

---

Subject: Re: I have the craziest idea...

Posted by [Paul van Delst](#) on Wed, 14 Feb 2001 20:27:51 GMT

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

---

tbowers wrote:

>  
> I've written a widget program that I've tried to design as a general data  
> import interface.

Cool.

<snip>

> Feedback would be appreciated.

Your snippet of code under the EXAMPLE: heading scared me away. I would pare it down to:

```
; slmported = ngDataImporter()
;
; ;//Let's say the data imported was a US ARMY GMS 3D grid file
; IDL> help, slmported, /STRUCT
; ** Structure <1354308>, 13 tags, length=2022472, refs=1:
; DATA      DOUBLE  Array[101, 61, 41]
; XDATA      DOUBLE  Array[101]
; YDATA      DOUBLE  Array[61]
; ZDATA      DOUBLE  Array[41]
; TDATA      BYTE    0
; ANCILLARYDATA  BYTE    0
; DATANAME   STRING  'c532(1/m)'
; FILENAME   STRING  Array[1]
; DATACLASS STRING  'SCAL_GRID_3S0T'
; DATATYPE   STRING  'USA_GMS3DG'
; POLYLINES  BYTE    0
; ERROR      BYTE    0
; CANCEL     BYTE    0
```

So how would you enable it to read netCDF and HDF data files and all their associated attributes? Would that be a simple addition? I have a netCDF reader function, that returns data in a structure sorta like your, to plug in if you'd like to try it. I'm sure others out there have similar for HDF.

paulv

--

Paul van Delst           A little learning is a dangerous thing;  
CIMSS @ NOAA/NCEP       Drink deep, or taste not the Pierian spring;  
Ph: (301)763-8000 x7274  There shallow draughts intoxicate the brain,  
Fax:(301)763-8545       And drinking largely sobers us again.

---

Subject: Re: I have the craziest idea...

Posted by [Martin Schultz](#) on Fri, 16 Feb 2001 11:02:31 GMT

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

---

tbowers wrote:

- >
- > I've written a widget program that I've tried to design as a general data
- > import interface. What I mean by general is that it's a wrapper widget to
- > import various types of data and present it back to the user in some kind of
- > standardized format for easy use. E.g., I have some 3D gridded data in ASCII
- > text files of one format, 2D scattered data in ASCII text files of another
- > format, etc. etc. What I did was write the widget as a function call so all
- > I have to do is call it and I always have access to ALL data files that I
- > know how to read. And, more importantly, the data is passed back to me in a
- > format that's consistent and easy to process and visualize in IDL.
- >
- > So, what do all of you think about the crazy idea of me putting this up on a
- > server and all of us passing this around, adding code where we can so that
- > we all have a centralized function that'll read many data formats? Seems to
- > me that we all have our specialty, and when one of us wants to read in a new
- > type we just come to this newsgroup for the answer anyway. I call it the
- > ngDataImporter. I usually put my initials in front of the name of each
- > function so I don't ever step on other functions. Since this would be a
- > group effort by us all, I just fronted it with ng for newsgroup, the IDL
- > newsgroup users.
- >
- > From the intro comments: [snipped][color]

Excellent idea, but from glancing over it, I am not sure that the structure is general and flexible enough. I started with a similar effort based first on the netcdf format and its ideas of metadata. Unfortunately I haven't gotten too far yet (because people don't let me spend so much time on IDL programming here), but I do have a working version of a netcdf reader, all object oriented, and with a class hierarchy that would allow just what you propose. It is based on a generic "basefile" object (of course with initials up front ;-)) and that object already does a lot of the work like opening the file, closing it, searching for it, etc. The idea is then to inherit from this object and implement only the file format dependent stuff and put the data into the generic format that is used by the generic object - and these are, yes: variables implemented as objects. The concept is most powerful, but apparently relatively hard to realize, especially when I work on it only one or two evenings in a period of 2-3 months. ;-( If you are interested, I can send you the current state of the development (as I say: reading netcdf already works, and I also have

the beginnings of a graphical viewer that can take the variable objects that are generated by the data reader and plot them as contours etc. (only 2D data so far)).

A few more general issues in terms of such a project: If you really want a tool that almost everyone on this newsgroup can use, you need to make sure that you can access portions of the data in the file upon request. For example, here we often deal with model output on the order of 2GB (and it could be larger were there not the Linux filesystem limitations), and you certainly don't want to read the complete file before starting to work with these data. Now, this requires a program structure where you analyse the headers first, let the user select what he/she wants to read, and then go out and grab it. This idea is what prompted me to mirror the netcdf attribute structure in my variable objects, so that these objects contain all the information about the data (the "metadata"), and you could even reproduce a netcdf file (or convert it to an HDF file) with the information you have stored. And then it's only a small step to have a file converter in addition to the reader program, which, I am sure, many of us need almost as badly!

... glancing over this again, I must be out of my mind to give you such a hard time when you are proposing such a good thing. I guess, at this point, any tool that is easy to handle (so that even a scientists can use it ;-)) and flexible enough, would already be most welcome.

Good luck,

Martin

--

```

[[ Dr. Martin Schultz  Max-Planck-Institut fuer Meteorologie  [[
[[          Bundesstr. 55, 20146 Hamburg          [[
[[          phone: +49 40 41173-308          [[
[[          fax: +49 40 41173-298          [[
[[ martin.schultz@dkrz.de          [[

```

---

Subject: Re: I have the craziest idea...  
Posted by [T Bowers](#) on Fri, 16 Feb 2001 16:59:34 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

> Excellent idea, but from glancing over it, I am not sure that the  
> structure is general and flexible enough. I started with a similar

That's exactly why I begged for comments. I knew special types like netCDF and HDF would be different, but I've never worked with either and I need input from people who know about those types.

> Unfortunately I haven't gotten too far yet (because people don't let  
> me spend so much time on IDL programming here), but I do have a

I know the feeling. I'm doing this on my own time. Just thought it'd be very useful to us all.

> A few more general issues in terms of such a project: If you really  
> want a tool that almost everyone on this newsgroup can use, you need  
> to make sure that you can access portions of the data in the file upon  
> request. For example, here we often deal with model output on the  
> order of 2GB (and it could be larger were there not the Linux  
> filesystem limitations), and you certainly don't want to read the  
> complete file before starting to work with these data. Now, this  
> requires a program structure where you analyse the headers first, let  
> the user select what he/she wants to read, and then go out and grab

Yes. I have a similar situation. I use a text file format which has h lines of header followed by n columns of scattered data where the columns have 1 or more of position/time data (eg. Lat Lon depth) and then 1 to 100's of columns of parameter data (temp, salinity, etc.) but you never know which column contains what because there's no standard, depth could be in column 1 or 21 or 101. So, I quickly wrote a little front end widget that first presents to the user cw\_field()'s that prompt for the data to extract.

Similar to:

Number of Header Lines: \_\_\_\_  
X Axis Data Column: \_\_\_\_  
Y Axis Data Column: \_\_\_\_  
Z Axis Data Column: \_\_\_\_  
T Axis Data Column: \_\_\_\_  
Variable Data Column: \_\_\_\_

\_\_\_\_\_  
|Accept | Cancel|  
=====

The user types in what he wants to extract, clicks Accept button, and that data is read in. The dimensionality of the data is determined by how many X,Y,Z,T fields the user filled out.

> ... glancing over this again, I must be out of my mind to give you  
> such a hard time when you are proposing such a good thing. I guess, at

Absolutely not. Honest constructive criticism (with helpful suggestions,

of course ;) ) is what I'm looking for. Thanks. I'm kinda surprised more people don't have an opinion on this.

Since I'm a netCDF/HDF moron, I don't think I could easily decide how to handle/integrate your data reader. It may be easy, but I don't know. How about I send you my code and you send me yours. I'll look at it this weekend. Maybe together we can determine how easy/difficult this will be.

I also don't want to discourage others. It's easy to see that netCDF/HDF are special cases. I'm sure there are a lot of file formats that could very easily integrate into this code (e.g. FITS, Surfer grid, Sigma Plot grid, medical image formats, etc.) formats which I have no experience with but others do. Also, not all netCDF/HDF files are so general. E.g., SeaWiFs satellite imagery comes in an HDF file for each image. This, it seems, could be returned rather easily as a standard 2D image type.

Thanks Martin. Your comments and advice is greatly appreciated.  
Todd

---

Subject: Re: I have the craziest idea...  
Posted by [T Bowers](#) on Fri, 16 Feb 2001 17:03:53 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

> So how would you enable it to read netCDF and HDF data files and all their associated  
> attributes? Would that be a simple addition? I have a netCDF reader function, that returns

This brings up a good point. I tried to put this together so it is easily enhanced. If you have code that already reads a type of data, then all that needs to be done is to insert your call in the proper place. If you need more fields than the present return data structure allows, just add a new field. I don't care, just as long as people don't start haphazardly adding fields that aren't necessary so that it becomes unwieldy. But, I had thought about HDF and netCDF and thought they'd be prime candidates for possibly more fields. I just don't really know much about those formats. Never used them so I couldn't 'pre-plan' for them. I was hoping someone with netCDF/HDF knowledge would help out.

> Your snippet of code under the EXAMPLE: heading scared me away. I would pare it down to:

wimp. ;) I know, it's a bit outta hand, but I'm trying to emphasize to contributors that they need to think about how their data will be used on return.

Thanks for the input,  
Todd

---