
Subject: Re: Running IDL .pro script from another app, using the IDL API
Posted by [mankoff](#) on Fri, 14 Sep 2007 17:06:47 GMT

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

On Sep 14, 10:33 am, "Jeff N." <jnett...@utk.edu> wrote:

> On Sep 14, 10:12 am, mankoff <mank...@gmail.com> wrote:

>

>

>

>> On Sep 14, 7:23 am, Mike <Michael.Mill...@gmail.com> wrote:

>

>>> On Sep 13, 5:30 pm, mankoff <mank...@gmail.com> wrote:

>

>>>> Wait... can't this be like a hundred times easier? On *nix you can
>>>> just execute from the command line "idl foo.pro" and foo will be
>>>> executed in IDL. Doesn't windows have some way to run startup
>>>> commands? I know everything is a bit harder to do in Windows, but I
>>>> don't think callable IDL is the necessary solution to this. Sorry, I
>>>> don't have IDL in Windows to test this on.

>

>>> I don't think that the problem is with windows itself. I'm sure there
>>> is a way to spawn or otherwise start something from within their
>>> desktop application (details depending on how it was written). The
>>> problem is that there is not a command-line idl on windows, so the
>>> "idl foo.pro" idiom is not available. To start IDL, they have to
>>> start the IDL IDE. On windows, the "I" really does mean interactive!
>>> That's one reason (among others) why we run all of our "batch" IDL
>>> codes on *nix boxen.

>

>>> Mike

>

>> I've never once run IDLDE, but I just launched it on OS X (unix), and
>> under the startup tab in Preferences you can set a procedure. I assume
>> you can do this on Windows too. That procedure should run whatever
>> code the original poster wants to run. All they then need to do is
>> launch the IDLDE.

>

>> The last line of the startup procedure could be "exit" to shut things
>> down too. Of course, it would now be hard to use IDL for anything
>> other than this one task, so I would write the startup procedure like
>> so:

>

>> 1) widget with 5 second countdown timer that alerts that this 'auto
>> processing for Jim Brown' will be run in 5 seconds unless you click
>> "no"

>

>> 2) If "no" then drop into regular IDL/IDLDE mode

>

```
>> 3) IF time times out, then run auto processing, and exit (shutting
>> down the IDLDE)
>
>> -k.
>
> Well, two things come to mind: 1) Remember that the startup procedure
> can't be any old idl procedure: it has to be "batch jobs" like you
> would run with the @ symbol, or enter yourself at the command line (no
> multiple line statements). 2) What you *can* do is give the idlde
> some command line arguments using the -args switch. During my
> dissertation I used these, for example, to set my path preferences
> based on whether or not i had my external hard drive hooked up on my
> desktop, which had lots of data i'd need. You can parse whatever's
> passed via -args with the COMMAND_LINE_ARGS function.
>
> What i think i would do is this: when i want idl to get started and
> do some automatic processing, i'd start it with an -args switch. I'd
> set up my startup file to look for that switch, and if it finds it, do
> the automatic processing and then quit. If that switch isn't found,
> start normally.
>
> Jeff
```

Yes all those caveats apply. But presumably the startup.pro (which is limited to batch) would be something like:

```
startup_real_procedure_abilities_go_here.pro
exit
```

and then you can put whatever you want into the startup_real....pro procedure.

Anyway, I think it is pretty clear from this thread that windows and batch jobs can do what was requested without resorting to advanced programming techniques. Its up to Jim Brown which approach to take now...

-k.
