
Subject: Re: IDL, scripts and command line arguments?
Posted by [David Fanning](#) on Thu, 28 Mar 2002 01:20:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

Michael A. Miller (mmiller3@iupui.edu) writes:

> 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...

I think that about sums up the possibilities, really. :-)

Cheers,

David

--

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

Subject: Re: IDL, scripts and command line arguments?
Posted by [Mark Hadfield](#) on Thu, 28 Mar 2002 01:56:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.170c1ddc7a5616b8989855@news.frii.com...

> Michael A. Miller (mmiller3@iupui.edu) writes:
>

>> 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...

>

> I think that about sums up the possibilities, really. :-)

Come on, David! There are lots of other possibilities. Unfortunately

most of them are even uglier than writing .pro files on the fly. :-)

One Unix you can send information to console-mode IDL via stdin. Eg. typing this at a shell prompt...

```
echo "plot, findgen(11) & wait, 1" | idl
```

...shows a plot window for a second.

--

Mark Hadfield

m.hadfield@niwa.co.nz

Ka puwaha et tai nei

http://katipo.niwa.co.nz/~hadfield Hoea tatou

National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: IDL, scripts and command line arguments?

Posted by [David Fanning](#) on Thu, 28 Mar 2002 02:12:46 GMT

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

Mark Hadfield (m.hadfield@niwa.co.nz) writes:

```
> Come on, David! There are lots of other possibilities. Unfortunately
> most of them are even uglier than writing .pro files on the fly. :-)
>
> One Unix you can send information to console-mode IDL via
> stdin. Eg. typing this at a shell prompt...
>
> echo "plot, findgen(11) & wait, 1" | idl
>
> ...shows a plot window for a second.
```

Neat! I hadn't thought of that. :^)

Cheers,

David

--

David W. Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438, E-mail: david@dfanning.com

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: IDL, scripts and command line arguments?

Posted by [T Bowers](#) on Fri, 29 Mar 2002 15:29:15 GMT

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
```

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
>

Subject: Re: IDL, scripts and command line arguments?
Posted by [Ken Mankoff](#) on Sun, 31 Mar 2002 17:22:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 27 Mar 2002, Michael A. Miller wrote:

> 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...

That is how I managed to do it, but I like to think of it as
'very purty' as opposed to 'not very pretty'.

Here is my script. I name it "idl_argv.pl", and here are some examples
of how to call it (from the unix prompt):

```
./idl_argv.pl  
./idl_argv.pl 's = "a silly string"  
./idl_argv.pl 'for i=0,10 do print, i'  
./idl_argv.pl 'foo=42 & help, foo & plot, indgen(foo)'  
./idl_argv.pl 'pi=3.14159D & almost_pi = 4 ; spaces and comments!'
```

Here is the script:

```
-----  
#!/usr/bin/perl  
  
$input = $ARGV[0];  
@list = split( /&/, $input );  
$uniq = $$;  
  
open( IDL_FILE, ">idl.$$pro" );  
for ($i=0; $i<=$#list; $i++)  
{ print IDL_FILE "$list[$i]\n"; }  
close( IDL_FILE );
```

```
system( "idl idl.$$pro" );  
system( "/bin/rm idl.$$pro" );  
exit(0);  
-----
```

Is this even close to what you were trying to do?

-k.

--

Kenneth Mankoff
LASP://303.492.3264
<http://lasp.colorado.edu/~mankoff/>
<http://lasp.colorado.edu/snoe/>
<http://lasp.colorado.edu/marsrobot/>

Subject: Re: IDL, scripts and command line arguments?
Posted by [Ken Mankoff](#) on Sun, 31 Mar 2002 19:59:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Sorry to reply to my own post, but I found a bug.
My code allows you to issue an IDL command like this:

```
IDL> a = 42 ; comment & b = 43
```

And this is technically outside the scope of IDL, as *everything*
after a ; is supposed to be ignored as a comment. So, below is the
revised (and much shorter) idl_argv.pl script:

```
#!/usr/bin/perl  
open( IDL_FILE, ">idl.$$pro" );  
print IDL_FILE "$ARGV[0]\n";  
close( IDL_FILE );  
system( "idl idl.$$pro" );  
system( "/bin/rm idl.$$pro" );  
exit(0);
```
