Subject: Re: Passing Batch File Names
Posted by Michael Galloy on Tue, 02 Sep 2008 15:54:19 GMT
View Forum Message <> Reply to Message

On Sep 2, 9:41 am, Trae <traewin...@gmail.com> wrote:
> Hey everyone,
>
> I have a code that calls a batch file to assign variables that I want
> the main level program to have. The names of the variables and the
> number may change randomly, and not by me, so I didn't want to use a
> procedure where I would have to change it every time an alteration was
> made in the save file.
>
> Now, I need to execute a batch file whose name is passed by another
> program. I've used call_method, call_procedure, and call_function,
> but I see nothing like call_batch.
>
> My current fix is the following:
>
> spawn, 'cp ' + in_name+' temp_name.pro'
>
> @temp_name.pro
>
> This doesn't seem very clean, but it makes all of the variables
> available to the main level.
>
> Any brighter ideas?

I thought EXECUTE would be the answer, but it appears that it can't
call a batch file:

```
IDL> @test
       5
IDL> print, execute('@test')

@test
 ^
% Illegal character in program text.
       0
```

Mike
--
www.michaelgalloy.com
Tech-X Corporation
Software Developer II

Subject: Re: Passing Batch File Names
Posted by Trae on Tue, 02 Sep 2008 16:03:06 GMT
View Forum Message <> Reply to Message

Good suggestion. I just tried it too and was about to post my null result.

I think the person that started writing this code is making life more difficult than it has to be. The best way to go about this, I think, is to write a procedure that reads the input file and returns an undefined structure. You can then check for tags on the structure, which gives you the necessary flexibility.

One of the problems is that the original input file must be human readable text. So no .sav files.

Still, if anyone has a way to dynamically call a batch file with a name that is passed as a variable, I would love to hear about it.

Cheers,
-Trae

On Sep 2, 11:54 am, "mgal...@gmail.com" <mgal...@gmail.com> wrote:
> On Sep 2, 9:41 am, Trae <traewin...@gmail.com> wrote:
>
>
>
>> Hey everyone,
>
>> I have a code that calls a batch file to assign variables that I want
>> the main level program to have. The names of the variables and the
>> number may change randomly, and not by me, so I didn't want to use a
>> procedure where I would have to change it every time an alteration was
>> made in the save file.
>
>> Now, I need to execute a batch file whose name is passed by another
>> program. I've used call_method, call_procedure, and call_function,
>> but I see nothing like call_batch.
>
>> My current fix is the following:
>
>> spawn, 'cp ' + in_name+' temp_name.pro'
>
>> @temp_name.pro
>
>> This doesn't seem very clean, but it makes all of the variables
>> available to the main level.
>
>> Any brighter ideas?

```
>
> I thought EXECUTE would be the answer, but it appears that it can't
> call a batch file:
>
> IDL> @test
>       5
> IDL> print, execute('@test')
>
> @test
>  ^
> % Illegal character in program text.
>       0
>
> Mike
> --www.michaelgalloy.com
> Tech-X Corporation
> Software Developer II
```

Subject: Re: Passing Batch File Names
Posted by pgrigis on Tue, 02 Sep 2008 16:03:24 GMT
View Forum Message <> Reply to Message

I guess it would be cleaner to put all variables in a structure
with a fixed name and just refer to the structure, but I understand
this would require rewriting the code, so this is not a good idea
now;-)

Ciao,
Paolo

Trae wrote:
> Hey everyone,
>
> I have a code that calls a batch file to assign variables that I want
> the main level program to have. The names of the variables and the
> number may change randomly, and not by me, so I didn't want to use a
> procedure where I would have to change it every time an alteration was
> made in the save file.
>
> Now, I need to execute a batch file whose name is passed by another
> program. I've used call_method, call_procedure, and call_function,
> but I see nothing like call_batch.
>
> My current fix is the following:
>
> spawn, 'cp ' + in_name+' temp_name.pro'
>

> @temp_name.pro
>
>
> This doesn't seem very clean, but it makes all of the variables
> available to the main level.
>
> Any brighter ideas?
>
> -Trae

Subject: Re: Passing Batch File Names
Posted by Brian Larsen on Tue, 02 Sep 2008 16:14:38 GMT
View Forum Message <> Reply to Message

Trae,

I'm thinking how one accomplishes this in C, seems to me that this is
an include file. Is there anything that thinking of it that way makes
clearer or different? Maybe even as weird as restore and save?

One cleaner and better solution to the call_batch thing is to use
file_copy instead of spawn 'cp '. You take a huge time hit on spawn
since it executes your .cshrc file each time and with solarsoft in
your .cshrc that takes a while. I learned this the hard way doing
some sqlite work from idl.

Cheers,

Brian

-- --------------
Brian Larsen
Boston University
Center for Space Physics
http://people.bu.edu/balarsen/Home/IDL

Subject: Re: Passing Batch File Names
Posted by Trae on Tue, 02 Sep 2008 16:27:35 GMT
View Forum Message <> Reply to Message

On Sep 2, 12:14 pm, Brian Larsen <balar...@gmail.com> wrote:
> Trae,
>
> I'm thinking how one accomplishes this in C, seems to me that this is

> an include file. Is there anything that thinking of it that way makes
> clearer or different? Maybe even as weird as restore and save?

Got it!!!

Even though EXECUTE looks like a function, any variables defined in
the string are inherited by the next level up. So the easy thing to
do is

1. Open the passed file using OPENR
2. Read the contents of the batch file into a string variable.
3. Use EXECUTE to, well, execute the string variable.

So this is the exact same as @batch_name, but this way I don't need
to know the batch file name a priori, at the expense of a few lines of
code.

YAY!

-Trae

PS Brian I will use the file_copy trick in future. Good tip!

>
> One cleaner and better solution to the call_batch thing is to use
> file_copy instead of spawn 'cp '. You take a huge time hit on spawn
> since it executes your .cshrc file each time and with solarsoft in
> your .cshrc that takes a while. I learned this the hard way doing
> some sqlite work from idl.
>
> Cheers,
>
> Brian
>
>   -- --------------
> Brian Larsen
> Boston University
> Center for Space Physicshttp://people.bu.edu/balarsen/Home/IDL

Subject: Re: Passing Batch File Names
Posted by Trae on Tue, 02 Sep 2008 16:50:00 GMT
View Forum Message <> Reply to Message

Just in case another poor soul needs this and searches the group, here
are the lines of code that made it work. NB: EXECUTE only accepts
scalars as an input. Bogus.

First define a batch file named batch_test.pro with the commands

```
a=5
b=6
c=sin(!dpi/4.)
```

Now here are the commands to read and execute it.

```
file='batch_test.pro'




openr, lun, file, /GET_LUN
var_arr=''
;while not eof do begin

WHILE ~ EOF(lun) DO BEGIN
   line=''
   READF,lun , line
   var_arr=[var_arr,line]
ENDWHILE
FREE_LUN, lun
var_arr=var_arr[1:*]

for i=0ul,n_elements( var_arr) -1ul do result=execute(var_arr[i])

END
```

So you can define the variable 'file' anyway you want and it can have
as many or as few commands as you need.

Thanks for the help everyone!

I'm off to be quite pleased with myself for awhile! :)

Cheers,
-Trae

Subject: Re: Passing Batch File Names
Posted by pgrigis on Tue, 02 Sep 2008 16:57:14 GMT

View Forum Message <> Reply to Message

Does it work with multiline statements such as

```
for i=0,10 do begin &$
   print,i &$
endfor
```

which are legal for batch files?

Ciao,
Paolo

Trae wrote:

```
> Just in case another poor soul needs this and searches the group, here
> are the lines of code that made it work.  NB: EXECUTE only accepts
> scalars as an input.  Bogus.
>
> First define a batch file named batch_test.pro with the commands
>
> a=5
> b=6
> c=sin(!dpi/4.)
>
>
> Now here are the commands to read and execute it.
>
>
> file='batch_test.pro'
>
>
>
>
> openr, lun, file, /GET_LUN
> var_arr=''
> ;while not eof do begin
>
> WHILE ~ EOF(lun) DO BEGIN
>    line=''
>    READF,lun , line
>    var_arr=[var_arr,line]
> ENDWHILE
> FREE_LUN, lun
> var_arr=var_arr[1:*]
>
> for i=0ul,n_elements( var_arr) -1ul do result=execute(var_arr[i])
>
> END
>
```

>
> So you can define the variable 'file' anyway you want and it can have
> as many or as few commands as you need.
>
> Thanks for the help everyone!
>
> I'm off to be quite pleased with myself for awhile! :)
>
> Cheers,
> -Trae

Subject: Re: Passing Batch File Names
Posted by Trae on Tue, 02 Sep 2008 17:15:08 GMT
View Forum Message <> Reply to Message

No. It fact, it can lead to segmentation faults.

You could easily test the string for '$' and then combine lines as
needed.

However, for this application, a series of variables are being
defined. It is easier to keep 1 line per command than try to make
anything to tricky. Of course, if you have time to play you can make
this procedure as general as you want.

-Trae

On Sep 2, 12:57 pm, pgri...@gmail.com wrote:
> Does it work with multiline statements such as
>
> for i=0,10 do begin &$
>   print,i &$
> endfor
>
> which are legal for batch files?
>
> Ciao,
> Paolo
>
> Trae wrote:
>> Just in case another poor soul needs this and searches the group, here
>> are the lines of code that made it work. NB: EXECUTE only accepts
>> scalars as an input. Bogus.
>
>> First define a batch file named batch_test.pro with the commands
>
>> a=5

```
>>  b=6
>>  c=sin(!dpi/4.)
>
>>  Now here are the commands to read and execute it.
>
>>  file='batch_test.pro'
>
>>  openr, lun, file, /GET_LUN
>>  var_arr=''
>>  ;while not eof do begin
>
>>  WHILE ~ EOF(lun) DO BEGIN
>>     line=''
>>     READF,lun , line
>>     var_arr=[var_arr,line]
>>  ENDWHILE
>>  FREE_LUN, lun
>>  var_arr=var_arr[1:*]
>
>>  for i=0ul,n_elements( var_arr) -1ul do result=execute(var_arr[i])
>
>>  END
```

> \>> So you can define the variable 'file' anyway you want and it can have
> \>> as many or as few commands as you need.
>
> \>> Thanks for the help everyone!
>
> \>> I'm off to be quite pleased with myself for awhile! :)
>
> \>> Cheers,
> \>> -Trae

Subject: Re: Passing Batch File Names
Posted by Jean H. on Tue, 02 Sep 2008 17:34:50 GMT

View Forum Message <> Reply to Message

it might be shorter / faster to create a dummy batch file, in its own
file, that calls the real batch file. In your code, you always call the
same batch file, and the only thing you have to do is to write the batch
file name (the one defining your variables) in that pro file. 1 write
statement and your are good!

something like:

```
openW, lun, 'myDummyBatchFile.pro', /get_lun
printF,lun, '@' + myGoodBatchFile_fileName
```

close,lun

then in your code
@myDummyBatchFile

Jean

Subject: Re: Passing Batch File Names
Posted by Brian Larsen on Tue, 02 Sep 2008 18:15:39 GMT
View Forum Message <> Reply to Message

Trae,

just because I like messing with you, don't forget that you can read
the whole file at once w/o having to use that dreaded "WHILE ~EOF(lun)
DO BEGIN" line for a huge speed improvement, not that it matters for
this application.

```
file='batch_test.pro'
;; how many lines are in the file
lines = file_lines(file)
;; an array to hold the whole file, execute doesn't care if the string
is '' it just executes null
var_arr=strarr(lines)
openr, lun, file, /GET_LUN
READF,lun , var_arr
for i=0ul,n_elements( var_arr) -1ul do result=execute(var_arr[i])
free_lun, lun
END
```

Cheers,

Brian

-- --------------
Brian Larsen
Boston University
Center for Space Physics
http://people.bu.edu/balarsen/Home/IDL

Subject: Re: Passing Batch File Names

Posted by Paul Van Delst[1] on Tue, 02 Sep 2008 18:16:56 GMT
View Forum Message <> Reply to Message

Trae wrote:
> Just in case another poor soul needs this and searches the group, here
> are the lines of code that made it work. NB: EXECUTE only accepts
> scalars as an input. Bogus.

Your solution is very handy, but one should point out that blind execution of commands read from a file is potentially dangerous. It's probably not that big of an issue for IDL, but some of those batch file lines could be something like,

```
stuff = file_search('*')
for i=0,n_elements(stuff)-1 do file_delete, stuff[i], /quiet,/recursive
```

Keep those batch files protected from disgruntled employees. :o)

Using exec capabilities blindly is generally discouraged in scripting languages.

cheers,

paulv

```
>
> First define a batch file named batch_test.pro with the commands
>
> a=5
> b=6
> c=sin(!dpi/4.)
>
>
> Now here are the commands to read and execute it.
>
>
> file='batch_test.pro'
>
>
>
>
> openr, lun, file, /GET_LUN
> var_arr=''
> ;while not eof do begin
>
> WHILE ~ EOF(lun) DO BEGIN
>    line=''
>    READF,lun , line
>    var_arr=[var_arr,line]
> ENDWHILE
```

> FREE_LUN, lun
> var_arr=var_arr[1:*]
>
> for i=0ul,n_elements(var_arr) -1ul do result=execute(var_arr[i])
>
> END
>
>
> So you can define the variable 'file' anyway you want and it can have
> as many or as few commands as you need.
>
> Thanks for the help everyone!
>
> I'm off to be quite pleased with myself for awhile! :)
>
> Cheers,
> -Trae
>

Subject: Re: Passing Batch File Names
Posted by Bob[3] on Tue, 02 Sep 2008 20:20:05 GMT
View Forum Message <> Reply to Message

Since (among other things) EXECUTE is not allowed in VM, does anyone
have any ideas on how to do this WITHOUT using EXECUTE?

Subject: Re: Passing Batch File Names
Posted by Michael Galloy on Tue, 02 Sep 2008 20:32:36 GMT
View Forum Message <> Reply to Message

On Sep 2, 2:20 pm, Bob Crawford <Snowma...@gmail.com> wrote:
> Since (among other things) EXECUTE is not allowed in VM, does anyone
> have any ideas on how to do this WITHOUT using EXECUTE?

The OP wanted a quick fix that imported the contents of a dynamically
written batch file into a main-level program. This whole scenario is
not allowed in the VM.

If this were something more permanent (maybe for an application in the
VM), I would place the data in a data file, not a batch file. Then
have my code read the data into a hash table and use that to look up
"variable names" (I assume the variable names are supposed to actually
represent some other quantity).

Mike

--
www.michaelgalloy.com
Tech-X Corporation
Software Developer II

Subject: Re: Passing Batch File Names
Posted by Jean H. on Tue, 02 Sep 2008 20:33:51 GMT
View Forum Message <> Reply to Message

Bob Crawford wrote:
> Since (among other things) EXECUTE is not allowed in VM, does anyone
> have any ideas on how to do this WITHOUT using EXECUTE?

can any batch file (even if you hard code its name) can be ran in the
VM?? ... wouldn't it have to be converted to a sav file first?

if it is not a problem, read my previous post then...

Jean

Subject: Re: Passing Batch File Names
Posted by Trae on Wed, 03 Sep 2008 02:35:27 GMT
View Forum Message <> Reply to Message

Well this got to be too much of a kludge.

I convinced the others involved in this project that it was wise to
change gears and make a series of procedures that passed out variables
with positional parameters. The name of the procedure can be passed
as a variable name and called with CALL_PROCEDURE.

Still, it seems like there should have been an easy way to do this.

Cheers,
-Trae

Subject: Re: Passing Batch File Names
Posted by Trae on Wed, 03 Sep 2008 19:18:14 GMT
View Forum Message <> Reply to Message

> Your solution is very handy, but one should point out that blind execution of commands
> read from a file is potentially dangerous. It's probably not that big of an issue for IDL,
> but some of those batch file lines could be something like,

Oh yeah. I would never do this in perl or anything that had write
permissions in critical areas. The batch files made would be few in
number and only made a handful of people.

Still, it just became easier to change the structure of the previously
written codes then to deal with all of the issues, this being a big
one, that came about by trying to use batch files.

Good comment.

-Trae

On Sep 2, 2:16 pm, Paul van Delst <Paul.vanDe...@noaa.gov> wrote:
> Trae wrote:
>> Just in case another poor soul needs this and searches the group, here
>> are the lines of code that made it work. NB: EXECUTE only accepts
>> scalars as an input. Bogus.
>
> Your solution is very handy, but one should point out that blind execution of commands
> read from a file is potentially dangerous. It's probably not that big of an issue for IDL,
> but some of those batch file lines could be something like,
>
> stuff = file_search('*')
> for i=0,n_elements(stuff)-1 do file_delete, stuff[i], /quiet,/recursive
>
> Keep those batch files protected from disgruntled employees. :o)
>
> Using exec capabilities blindly is generally discouraged in scripting languages.
>
> cheers,
>
> paulv
>
>
>
>> First define a batch file named batch_test.pro with the commands
>
>> a=5
>> b=6
>> c=sin(!dpi/4.)
>
>> Now here are the commands to read and execute it.
>
>> file='batch_test.pro'
>
>> openr, lun, file, /GET_LUN

```
>>  var_arr=''
>>  ;while not eof do begin
>
>>  WHILE ~ EOF(lun) DO BEGIN
>>     line=''
>>     READF,lun , line
>>     var_arr=[var_arr,line]
>>  ENDWHILE
>>  FREE_LUN, lun
>>  var_arr=var_arr[1:*]
>
>>  for i=0ul,n_elements( var_arr) -1ul do result=execute(var_arr[i])
>
>>  END
```

>
>> So you can define the variable 'file' anyway you want and it can have
>> as many or as few commands as you need.
>
>> Thanks for the help everyone!
>
>> I'm off to be quite pleased with myself for awhile! :)
>
>> Cheers,
>> -Trae