
Subject: How do I save parameters for next run?

Posted by [Yunxiang Zhang](#) on Wed, 31 Mar 2004 22:28:58 GMT

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

Folks,

I am wondering if I can save the user input parameters for the next run of my program without creating additional files for it's "memory".

Is it possible? Thanks.

Yunxiang

Subject: Re: How do I save parameters for next run?

Posted by [Rick Towler](#) on Fri, 02 Apr 2004 00:09:06 GMT

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

"Yunxiang Zhang" wrote...

- > Sorry for the confusion. I just want my stand alone application to have a
- > small "memory" interactively. For example, I have a program, say,
- > 'mypro.sav' and needs user to input a series of parameters, say p1, p2
- > ...pn through an input gui. Then the program runs perfect. But then if
- > the next run the user only need to change a small fraction of the
- > parameters, it would be nice that my program can have a small memory so
- > that my gui displays the parameters input last time and the user need
- > only change maybe one parameter and press 'enter' to continue the
- > rest of the calculations.
- >
- > The idea comes from my colleagues and I still can't think of a way without
- > creating some additional log file. Maybe somebody can tell me how to hack
- > the .sav file so that i can dock some data into a specific region without
- > breaking the .sav file.

I am assuming you mean between IDL VM sessions. For instance, you run the application today, close it, then next week you run it again and you want those previous parameters to be "sticky".

You could hack the .sav file but why waste the time? I mean, I am all about hacks like this but... You could have the same functionality using a simple log file in minutes.

But if you are dead set against a second file, then I would try adding a function that returns your parameters which you would use to set the defaults when the application launches. Something simple like:

```
function setParms
```

```
p1 = 0.1  
p2 = 0.2  
p3 = 4.5
```

```
RETURN, [p1,p2,p3]
```

```
end
```

Then I would add an updateParms procedure that opens your .sav file and given an array of offsets writes values into that file. Since you won't know the offsets yet just make them up.

Compile your .sav file and with your favorite hex editor go looking for your parameters. Note the offsets, then go back to your IDL code and change the offsets in your updateParms pro.

Then compile it and hope that the offsets haven't changed.

A slightly more brutish approach would be to simply create a dummy function that explicitly defines a large array. Then just find the offset to the beginning of the data and read/write your parameters in there.

This of course will probably not work if you have compressed your .sav file. You'll also have to figure out path and filename details. And then there are endianness issues. I don't know how the .sav files deal with this.

Have fun.

-Rick

Subject: Re: How do I save parameters for next run?
Posted by [Pepijn Kenter](#) on Fri, 02 Apr 2004 07:23:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yunxiang Zhang wrote:

```
> Hi,  
>  
> Sorry for the confusion. I just want my stand alone application to have a  
> small "memory" interactively. For example, I have a program, say,  
> 'mypro.sav' and needs user to input a series of parameters, say p1, p2  
> ...pn through an input gui. Then the program runs perfect. But then if  
> the next run the user only need to change a small fraction of the  
> parameters, it would be nice that my program can have a small memory so  
> that my gui displays the parameters input last time and the user need
```

> only change maybe one parameter and press 'enter' to continue the
> rest of the calculations.
>

It seems to me that the best design is to let the gui remember the last users parameters. The mypro program must only care about the values of the parameters, not about user convenience. This approach has the advantage that, if you later decide that some parameters should not be shared between users, you'll only have to change the gui, the myprog program can remain unchanged.

Of course, the gui will need a config file if you want the values to be remembered when the gui is closed. :-)

Pepijn.

Subject: a simple hack!Re: How do I save parameters for next run?
Posted by [Yunxiang Zhang](#) on Mon, 05 Apr 2004 21:17:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

Folks,

Thanks for all your input here. I was trying to look into Craig's library and see if I can come up with a solution. But it turned to be a little bit complicated for me. Today, I finally did it with a simple hack by appending the default data to the end of the .sav file. The code will read itself, the .sav file, to get the default value and after user input it will update the data appended. The only drawback is you can't run the program before you generate the .sav file and append the default data to the end of the .sav file. Here's how the code looks like,

PRO MYPRO

```
...
values=fltarr(n_para)
openr,lun,'mypro.sav',/append,/get_lun
point_lun,-lun,lunpos
point_lun,lun,lunpos-4*n_para
readu,lun,values
free_lun,lun

;function that returns user input parameters
finput=InputBoxes(title=mytitle,$
prompts=prompts, values=values, labels=160)

openu,lun,'mypro.sav',/append,/get_lun
point_lun,-lun,lunpos
point_lun,lun,lunpos-4*n_para
```

```
writetu,lun,float(finput)
free_lun,lun
...
END
```

Cheers,
Yunxiang

On Thu, 1 Apr 2004, Yunxiang Zhang wrote:

```
>
> Hi,
>
> Sorry for the confusion. I just want my stand alone application to have a
> small "memory" interactively. For example, I have a program, say,
> 'mypro.sav' and needs user to input a series of parameters, say p1, p2
> ...pn through an input gui. Then the program runs perfect. But then if
> the next run the user only need to change a small fraction of the
> parameters, it would be nice that my program can have a small memory so
> that my gui displays the parameters input last time and the user need
> only change maybe one parameter and press 'enter' to continue the
> rest of the calculations.
>
> The idea comes from my colleagues and I still can't think of a way without
> creating some additional log file. Maybe somebody can tell me how to hack
> the .sav file so that i can dock some data into a specific region without
> breaking the .sav file.
>
> Thanks,
>
> Yunxiang
>
> On Thu, 1 Apr 2004, Reimar Bauer wrote:
>
>> Craig Markwardt wrote:
>>
>>>
>>> Reimar Bauer <R.Bauer@fz-juelich.de> writes:
>>>> > Folks,
>>>> >
>>>> > I am wondering if I can save the user input parameters for the next
>>>> > run of my program without creating additional files for it's "memory".
>>>> >
>>>> > Is it possible? Thanks.
>>>> >
>>>> > Yunxiang
>>>> >
>>>>
```

>>>> Take a look on save and restore
>>>
>>> But he asked about not creating additional files. Not too many
>>> options for that. If running on Unix, it's possible to wrap IDL in a
>>> script, either Perl or Bourne shell or whatever, and then establish a
>>> protocol for the script to capture your crucial outputs. Not elegant
>>> but it could work.
>>>
>>> Craig
>>
>> I read twice things not right. He asked for user input parameters.
>>
>> But I can't answer how to do it without writing to a file.
>>
>> Did you know the idl journal command?
>>
>>> idl
>>
>> IDL> journal, 'my_user_journal.pro' start the journal file and
>> IDL> a=10
>> IDL>
>> IDL> journal ends recording the user inputs.
>> IDL> exit
>>
>>
>> Afterwards this file could be called again by @my_user_journal from the IDL>
>> prompt, And print,a gives 10.
>>
>>
>> Reimar
>>
>>
>>
>
>
>
