Subject: Re: Library/Functions to write configuration file for application
Posted by penteado on Mon, 31 Jan 2011 19:44:48 GMT
View Forum Message <> Reply to Message

I have some very old routines I wrote for exactly that purpose. They worked with text files, recognizing a few comment markers, and optionally converting the values to numbers (reals or integers) if they are compatible with being numbers. I can tidy them up and publish them, maybe later tonight. They fit well into some classes I was preparing.

Another approach I used for the same purpose is NetCDF files, which nicely handle names dimensions and types, and can be easily put into hashes or structures. Writing takes slightly more work, due to the need to associate dimensions with the variables, but can also be nicely expressed by hashes.

On Jan 31, 5:05 pm, Robin Wilson <ro...@rtwilson.com> wrote:
> What I was wondering is whether there are any libraries/sets of
> functions in IDL that would help with reading and writing these
> configuration files? I know that a number of programming languages
> either have inbuilt functions for writing configuration files or some
> add-ons which do that - do any of these exist for IDL?
>
> I could knock up some simple file reading/writing code, but that would
> make it rather difficult if I changed the order/number of variables in
> the file, as I'd probably have to read some lines as strings and some as
> floats, ints etc, and I could see it getting very complicated and
> difficult to maintain.

Subject: Re: Library/Functions to write configuration file for application
Posted by penteado on Mon, 31 Jan 2011 19:47:55 GMT
View Forum Message <> Reply to Message

On Jan 31, 5:44 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
> I have some very old routines I wrote for exactly that purpose. They
> worked with text files, recognizing a few comment markers, and
> optionally converting the values to numbers (reals or integers) if
> they are compatible with being numbers. I can tidy them up and publish
> them, maybe later tonight. They fit well into some classes I was
> preparing.
>
> Another approach I used for the same purpose is NetCDF files, which
> nicely handle names dimensions and types, and can be easily put into
> hashes or structures. Writing takes slightly more work, due to the
> need to associate dimensions with the variables, but can also be
> nicely expressed by hashes.

By the way, those formats (text and NetCDF) were chosen for
portability with code in other languages. But if you are using only
IDL, maybe savefiles would be easier. Especially if handled through
Craig Markwardt's cmsvlib (http://www.physics.wisc.edu/~craigm/idl/
cmsave.html), or IDL_Savefile objects.

Subject: Re: Library/Functions to write configuration file for application
Posted by pgrigis on Mon, 31 Jan 2011 19:52:19 GMT
View Forum Message <> Reply to Message

On Jan 31, 2:05 pm, Robin Wilson <ro...@rtwilson.com> wrote:
> Hi,
>
> I've written a fairly complex piece of modelling software in IDL, and
> the model requires a number of parameters to run (input and output files
> as well as various numerical parameters).
>
> I want to be able to run this model from a batch file, and would like to
> have a configuration file in which I specify all of these parameters and
> then all I have to do is give the model the name of the configuration file.
>
> What I was wondering is whether there are any libraries/sets of
> functions in IDL that would help with reading and writing these
> configuration files? I know that a number of programming languages
> either have inbuilt functions for writing configuration files or some
> add-ons which do that - do any of these exist for IDL?
>
> I could knock up some simple file reading/writing code, but that would
> make it rather difficult if I changed the order/number of variables in
> the file, as I'd probably have to read some lines as strings and some as
> floats, ints etc, and I could see it getting very complicated and
> difficult to maintain.

Well, if the values can only be int, floats or strings, nothing
like a good old-fashioned text file in format

parname = parvalue
anotherparname = anothervalue

etc.

which is quite easy to parse.

Ciao,
Paolo

>
> Any ideas?
>
> Cheers,
>
> Robin
>
> ----------
> Robin Wilson
> A PhD student studying complexity in remote sensingwww.rtwilson.com/academic

---

Subject: Re: Library/Functions to write configuration file for application
Posted by penteado on Mon, 31 Jan 2011 21:08:59 GMT
View Forum Message <> Reply to Message

On Jan 31, 5:52 pm, Paolo <pgri...@gmail.com> wrote:
> Well, if the values can only be int, floats or strings, nothing
> like a good old-fashioned text file in format
>
> parname = parvalue
> anotherparname = anothervalue
>
> etc.
>
> which is quite easy to parse.

Yes, even for 1D arrays. I often had several parameters which were
arrays with more than 1D, so NetCDF was nicer for those.

Subject: Re: Library/Functions to write configuration file for application
Posted by Gray on Mon, 31 Jan 2011 21:09:13 GMT
View Forum Message <> Reply to Message

On Jan 31, 2:52 pm, Paolo <pgri...@gmail.com> wrote:
> On Jan 31, 2:05 pm, Robin Wilson <ro...@rtwilson.com> wrote:
>
>
>
>
>
>
>
>
>
>

>> Hi,
>
>> I've written a fairly complex piece of modelling software in IDL, and
>> the model requires a number of parameters to run (input and output files
>> as well as various numerical parameters).
>
>> I want to be able to run this model from a batch file, and would like to
>> have a configuration file in which I specify all of these parameters and
>> then all I have to do is give the model the name of the configuration file.
>
>> What I was wondering is whether there are any libraries/sets of
>> functions in IDL that would help with reading and writing these
>> configuration files? I know that a number of programming languages
>> either have inbuilt functions for writing configuration files or some
>> add-ons which do that - do any of these exist for IDL?
>
>> I could knock up some simple file reading/writing code, but that would
>> make it rather difficult if I changed the order/number of variables in
>> the file, as I'd probably have to read some lines as strings and some as
>> floats, ints etc, and I could see it getting very complicated and
>> difficult to maintain.
>
> Well, if the values can only be int, floats or strings, nothing
> like a good old-fashioned text file in format
>
> parname = parvalue
> anotherparname = anothervalue
>
> etc.
>
> which is quite easy to parse.
>
> Ciao,
> Paolo
>
>
>
>
>
>
>
>
>
>
>> Any ideas?
>
>> Cheers,
>
>> Robin

>
>> ----------
>> Robin Wilson
>> A PhD student studying complexity in remote sensingwww.rtwilson.com/academic

Or, something along the lines of a FITS header, which has a lot of
stuff written for parsing already. Check out the NASA astronomy IDL
library for examples.

Subject: Re: Library/Functions to write configuration file for application
Posted by Robin Wilson on Mon, 31 Jan 2011 22:01:49 GMT
View Forum Message <> Reply to Message

On 31/01/2011 19:44, Paulo Penteado wrote:
>
> I have some very old routines I wrote for exactly that purpose. They
> worked with text files, recognizing a few comment markers, and
> optionally converting the values to numbers (reals or integers) if
> they are compatible with being numbers. I can tidy them up and publish
> them, maybe later tonight. They fit well into some classes I was
> preparing.

Those routines sound great - it'd be brilliant if you could publish them :-)

Cheers,

Robin

Subject: Re: Library/Functions to write configuration file for application
Posted by Robin Wilson on Mon, 31 Jan 2011 22:04:35 GMT
View Forum Message <> Reply to Message

On 31/01/2011 19:52, Paolo wrote:>
> Well, if the values can only be int, floats or strings, nothing
> like a good old-fashioned text file in format
>
> parname = parvalue
> anotherparname = anothervalue
>
> etc.
>
> which is quite easy to parse.

This is the sort of format I was thinking of, but I wasn't sure how to
parse it in a nice way. 'Nice' in this context means that:

* It can cope with strings, integers and floats
* The order of the file either doesn't matter, or can be changed fairly
easily
* New parameter names can be added to the code fairly easily
* Comment characters can be used
* Preferably that I don't have to write loads and loads of IF statements
to parse the file correctly.

Is something along those lines fairly easy to do myself? Or should I
wait for Paulo's routines?

Cheers,

Robin

Subject: Re: Library/Functions to write configuration file for application
Posted by penteado on Mon, 31 Jan 2011 22:14:22 GMT
View Forum Message <> Reply to Message

On Jan 31, 8:04 pm, Robin Wilson <ro...@rtwilson.com> wrote:
> This is the sort of format I was thinking of, but I wasn't sure how to
> parse it in a nice way. 'Nice' in this context means that:
>
> * It can cope with strings, integers and floats
> * The order of the file either doesn't matter, or can be changed fairly
> easily
> * New parameter names can be added to the code fairly easily
> * Comment characters can be used
> * Preferably that I don't have to write loads and loads of IF statements
> to parse the file correctly.
>
> Is something along those lines fairly easy to do myself? Or should I
> wait for Paulo's routines?

That is pretty much what those text file routines do. I will clean
them up and post them later, and you can decide if they seem useful.

Subject: Re: Library/Functions to write configuration file for application
Posted by Karl[1] on Mon, 31 Jan 2011 23:55:27 GMT
View Forum Message <> Reply to Message

On Jan 31, 3:14 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
> On Jan 31, 8:04 pm, Robin Wilson <ro...@rtwilson.com> wrote:
>

>> This is the sort of format I was thinking of, but I wasn't sure how to
>> parse it in a nice way. 'Nice' in this context means that:
>
>> * It can cope with strings, integers and floats
>> * The order of the file either doesn't matter, or can be changed fairly
>> easily
>> * New parameter names can be added to the code fairly easily
>> * Comment characters can be used
>> * Preferably that I don't have to write loads and loads of IF statements
>> to parse the file correctly.
>
>> Is something along those lines fairly easy to do myself? Or should I
>> wait for Paulo's routines?
>
> That is pretty much what those text file routines do. I will clean
> them up and post them later, and you can decide if they seem useful.

XML is good for something like this.

Subject: Re: Library/Functions to write configuration file for application
Posted by sirvival on Tue, 01 Feb 2011 10:17:54 GMT
View Forum Message <> Reply to Message

Hi,
I got a pro (not sure from where) that maybe does what you want:

```
 ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ ++++++++++++
++++++++

FUNCTION DATA_INPUT, key, default, FILE=file

;_Purpose:      ASCII data input from runtime file specification
;
;_Description:  This is an extension of the FITS format such that a
single
;               keyword may imply input of up to 35 data items,
provided the
;               data are non-character, and they are read from a
single input
;               line, the length of which is < 81 characters.
;               Data delimiter is a comma. The number of data items
identified
;               is equal to the number of commas found + 1.
;               Character input is constrained to a single item.
;               A '!' marks the start of a comment.
```

```
;
;_Language:     IDL V3
;
;_Input Par:    key         Keyword (FITS type) to extract
;               default     Default value assigned if key or file not
found
;
;_Keywords:     FILE        Input from specified disk file
;                           Default is 'USER.PAR'
;
;_Result:       Value(s) of data following the keyword in the input
file. The
;               data type is determined at run time.
;
;_History:      TG          28-May-93   Initial programming
;               TG          10-Feb-96   '.' recognition produces FLOAT
;
;_Copyright:    (c) FOCES project, University Observatory Munich
;
 ;------------------------------------------------------------ ------------------

line = STRING(' ',FORMAT='(A80)')                       ; Single
line format

ON_IOERROR, no_such_file
IF (NOT KEYWORD_SET(file)) THEN file = 'ORIENTATION.PAR'         ;
Default input file
GET_LUN, lun
OPENR, lun, file
k = -1

WHILE (k EQ -1) AND NOT EOF(lun) DO BEGIN                    ; Read
until EOF
  READF, lun, line
  keyword = STRMID(line, 0, 8)
  k = STRPOS(keyword, key)
  IF (k NE -1) THEN line = STRMID(line, 10, 70)      ; Extract keyword
contents
ENDWHILE

FREE_LUN, lun
IF (k EQ -1) THEN BEGIN                                 ; Keyword
not found
  value = default                                      ;
Set default
  RETURN, value                         ; ... and return to
calling program
ENDIF
```

```
IF (STRPOS(line, """) EQ 0) THEN BEGIN                ; String value
detected
  tmp = STRPOS(line, """, 1)
  IF (tmp EQ -1) THEN BEGIN                  ; Second string
delimiter missing
    PRINT, '###ERROR: non-FITS format'
    PRINT, line
    STOP
  ENDIF
  value = STRMID(line, 1, tmp-1)                          ;
Extract string
ENDIF ELSE BEGIN                                  ; Number(s)
detected
  i_elem = 0
  k = -1
  i = 0
  WHILE (i NE -1) DO BEGIN
    i = STRPOS(line, ",", k+1)                        ; Detect
next comma
    j = STRPOS(line, "!", k+1)                    ; Detect
exclamation mark
    ii = i
    IF (ii EQ -1) THEN BEGIN
      ii = STRLEN(line)                             ; End of
input line
      IF (j NE -1) THEN ii = j-1                  ; End of input
with comment
    ENDIF
    tmp = STRMID(line, k+1, ii-k-1)          ; Extract substring
between commas
    CASE (1) OF                                   ; Determine
data type
      (STRPOS(tmp, 'D') NE -1): tmp = DOUBLE(tmp)
      (STRPOS(tmp, 'E') NE -1): tmp = FLOAT(tmp)
      (STRPOS(tmp, '.') NE -1): tmp = FLOAT(tmp)
      ELSE:                     tmp = LONG(tmp)
    ENDCASE
    i_elem = i_elem+1
    IF (i_elem EQ 1) THEN value = tmp ELSE value = [value,tmp]
    k = i
  ENDWHILE
ENDELSE
RETURN, value

no_such_file:                                     ; File
not found
ON_IOERROR, NULL                                      ; Reset
```

error status
value = default ;
Set default
RETURN, value

END

e.g. name.par

!Input file for the orientation simulation giving star and
spectrograph parameters
!
Dtel = 1.2
Fnum = 13.0
Groove = 31.6e3

---------- --------
a short introduction to the parameters

#Dtel : Diameter of the telescope in m
#Fnum : Fnumber of the telescope
#groove : of the grating in m

Subject: Re: Library/Functions to write configuration file for application
Posted by penteado on Tue, 01 Feb 2011 17:32:22 GMT
View Forum Message <> Reply to Message

On Jan 31, 8:14 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
> That is pretty much what those text file routines do. I will clean
> them up and post them later, and you can decide if they seem useful.

You can find them at

http://www.ppenteado.net/idl/pp_lib/doc/pp_readpars.html
http://www.ppenteado.net/idl/pp_lib/doc/pp_writepars.html

Subject: Re: Library/Functions to write configuration file for application
Posted by Robin Wilson on Tue, 01 Feb 2011 22:51:39 GMT
View Forum Message <> Reply to Message

Paulo,

Thank you very much - they look like just what I was looking for.

Just one minor problem: my university hasn't yet got IDL 8, so I don't have access to hashes (I believe they were introduced in version 8). I'll have a look at the code tomorrow and see if I can convert it to use anonymous structures.

Thanks again,

Robin

On 01/02/2011 17:32, Paulo Penteado wrote:
> On Jan 31, 8:14 pm, Paulo Penteado<pp.pente...@gmail.com> wrote:
>> That is pretty much what those text file routines do. I will clean
>> them up and post them later, and you can decide if they seem useful.
>
> You can find them at
>
> http://www.ppenteado.net/idl/pp_lib/doc/pp_readpars.html
> http://www.ppenteado.net/idl/pp_lib/doc/pp_writepars.html
