen:

```
% Compiled module: IDL_PASS.
integer =      1
```

```
float = 2.40000
string = this is a test
IDL>
```

Mike Zingale
zingaleATucolick.org

tbowers@nrlssc.navy.mil wrote:

```
> Mike, if I understand you correctly...
> If I have an IDL app that I want to be able to pass parameters to from
> the command line, I:
>
> 1) write my parameters that i wanna pass to a params.txt file, 1
> parameter per line. e.g, if my code takes 2 parameters, params.txt
> will read:
> ./indata/file1.hdf
> ./outdata/outfile1.hdf
> For what you want, passing from a command shell I think, this params
> file is written to with a command script.
>
> 2) I write a wrapper routine (e.g. wrapper.pro) who's job is to read
> in the params file then call my app with the appropriate params.
>
> 3) Open IDL with this wrapper.pro, compile it, resolve_all, save,
> /routines, filename='whatever.sav'
>
> Now, I can run from a command shell. Run the script that that'll write
> your params to params.txt. In the script call idl runtime. On unix,
> idl -rt=whatever.sav. On win, idlrt whatever.sav
>
> Of course, if your parameters never change, or you just wanna change
> them yourself with a text editor, you don't need to bother writing a
> shell script. I keep mine general b/c many times i'm processing files
> automatically at a certain time of day and i use a script to query an
> input dir to see when new files arrive (script grabs by ftp) for
> processing. When a new file arrives, regardless of it's name, the code
> runs automagically.
>
> Hope this helps,
> todd
>
> mmiller3@iupui.edu (Michael A. Miller) wrote in message
news:<877knxzmvp.fsf@lumen.indyrad.iupui.edu>...
```

>> Does anyone have any suggestions on ways to trick IDL into
>> getting access to command line arguments (argv- and argc-like
>> functionality)? The only trick I've come up with so far is to
>> write a script in <scripting-language-of-choice> that writes an
>> idl procedure with the parameters hard coded into it and then
>> executes it. It's not very pretty though and I wonder if there
>> are any other possibilities...
>>
>> Mike
>
