

---

Subject: Re: Path info in \*.sav file

Posted by [Dick Jackson](#) on Mon, 24 Aug 1998 07:00:00 GMT

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

---

Herbert Tsang wrote:

>

> In my program, I have to load a couple gif images, however, I would like  
> the path to these images be more flexible so that I don't need to define the  
> absolute path in compile time.

Hi Herbert,

I've had to deal with this issue, I think I can lend a hand.

> Is it possible to:

> 1) incorporate the gif file in the sav file?

You could, of course, read the gif file into an array, but one sav file can contain only routines \_or\_ variables, not both. Saving a second sav file doesn't get you any closer to your goal, so this doesn't look good.

> 2) setup relative path for everytime when I need to load the image?

Unfortunately, a routine in a sav file seems to have no way of knowing what file it came from. To get around this, I suggest using two names, a 'SavName' and a 'ProName' as follows:

- savname.pro - the main program file
- savname.sav - saved from savname.pro
- pronaame.pro - what you will call from IDL, and which \_can\_ find its own location

```
===== pronaame.pro =====
```

```
PRO ProName
```

```
COMMON DirCommon, dir
```

```
;; Find out the directory that this file is in
```

```
Help, Calls = c
```

```
l = c(0) ; c(0) describes THIS routine
```

```
;; e.g.: PRONAME </home/djackson/idl/lib/pronaame.pro( 5)>
```

```
;; Note: the use of '/' here is Unix-specific
```

```
;; Extract dir from the string
```

```
IF StrPos(l, '/') EQ -1 THEN dir = './' ELSE $
```

```
dir = StrMid(l, StrPos(l, '<')+1, RStrPos(l, '/')-StrPos(l, '<'))
```

```

Print, 'Using directory: ' + dir

;; Restore the .sav file and call the main routine
;; (if savname.pro and savname.sav exist, savname.pro would be used)
Restore, dir + 'savname.sav'
SavName

```

END

```

;===== savname.pro =====

```

PRO SavName

```

COMMON DirCommon, dir

```

```

;; Show it works by printing contents of this directory.
Print, FindFile(dir)

```

END

```

;===== then, in IDL: =====
IDL> .run savname
% Compiled module: SAVNAME.
IDL> save,/routines,filename='savname.sav'

```

```

;===== then, in a new IDL session: =====
IDL> prona
% Compiled module: PRONAME.
prona.pro prona.pro~ savname.pro savname.pro~ savname.sav

```

Your aim in doing this may be to distribute a .sav file without the .pro source, so in this case you would send savname.sav and prona.pro (with your .gif files!) that simply need to stay in the same directory, somewhere on the IDL !PATH.

While you are debugging the code in savname.pro, you may want to call it directly, without going through 'prona'. If so, then you would need to find the directory in the savname routine in savname.pro, by adding something like this after the COMMON statement:

```

IF N_Elements(dir) EQ 0 THEN BEGIN ; If dir is undefined
  ;; same code to find out the directory that this file is in
  Help, Calls = c
  ;; [...]
END

```

This may look a bit messy, but this is the best way I've found to handle it.

Best regards,

--

-Dick

Dick Jackson            Fanning Software Consulting, Canadian Office  
djackson "at" dfanning "dot" com   Winnipeg, Manitoba   (204) 885-0331  
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

---