

---

Subject: Re: "include" a file

Posted by [JDS](#) on Fri, 26 Jun 2009 19:54:09 GMT

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

---

On Jun 26, 3:50 pm, JD Smith <jdsmith.nos...@yahoo.com> wrote:

> Various other languages have the option to include and evaluate the  
> program contents of another file at runtime. IDL has the "@"  
> operator, but that happens at compile time, so you need to know which  
> file to include in advance. I find myself needing to drop small  
> "parameter" files in individual directories for a routine to process  
> as it crawls through. I could certainly prepare an IDL .sav file, or  
> some other data format, parse that, and set-up structures and  
> variables as needed, but that makes editing and updating the file very  
> painful. What IDL needs is a way to "include" a file directly, and  
> evaluate its contents. Finding nothing, I came up with the following  
> concept:

>  
> ;; include -- Include and evaluate the IDL command contents of a file.  
> ;; To use, give the variable "include\_file" in the same scope the name  
> ;; of a valid file containing IDL commands (batch syntax only), then  
> ;; batch include this file, ala:

> ;;  
> ;; include\_file='/path/to/file'  
> ;; @include  
> ;;  
> \_inc\_lines=replicate('file\_lines(include\_file))  
> openr,\_inc\_un,include\_file,/get\_lun  
> readf,\_inc\_un,\_inc\_lines  
> \_inc\_wh=where(~stregex(\_inc\_lines,'\\$(;.\*)?[ \t]\*\$',/  
> BOOLEAN),\_inc\_cnt)  
> \_inc\_start=0L  
> for \_inc\_i=0L,\_inc\_cnt-1 do begin  
> \_inc\_parts=\_inc\_lines[\_inc\_wh[\_inc\_i]]  
> if \_inc\_wh[\_inc\_i] gt \_inc\_start then \$  
> \_inc\_parts=strjoin( \$  
> reform((stregex(\_inc\_lines[\_inc\_start:\_inc\_wh  
> [\_inc\_i]-1],\$  
> '\*(.\*) \*\$\\$(;.\*)?[ \t]\*\$', \$  
> /SUBEXPR,/EXTRACT))[1,\*])) +  
> \_inc\_parts  
> \_inc\_void=execute(\_inc\_parts)  
> \_inc\_start=\_inc\_wh[\_inc\_i]+1L  
> endfor  
> free\_lun,temporary( \$ ; Clean-up all variables  
> (\_inc\_parts=temporary( \$  
> (\_inc\_wh=temporary( \$  
> (\_inc\_lines=temporary( \$  
> (\_inc\_void=temporary( \$

```
>      (_inc_cnt=temporary( $
>      (_inc_i=temporary( $
>      (_inc_start=temporary(_inc_un)))))))))))))
>
> This works just fine, collapsing multi-line commands, and executing
> them, at the cost of temporarily polluting the current scope with
> "_inc_" variables (these are left undefined after the @include). You
> have to use "batch syntax", aka as "standalone single line command"
> syntax, but for my purposes this isn't a major limitation. It uses
> execute, so won't work in the IDL_VM, and if you try to do it many
> times in a loop, you might regret it. But for quickly setting up
> human-editable parameter lists, I find it works great.
>
> Do others encounter this problem, and has anyone solved it in a
> different way?
```

OK, that got poorly formatted by the news relay:

<http://tir.astro.utoledo.edu/idl/include.pro>

---