Subject: programmatically compile a procedure without a .pro extension
Posted by wlandsman on Sun, 25 Aug 2013 01:10:20 GMT
View Forum Message <> Reply to Message

Does anyone know if there is a way to programmatically compile a procedure that does not have a .pro extension?   For example, if I have a file named a1.bak containing the following 4 lines:

```
pro a1
print,'Inside procedure a1'
return
end
```

then the user can successfully compile this using

```
IDL> .run a1.bak
```

But the tools to programmatically compile a file don't work in this case, i.e.

```
IDL> resolve_routine,'a1.bak'
or
IDL> file_compile,'a1.bak'
```

where file_compile is from Craig Markwardt's library
 http://www.physics.wisc.edu/~craigm/idl/down/file_compile.pr o

It is reasonable to ask why I would not simply use a .pro extension.   But the answer is fairly complicated so I don't (yet) want to distract from the straightforward question above.   Thanks,
--Wayne

Subject: Re: programmatically compile a procedure without a .pro extension
Posted by Heinz Stege on Sun, 25 Aug 2013 12:10:37 GMT
View Forum Message <> Reply to Message

If you don't want to get a .pro file by renaming (or making a copy of)
the .bak file, you can do something like the following. The routine
below creates a temporary .pro file and includes the files with other
extensions by use of the @ include file character:

```
pro compile,filenames
;
tempfile='compile_temp'
openw,lun,/get_lun,tempfile+'.pro'
printf,lun,'@'+filenames,form='(a)'
free_lun,lun
;
catch,error
if ~error then resolve_routine,tempfile
```

```
catch,/cancel
;
end
```

The catch statement suppresses an error message from the compiler,
that there is no compile_temp routine.

Note that you can specify the filenames by a string array, if you want
to compile more than one file.

Cheers, Heinz

Subject: Re: programmatically compile a procedure without a .pro extension
Posted by wlandsman on Mon, 26 Aug 2013 03:23:53 GMT

View Forum Message <> Reply to Message

Heinz,

Thanks.   That is an elegant use of CATCH.

I still have mild trepidation about requiring an arbitrary user to have write access to a disk. But I've never had any problem using IDL_TMPDIR as a scratch writeable directory, so I should probably
stop worrying.   --Wayne

On Sunday, August 25, 2013 8:10:37 AM UTC-4, Heinz Stege wrote:

```
>  If you don't want to get a .pro file by renaming (or making a copy of)
>
>  the .bak file, you can do something like the following. The routine
>
>  below creates a temporary .pro file and includes the files with other
>
>  extensions by use of the @ include file character:
>
>
>
>  pro compile,filenames
>
>  ;
>
>  tempfile='compile_temp'
>
>  openw,lun,/get_lun,tempfile+'.pro'
>
>  printf,lun,'@'+filenames,form='(a)'
>
>  free_lun,lun
```

>
> ;
>
> catch,error
>
> if ~error then resolve_routine,tempfile
>
> catch,/cancel
>
> ;
>
> end
>
>
>
> The catch statement suppresses an error message from the compiler,
>
> that there is no compile_temp routine.
>
>
>
> Note that you can specify the filenames by a string array, if you want
>
> to compile more than one file.
>
>
>
> Cheers, Heinz