Subject: Re: readf and structures
Posted by Benjamin Hornberger on Tue, 10 Apr 2007 14:52:44 GMT
View Forum Message <> Reply to Message

Wox wrote:

```
> Hi All,
>
> Is there an easy way of printing arrays of structures to a file and
> read them in again, without reading the fields separate?
>
> This doesn't work (also tried with specifying formats):
>
> tout={a:0.,b:'name'}
> tin={a:0.,b:''}
>
> openw,lun,'c:\test.txt',/get_lun
> printf,lun,tout
> close,lun
> free_lun,lun
>
> openr,lun,'c:\test.txt',/get_lun
> readf,lun,tin
> close,lun
> free_lun,lun
>
>
```

If I want to store a single IDL variable in a file, I usually use a .sav file. Of course that's not a good solution if the data is to be read back with a different program or language.

If you just use the plain 'save' and 'restore' procedures, the name of the variable at the time of writing will be stored, and the variable will be restored with exactly the same name. I find that inconvenient, because often I want to name the variable independently when I read it back from the file. That's why I wrote the two routines attached:

```
IDL> a=dist(300)
IDL> write_sav,'var.sav',a
```

(two days later)

```
IDL> b=read_sav('var.sav')
```

Cheers,
Benjamin

--- ---------

```
;+
; NAME:
;    WRITE_SAV.PRO
;
;
; PURPOSE:
;
;    This is a simple wrapper routine to save a single IDL variable
;    (of any type) in an IDL .sav file. The advantage of using
;    write_sav and read_sav over using SAVE and RESTORE directly is
;    that when using RESTORE, the variable will be restored under the
;    exact name it had when saved. This is not always desired.
;
;    See the IDL help on the SAVE and RESTORE routines for more
;    information on IDL .sav files.
;
;
; AUTHOR:
;
;    Benjamin Hornberger
;    benjamin.hornberger@stonybrook.edu
;
;
; CATEGORY:
;
;    Utilities
;
;
; CALLING SEQUENCE:
;
;    write_sav, filename, var
;
;
; INPUT PARAMETERS:
;
;    filename: Name / path of the file to be saved.
;
;    var: IDL variable of any type.
;
;
; KEYWORDS:
;
;    None.
;
;
; MODIFICATION HISTORY:
```

```
;
;    2006-05-01 BH: Written.
;-

PRO write_sav, filename, var

   on_error, 2
   compile_opt idl2

   IF n_elements(filename) EQ 0 THEN message, 'must provide filename'
   IF n_elements(var) EQ 0 THEN message, 'must provide variable'

   save, var, /compress, filename=filename, $
         description='bh_idl_saved_variable'

END
```

--- ---------

```
;+
; NAME:
;    READ_SAV.PRO
;
;
; PURPOSE:
;
;    This is a function to read an IDL variable from a .sav file which
;    has been created with write_sav. pro. See write_sav.pro for
;    details.
;
;
; AUTHOR:
;
;    Benjamin Hornberger
;    benjamin.hornberger@stonybrook.edu
;
;
; CATEGORY:
;
;    Utilities
;
;
; CALLING SEQUENCE:
;
;    result = READ_SAV(filename)
;
;
```

```
; RETURN VALUE:
;
;   This function returns the IDL variable which had been stored in
;   the .sav file.
;
;
; INPUT PARAMETERS:
;
;   filename: Name / path of the file to be restored.
;
;
; INPUT KEYWORDS:
;
;   RELAXED_STRUCTURE_ASSIGNMENT: This keyword is passed to the
;      RESTORE routine. See the documentation to RESTORE for
;      details.
;
;
; MODIFICATION HISTORY:
;
;   2006-05-01 BH: Written.
;-

FUNCTION read_sav, filename, $

relaxed_structure_assignment=relaxed_structure_assignment

  on_error, 2
  compile_opt idl2

  IF n_elements(filename) EQ 0 THEN message, 'must provide filename'

  restore, filename, description=desc, $
         relaxed_structure_assignment=relaxed_structure_assignment

  IF desc NE 'bh_idl_saved_variable' THEN message, $
    'this save file was not created with write_sav.pro'
  IF n_elements(var) EQ 0 THEN message, 'error restoring variable'

  return, var

END
```

Subject: Re: readf and structures
Posted by David Fanning on Tue, 10 Apr 2007 15:34:04 GMT
View Forum Message <> Reply to Message

Wox writes:

```
>  Is there an easy way of printing arrays of structures to a file and
>  read them in again, without reading the fields separate?
>
>  This doesn't work (also tried with specifying formats):
>
>  tout={a:0.,b:'name'}
>  tin={a:0.,b:''}
>
>  openw,lun,'c:\test.txt',/get_lun
>  printf,lun,tout
>  close,lun
>  free_lun,lun
>
>  openr,lun,'c:\test.txt',/get_lun
>  readf,lun,tin
>  close,lun
>  free_lun,lun
```

This basically doesn't work because you have a string field.
All variables in IDL have a defined length, EXCEPT strings.
Thus, unless you can tell IDL how long your strings are,
it just keeps reading until it gets to the end of an input
line.

One way to tell IDL how long your strings are is to use
the XDR format for input/output. Thus, these commands would
work for you:

```
   tout={a:0.,b:'name'}
   tin={a:0.,b:''}

   openw,lun,'test.txt',/get_lun, /xdr
   writeu,lun,tout
   free_lun,lun

   openr,lun,'test.txt',/get_lun, /xdr
   readu,lun,tin
   free_lun,lun
```

Note that there is no need to have a CLOSE statement in your
code. FREE_LUN both closes the logical unit and frees it up
for future use.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: readf and structures
Posted by Wox on Tue, 10 Apr 2007 15:38:33 GMT
View Forum Message <> Reply to Message

Ok, thanks for the suggestions guys. Didn't know about XDR.
Interesting!

Subject: Re: readf and structures
Posted by Benjamin Hornberger on Tue, 10 Apr 2007 16:40:29 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Wox writes:
>
>
>> Ok, thanks for the suggestions guys. Didn't know about XDR.
>> Interesting!
>
>
> Well, it also has the advantage in that it is a binary
> format that is totally portable across architectures. It
> even solves some of Benjamin's problems. :-)

On the other hand, this solution requires that you need to know the structure of the variable before you read it, and you have to set up a "blank" version of it in advance. For quick command-line stuff, the sav version saves you a lot of thinking and typing -- it just saves and restores any arbitrary variable with a single command. I use it a lot.

Cheers,
Benjamin

Subject: Re: readf and structures
Posted by David Fanning on Tue, 10 Apr 2007 17:30:22 GMT
View Forum Message <> Reply to Message

Wox writes:

> Ok, thanks for the suggestions guys. Didn't know about XDR.
> Interesting!

Well, it also has the advantage in that it is a binary format that is totally portable across architectures. It even solves some of Benjamin's problems. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")