

---

Subject: Re: IDL batch mode: Command line args?

Posted by [mgs](#) on Tue, 31 Mar 1998 08:00:00 GMT

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

---

In article <MPG.f8b2018b0067093989770@news.frii.com>, davidf@dfanning.com (David Fanning) wrote:

> Mark Elliott (mark@mail.mmrrcc.upenn.edu) writes:

>

>> I'd like to run IDL (from UNIX) in batch mode with a command

>> like:

>>

>> unixprompt> idl batchprog.pro

>>

>> How can I provide additional arguments to be used by the batch  
>> program. For example, is there a syntax like:

>>

>> unixprompt> idl batchprog.pro inputfile.txt

>> unixprompt> idl batchprog.pro 5 3.78 "John Doe" 24.0

>

> As far as I know, this can't be done the way you want to  
> do it. You do have one other option, however. You can define  
> environment variables for your input parameters. Then your  
> batchprog.pro program calls GETENV to get the values of  
> the environment variables before passing these onto the other  
> programs that you want to run in batch mode.

Here's a variation along those lines. You could set the environment variable IDL\_STARTUP to point to different batch programs, then call IDL. This would start IDL and run the batch file pointed to by IDL\_STARTUP. Similarly, you could keep IDL\_STARTUP pointing to the same file, but create that file using your UNIX program (shell, perl, etc.). Get the help of your local UNIX shell programming fanatic and you might end up with something pretty entertaining. I was doing stuff like this a couple years back. I could dig up some ideas from those efforts if you're interested.

...

> Cheers,

>

> David

--

Mike Schienle  
mgs@sd.cybernex.net

Interactive Visuals  
<http://ww2.sd.cybernex.net/~mgs/>

---

---

Subject: Re: IDL batch mode: Command line args?

Posted by [davidf](#) on Tue, 31 Mar 1998 08:00:00 GMT

Mark Elliott (mark@mail.mmrrcc.upenn.edu) writes:

```
> I'd like to run IDL (from UNIX) in batch mode with a command
> like:
>
> unixprompt> idl batchprog.pro
>
> How can I provide additional arguments to be used by the batch
> program. For example, is there a syntax like:
>
> unixprompt> idl batchprog.pro inputfile.txt
>
> or
>
> unixprompt> idl batchprog.pro 5 3.78 "John Doe" 24.0
```

As far as I know, this can't be done the way you want to do it. You do have one other option, however. You can define environment variables for your input parameters. Then your batchprog.pro program calls GETENV to get the values of the environment variables before passing these onto the other programs that you want to run in batch mode.

Naturally, the return value from GETENV is a string, so you will have to take care to convert the value to a number, if that is what you want, etc. I've never tried it myself, but I think it is possible.

Cheers,

David

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

---

---

Subject: Re: IDL batch mode: Command line args?  
Posted by [Martin Schultz](#) on Wed, 01 Apr 1998 08:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Mark Elliott wrote:

```
>
> I'd like to run IDL (from UNIX) in batch mode with a command
```

> like:  
>  
> unixprompt> idl batchprog.pro  
>  
> How can I provide additional arguments to be used by the batch  
> program. For example, is there a syntax like:  
>  
> unixprompt> idl batchprog.pro inputfile.txt  
>  
> or  
>  
> unixprompt> idl batchprog.pro 5 3.78 "John Doe" 24.0  
>  
> If this can be done, how are the additional args obtainable within the  
> batch program? I can find no documentation that this is possible.  
> Still I can always hope...  
>  
> Thanks.

Hi Mark,

your question intrigued me, and I came up with a solution involving a temporary file that is written by a little Unix shell script and contains the arguments passed to IDL. This is certainly not perfect, and I haven't spent any time on error checking etc., but it may provide a start. I must admit that I have never tried to execute IDL in batch mode before, and my 2-minute primer on this led to the conclusion that you can only run main programs this way and that you have to provide an EXIT command in order to leave IDL after your program has terminated. If this is not true, I'd be happy to learn more!

Now, here is how it works:  
the shellscript idlbatch (make sure it's executable):

```
# Unix shell script to pass arguments into idl
# when running in batch mode
# uses echo to store the command line arguments in a
# temporary file which can be read by batchtest.pro
# Example: idlbatch batchtest 5 8
echo $* > tmpfile
idl $1
```

Your main program (see batchtest.pro as an example) will have to read in 'tmpfile' and extract the arguments: the first argument is the name of the program itself. In batchtest.pro I simply call a second procedure batchtest2.pro (also attached below), which multiplies the first two arguments (defaulted by a value of 3). Certainly, there is

lot of room for improvements on the parameter parsing and type determination - who knows, maybe David is interested ?

Best regards,  
Martin.

-----  
Dr. Martin Schultz  
Department for Earth&Planetary Sciences, Harvard University  
186 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318  
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu  
IDL-homepage: <http://www-as.harvard.edu/people/staff/mgs/idl/>  
-----

```
; simple IDL test program to be run in batch mode  
; on the Unix xshell type idlbatch batchtest [arguments]
```

```
print,"I'm in !"  
; read tmpfile produced from the shell script  
args = "  
openr,ilun,'tmpfile',./get_lun  
readf,ilun,args  
free_lun,ilun
```

```
; for now we assume all arguments are numeric  
; except for first one which is the name of the IDL program  
argv = str_sep(args, ' )  
argv = float(argv(1:*))  
if (n_elements(argv) ge 2) then $  
    batchtest2,argv(0),argv(1) $  
else if(n_elements(argv) ge 1) then $  
    batchtest2,argv(0) $  
else $  
    batchtest2
```

```
exit,/no_confirm,status=0
```

```
pro batchtest2,a,b
```

```
if (n_elements(a) eq 0) then a=3
if (n_elements(b) eq 0) then b=3
```

```
print,'A*B=',a*b
```

```
return
end
```

## File Attachments

---

- 1) [batchtest.pro](#), downloaded 121 times
  - 2) [batchtest2.pro](#), downloaded 116 times
- 

---

Subject: Re: IDL batch mode: Command line args?  
Posted by [manizade](#) on Wed, 01 Apr 1998 08:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

> Mark Elliott (mark@mail.mmrrcc.upenn.edu) writes:  
>  
>> I'd like to run IDL (from UNIX) in batch mode with a command  
>> like:  
>>  
>> unixprompt> idl batchprog.pro  
>>  
>> How can I provide additional arguments to be used by the batch  
>> program. For example, is there a syntax like:  
>>  
>> unixprompt> idl batchprog.pro inputfile.txt  
>>  
>> or  
>>  
>> unixprompt> idl batchprog.pro 5 3.78 "John Doe" 24.0  
I implemented such a scheme by creating a unix shell which sets  
up various parameters before running idl. The unix shell writes  
into a file which is executed by IDL by means of the "setenv IDL\_STARTUP ..."  
scheme. The shell passes along arguments from the command line, which  
may contain IDL commands.

--  
Serdar S. Manizade <serdar.manizade@gssc.nasa.gov>  
Airborne Topographic Mapper Project  
NASA/GSFC/Wallops Flight Facility, Wallops Island, VA

---

---

Subject: Re: IDL batch mode: Command line args?  
Posted by [nospam](#) on Wed, 01 Apr 1998 08:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

In article <mgs-3103981949020001@sc9-14-46.thegrid.net> mgs@sd.cybernex.net (Mike Schienle) writes:

```
> Here's a variation along those lines. You could set the environment
> variable IDL_STARTUP to point to different batch programs, then call IDL.
> This would start IDL and run the batch file pointed to by IDL_STARTUP.
> Similarly, you could keep IDL_STARTUP pointing to the same file, but
> create that file using your UNIX program (shell, perl, etc.). Get the help
> of your local UNIX shell programming fanatic and you might end up with
> something pretty entertaining. I was doing stuff like this a couple years
> back. I could dig up some ideas from those efforts if you're interested.
> ...
```

I think it is better to use perl (or sh) with IDL to do this. The nicest way is like this:

```
#!/usr/local/bin/perl -w

open(IDL, "|/usr/local/bin/rsi/idl/bin/idl") ||
  die "Can't open IDL: $!";
$\="\n";

print IDL "var1 = $ARGV[0]";
print IDL "var2 = $ARGV[1]";

print IDL "doit_procedure, var1, var2";

##etc. You can loop over filenames:

foreach $filename (glob(*)){
  print IDL "process_file, '$filename'";
}

print IDL "exit";
```

I've also written perl scripts that create long IDL scripts that I can then check over and run, for example to convert a bunch of BMP files to several files of raw 3D data:

```
#!/usr/sbin/perl -w

$\ = "\n";
$outdir = "/data1/stuarts/calsim/spatial/surrogate1/";
$indir = "/data1/eo1/jam/";

print "get_lun, F";
```

```

foreach $suff (qw(1 b c d e f g h i j k l m n o p q r s t u)){

    $outfilename = $outdir . "data" . $suff;
    print "openw, F, '$outfilename'";

    for ($x = 0; $x < 200; $x++){
        $filename = $indir . "data" . $suff . $x . ".bmp";
        if (-f $filename){
            print "frame = read_bmp('$filename')";
            print "writeu, F, frame";
        } else {
            die "$filename not found\n";
        }
    }
    print "close, F";
}
print "free_lun, F";
print, "end";
--
Scott Stuart
stuart at ll mit edu

```

---

**Subject: Re: IDL batch mode: Command line args?**  
**Posted by [bowman](#) on Thu, 02 Apr 1998 08:00:00 GMT**  
[View Forum Message](#) <> [Reply to Message](#)

In article <35215975.41C6@mail.mmrrcc.upenn.edu>, Mark Elliott <mark@mail.mmrrcc.upenn.edu> wrote:

```

> I'd like to run IDL (from UNIX) in batch mode with a command
> like:
>
> unixprompt> idl batchprog.pro

```

This is how I run an IDL program in batch mode with arguments.

Create a script file named batch\_myprog like this:

```

-----
echo Running MYPROG
rm $1.out
rm $1.err

(echo .run myprog ; \
echo myprog, '$1\'; \
echo exit)          | \
idl 1>>$1.out 2>>$1.err

```

-----  
Make sure batch\_myprog is set to have execute permission. (This example is K shell, but C shell would be similar.)

In this case, MYPROG is an IDL procedure in the file myprog.pro. It takes one argument (the name of file to be read by MYPROG in this case). The script will print a message ("Running MYPROG"). Then it will remove files called \$1.out and \$1.err, where \$1 is the first argument to the script batch\_myprog when you run it. Then it will create an input stream telling IDL to compile myprog and execute it with the value of \$1 as the argument. This input stream is piped to IDL.

To run the batch job use:

```
nohup batch_my_prog input_file_1 &
```

I use this where the argument \$1 is a file name to be read by myprog.pro. The output of the IDL job will be in input\_file\_1.out and input\_file\_1.err. If you want to pass numeric arguments, you should probably redirect IDL output to different files than \$1.out and \$1.err.

This is certainly not a generic solution, but for simple argument lists it works fine. An alternative is to write the batch script arguments to a file and then pipe that to IDL. This script avoids the intermediate file by creating the input stream with 'echo'.

Regards, Ken Bowman

--

Kenneth P. Bowman, Assoc. Prof.                   409-862-4060  
Department of Meteorology                       409-862-4132 fax  
Texas A&M University                            bowmanATcsrcp.tamu.edu  
College Station, TX 77843-3150

---

Subject: Re: IDL batch mode: Command line args?  
Posted by [mallors](#) on Thu, 02 Apr 1998 08:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

```
>> Mark Elliott (mark@mail.mmrrcc.upenn.edu) writes:  
>>  
>>> I'd like to run IDL (from UNIX) in batch mode with a command  
>>> like:  
>>>  
>>> unixprompt> idl batchprog.pro  
>>>  
>>> How can I provide additional arguments to be used by the batch
```

```
>>> program. For example, is there a syntax like:
>>>
>>> unixprompt> idl batchprog.pro inputfile.txt
>>>
>>> or
>>>
>>> unixprompt> idl batchprog.pro 5 3.78 "John Doe" 24.0
```

You can read your data at the top of the batch file, with input redirected from a separate text file. Here is an example, with an IDL batch file test.pro reading input from the file input.dat

```
----- file test.pro -----
READ, A
READ, B
READ, C
HELP, A, B, C
----- END file test.pro -----
```

```
----- file input.dat -----
10
20
30
----- END file input.dat -----
```

Then from the unix prompt,

```
$ idl test < input.dat
```

```
A      FLOAT  =   10.0000
B      FLOAT  =   20.0000
C      FLOAT  =   30.0000
```

```
--
Robert S. Mallozzi
http://cspar.uah.edu/~mallozzir/
```

---