

---

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. In 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 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
```

---