Subject: writing fixed-length string arrays to netcdf-4
Posted by Chris O'Dell on Wed, 30 Oct 2013 03:11:03 GMT
View Forum Message <> Reply to Message

Does anyone know if it is possible to write an array of strings to a variable in a netCDF-4 file as some fixed-length string that has a length > 1? Let me show what I mean, and preface this by saying that I understand HDF-5 better, and in HDf-5 you can definitely do this. I understand that netCDF-4 is built on HDF-5.

Let's define a string array:
stringarr = ['Renee','Tyler','Susan']

METHOD 1 for writing to netcdf file:
(length=1 CHAR array)

```
------------------------------------
id = NCDF_CREATE('myfile.nc', /CLOBBER, /NETCDF4_FORMAT)
dim1 = NCDF_DIMDEF(id,'dim1',n_elements(stringarr))
len_dim = NCDF_DIMDEF(id, 'string length', max(strlen(stringarr)))
vid = NCDF_VARDEF(id, 'Names', [len_dim, dim1], /CHAR)
NCDF_CONTROL, id, /ENDEF
NCDF_VARPUT, id, vid, stringarr
NCDF_CLOSE, id
```

This results in a 2D array of CHAR, with dimension (5,3).

METHOD 2 for writing to netcdf file:
(variable-length STRING)

```
------------------------------------
id = NCDF_CREATE('myfile.nc', /CLOBBER, /NETCDF4_FORMAT)
dim1 = NCDF_DIMDEF(id,'dim1',n_elements(stringarr))
vid = NCDF_VARDEF(id, 'Names', [dim1], /STRING)
NCDF_CONTROL, id, /ENDEF
NCDF_VARPUT, id, vid, stringarr
NCDF_CLOSE, id
```

This results in a 1D array of variable-length strings, with dimension (3).

What I really want is a 1D array of fixed-length strings, each with length=5 (in this example). Anyone know if this is possible?

Thanks!
Chris

Subject: Re: writing fixed-length string arrays to netcdf-4
Posted by David Fanning on Wed, 30 Oct 2013 04:13:18 GMT
View Forum Message <> Reply to Message

Chris O'Dell writes:

> What I really want is a 1D array of fixed-length strings, each with length=5 (in this example). Anyone know if this is possible?

Why don't you just make them fixed length strings before you write them?

Cheers,

David

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

Subject: Re: writing fixed-length string arrays to netcdf-4
Posted by Chris O'Dell on Wed, 30 Oct 2013 15:49:28 GMT

View Forum Message <> Reply to Message

On Tuesday, October 29, 2013 10:13:18 PM UTC-6, David Fanning wrote:
> Chris O'Dell writes:
>
>
>
>> What I really want is a 1D array of fixed-length strings, each with length=5 (in this example). Anyone know if this is possible?
>
>
>
> Why don't you just make them fixed length strings before you write them?
>
>
>
> Cheers,
>
>
>
> David
>
>
>
>
>
> --

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

Hi David,

Can you explain how? I seem to be missing this corner of IDL knowledge.

Thanks,
Chris

Subject: Re: writing fixed-length string arrays to netcdf-4
Posted by David Fanning on Wed, 30 Oct 2013 16:07:39 GMT

View Forum Message <> Reply to Message

Chris O'Dell writes:

> Can you explain how? I seem to be missing this corner of IDL knowledge.

```
sarray = ['Bob', 'Coyote', 'Duck']
flensarray = String(sarray, Format='(A8)')
for j=0,2 do print, 'Length of String: ', StrLen(flensarray[j])
```

Cheers,

David

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

Subject: Re: writing fixed-length string arrays to netcdf-4
Posted by David Fanning on Mon, 04 Nov 2013 18:21:32 GMT

View Forum Message <> Reply to Message

Chris O'Dell writes:

> Nice try but no cigar. I tried that but honestly they were already fixed length strings in IDL

anyway (in my case, length=14). NetCDF simply doesn't care. You have to somehow tell it to make the variable in the output netCDF file be an array of fixed-length strings, ideally through the NCDF_VARDEF command. But I don't know how, and there doesn't seem to be any documentation anywhere that explains it.
>
> Like I said, you can define the netCDF variable as a 2D character array (using /CHAR in the call to NCDF_VARDEF), or as an array of variable-length strings (using the /STRING keyword in the call to NCDF_VARDEF).
>
> If you do get something to work, and get HDFView to tell you the data type in the output netCDF file is "String, length = 8" (or whatever your string length is, so long as it is > 1) please post the code to do so!

Well, I don't know. My NCDF_File object seems to do it just fine.

```
 ;---------------------------------------------------------- -----------
; Create a ncdf file with variable length and fixed length strings.
fileObj = Obj_New('ncdf_file', 'test.nc', /Clobber, /Create)
fileObj -> WriteDim, 'length', [10], OBJECT=dimObj
v1 = 'cat'
v2 = 'coyote'
v3 = 'elephant'
v4 = String(v1, Format='(A10)')
v5 = String(v2, Format='(A10)')
v6 = String(v3, Format='(A10)')
fileObj -> WriteVarDef, 'v11', 'length', DATATYPE='STRING', OBJECT=v1Obj
fileObj -> WriteVarDef, 'v22', 'length', DATATYPE='STRING', OBJECT=v2Obj
fileObj -> WriteVarDef, 'v33', 'length', DATATYPE='STRING', OBJECT=v3Obj
fileObj -> WriteVarDef, 'v44', 'length', DATATYPE='STRING', OBJECT=v4Obj
fileObj -> WriteVarDef, 'v55', 'length', DATATYPE='STRING', OBJECT=v5Obj
fileObj -> WriteVarDef, 'v66', 'length', DATATYPE='STRING', OBJECT=v6Obj

fileObj -> WriteVarData, v1Obj, v1
fileObj -> WriteVarData, v2Obj, v2
fileObj -> WriteVarData, v3Obj, v3
fileObj -> WriteVarData, v4Obj, v4
fileObj -> WriteVarData, v5Obj, v5
fileObj -> WriteVarData, v6Obj, v6

fileObj -> Sync
Obj_Destroy, fileObj


; Read the data out of the file.
fObj = Obj_New('ncdf_file', 'test.nc')
v_1 = fObj -> GetVarData('v11')
v_4 = fObj -> GetVarData('v44')
Help, v_1, v_4
```

```
v_2 = fObj -> GetVarData('v22')
v_5 = fObj -> GetVarData('v55')
Help, v_2, v_5
v_3 = fObj -> GetVarData('v33')
v_6 = fObj -> GetVarData('v66')
Help, v_3, v_6

END
 ;--------------------------------------------------------- ----------
```

Here are the results I get when I run the code above:

```
IDL> .go
V_1             STRING    = 'cat'
V_4             STRING    = '       cat'
V_2             STRING    = 'coyote'
V_5             STRING    = '    coyote'
V_3             STRING    = 'elephant'
V_6             STRING    = '  elephant'
```

Cheers,

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

Subject: Re: writing fixed-length string arrays to netcdf-4
Posted by David Fanning on Mon, 04 Nov 2013 18:48:45 GMT

View Forum Message <> Reply to Message

David Fanning writes:

> Well, I don't know. My NCDF_File object seems to do it just fine.

Here is essentially the same program, but using string arrays.

```
 ;--------------------------------------------------------- ----------
; Create a ncdf file with variable length and fixed length strings.
fileObj = Obj_New('ncdf_file', 'tester.nc', /Clobber, /Create)
fileObj -> WriteDim, 'xdim', [10], OBJECT=xdimObj
fileObj -> WriteDim, 'ydim', [3], OBJECT=ydimObj
dimNames = [xdimObj->GetName(), ydimObj->GetName()]
v1 = ['cat','coyote','elephant']
v2 = String(v1, Format='(A10)')
```

```
fileObj -> WriteVarDef, 'v11', dimNames, DATATYPE='STRING', OBJECT=v1Obj
fileObj -> WriteVarDef, 'v22', dimNames, DATATYPE='STRING', OBJECT=v2Obj

fileObj -> WriteVarData, v1Obj, v1
fileObj -> WriteVarData, v2Obj, v2

fileObj -> Sync
Obj_Destroy, fileObj


; Read the data out of the file.
fObj = Obj_New('ncdf_file', 'tester.nc')
v_1 = fObj -> GetVarData('v11')
v_2 = fObj -> GetVarData('v22')
Help, v_1, v_2
Obj_Destroy, fObj
END
 ;---------------------------------------------------------- ----------
```

Here is what I get when I run it.

```
IDL> .go
V_1             STRING    = Array[3]
V_2             STRING    = Array[3]
IDL> print, v_1
cat coyote elephant
IDL> print, v_2
     cat     coyote   elephant
```

Cheers,

David

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

Subject: Re: writing fixed-length string arrays to netcdf-4
Posted by David Fanning on Mon, 04 Nov 2013 19:15:07 GMT

View Forum Message <> Reply to Message

By the way, I notice now you specifically were asking about netCDF-4
files. To create these kinds of files with my ncdf_file object you have
to set the /NETCDF4_FORMAT keyword when you open the file for writing.
Doing so doesn't affect the results of either of the two programs I sent

earlier.

```
  fileObj = Obj_New('ncdf_file', 'tester.nc', /Clobber, $
                /Create, /NETCDF4_FORMAT)
```

Cheers,

David

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

Subject: Re: writing fixed-length string arrays to netcdf-4
Posted by Chris O'Dell on Tue, 05 Nov 2013 18:34:58 GMT

View Forum Message <> Reply to Message

Thanks David! However, your program seems to require your special NetCDF libraries, which I downloaded and installed so I could run your test program. However, it still produces the same fundamental netCDF objects inside the actual netcdf file. The reason they come out as string arrays when you read them seems to be because of logic you have built into your netCDF reader. Something like, "When you encounter a field that is a n-dimensional array of characters (length 1 strings), interpret this as an array of strings, where the string length is the size of the first dimension". Something like that.

To prove this to you, I ran your program and took a screenshot of HDFView's look at the output file; HDFView (as you likely know) is a standard reader for HDF files which also works on netCDF-4 (which itself is built on HDF-5). The output shows that the internal storage is actually a 2D array of single characters. Which sounds fine, except I know from past experience that HDF-5 does support fixed-length strings of length > 1. I assumed that netCDF-4 did as well, but perhaps not. I really don't know. Maybe this is the way it has to be internally stored.

I will email you the screenshot directly , as I don't see a way to do attachments in this forum.

Thanks again,
Chris

On Monday, November 4, 2013 12:15:07 PM UTC-7, David Fanning wrote:
> By the way, I notice now you specifically were asking about netCDF-4
>
> files. To create these kinds of files with my ncdf_file object you have
>
> to set the /NETCDF4_FORMAT keyword when you open the file for writing.
>

> Doing so doesn't affect the results of either of the two programs I sent
>
> earlier.
>
>
>
>   fileObj = Obj_New('ncdf_file', 'tester.nc', /Clobber, $
>
>   /Create, /NETCDF4_FORMAT)
>
>
>
> Cheers,
>
>
>
> David
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.

Subject: Re: writing fixed-length string arrays to netcdf-4
Posted by David Fanning on Tue, 05 Nov 2013 18:50:42 GMT
View Forum Message <> Reply to Message

Chris O'Dell writes:

> Thanks David! However, your program seems to require your special NetCDF libraries, which I downloaded and installed so I could run your test program. However, it still produces the same fundamental netCDF objects inside the actual netcdf file. The reason they come out as string arrays when you read them seems to be because of logic you have built into your netCDF reader. Something like, "When you encounter a field that is a n-dimensional array of characters (length 1 strings), interpret this as an array of strings, where the string length is the size of the first dimension". Something like that.

Well, I think I wrote this code in 2007, which is the last time I worked extensively with netCDF files. I wrote the code because I couldn't make any sense at all of the IDL documentation and I thought I could write

something that worked better. As far as I can tell, the documentation
hasn't improved. And, generally speaking, I've found that my code always
works, so I've just continued using it. If I built logic into it, it
must have been because the code was missing it in the original. ;-)

Cheers,

David

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

Subject: Re: writing fixed-length string arrays to netcdf-4
Posted by Chris O'Dell on Tue, 12 Nov 2013 00:17:48 GMT

View Forum Message <> Reply to Message

On Tuesday, November 5, 2013 11:50:42 AM UTC-7, David Fanning wrote:
> Chris O'Dell writes:
>
>
>
>> Thanks David! However, your program seems to require your special NetCDF libraries, which I downloaded and installed so I could run your test program. However, it still produces the same fundamental netCDF objects inside the actual netcdf file. The reason they come out as string arrays when you read them seems to be because of logic you have built into your netCDF reader. Something like, "When you encounter a field that is a n-dimensional array of characters (length
>
> 1 strings), interpret this as an array of strings, where the string length is the size of the first dimension". Something like that.
>
>
>
> Well, I think I wrote this code in 2007, which is the last time I worked
>
> extensively with netCDF files. I wrote the code because I couldn't make
>
> any sense at all of the IDL documentation and I thought I could write
>
> something that worked better. As far as I can tell, the documentation
>
> hasn't improved. And, generally speaking, I've found that my code always
>

> works, so I've just continued using it. If I built logic into it, it
>
> must have been because the code was missing it in the original. ;-)
>
>
>
> Cheers,
>
>
>
> David
>
>
>
>
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Sorry it took so long to reply. I don't check this every day and it shows. Thanks for putting some thought into this, but this is about as far as I've gotten. It's good to know about your netCDF software though, thanks!

Chris