Subject: RESTORE-like function using H5
Posted by Nikola on Fri, 09 Dec 2016 23:47:49 GMT
View Forum Message <> Reply to Message

I made save- and restore-like functions using H5 instead of the xdr format. To resolve input variable names in saveh5, I used SCOPE_VARNAMES and that works fine. On the other hand, my restoreh5 is a function returning a structure. Just for curiosity, what is the trick with the native RESTORE procedure? How does it make the loaded variables available in memory after the execution? It behaves like a batch file that takes some input (filename, keywords) or as a procedure with a silent STOP before the END.

Subject: Re: RESTORE-like function using H5
Posted by Jim Pendleton on Sat, 10 Dec 2016 00:20:49 GMT
View Forum Message <> Reply to Message

On Friday, December 9, 2016 at 4:47:51 PM UTC-7, Nikola Vitas wrote:
> I made save- and restore-like functions using H5 instead of the xdr format. To resolve input variable names in saveh5, I used SCOPE_VARNAMES and that works fine. On the other hand, my restoreh5 is a function returning a structure. Just for curiosity, what is the trick with the native RESTORE procedure? How does it make the loaded variables available in memory after the execution? It behaves like a batch file that takes some input (filename, keywords) or as a procedure with a silent STOP before the END.

RESTORE is part of the C-level API (rather than .pro code) so it is able to create variables in whatever scope it wants. It is probably using the equivalent of the exposed entry point IDL_FindNamedVariable(). (See http://www.harrisgeospatial.com/docs/LookUpVariablesCurrentScope.html).

Jim P.

Subject: Re: RESTORE-like function using H5
Posted by Michael Galloy on Sat, 10 Dec 2016 04:30:59 GMT
View Forum Message <> Reply to Message

On 12/9/16 4:47 PM, Nikola Vitas wrote:
> I made save- and restore-like functions using H5 instead of the xdr
> format. To resolve input variable names in saveh5, I used
> SCOPE_VARNAMES and that works fine. On the other hand, my restoreh5
> is a function returning a structure. Just for curiosity, what is the
> trick with the native RESTORE procedure? How does it make the loaded
> variables available in memory after the execution? It behaves like a
> batch file that takes some input (filename, keywords) or as a
> procedure with a silent STOP before the END.
>

You can use SCOPE_VARFETCH on the restore side to insert a variable into
a particular variable name in a particular scope. You can do something
like the following on the command line:

```
IDL> (scope_varfetch('data', /enter, level=1)) = findgen(10)
IDL> help, data
DATA            FLOAT     = Array[10]
```

You'll just have to set LEVEL correctly, check out the docs.

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)

Subject: Re: RESTORE-like function using H5
Posted by Nikola on Sat, 10 Dec 2016 09:21:32 GMT

View Forum Message <> Reply to Message

On Saturday, December 10, 2016 at 4:31:04 AM UTC, Michael Galloy wrote:

```
> On 12/9/16 4:47 PM, Nikola Vitas wrote:
>> I made save- and restore-like functions using H5 instead of the xdr
>> format. To resolve input variable names in saveh5, I used
>> SCOPE_VARNAMES and that works fine. On the other hand, my restoreh5
>> is a function returning a structure. Just for curiosity, what is the
>> trick with the native RESTORE procedure? How does it make the loaded
>> variables available in memory after the execution? It behaves like a
>> batch file that takes some input (filename, keywords) or as a
>> procedure with a silent STOP before the END.
>>
>
> You can use SCOPE_VARFETCH on the restore side to insert a variable into
> a particular variable name in a particular scope. You can do something
> like the following on the command line:
>
> IDL> (scope_varfetch('data', /enter, level=1)) = findgen(10)
> IDL> help, data
> DATA            FLOAT     = Array[10]
>
> You'll just have to set LEVEL correctly, check out the docs.
>
> Mike
> --
> Michael Galloy
> www.michaelgalloy.com
> Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
```

Many thanks for the reply, Michael. It's the first time I'm using any of the scope functions. They seem to be my new favorites of IDL.

Subject: Re: RESTORE-like function using H5
Posted by Sergey Anfinogentov on Sat, 10 Dec 2016 12:12:02 GMT
View Forum Message <> Reply to Message

> I made save- and restore-like functions using H5 instead of the xdr format. To resolve input variable names in saveh5, I used SCOPE_VARNAMES and that works fine. On the other hand, my restoreh5 is a function returning a structure. Just for curiosity, what is the trick with the native RESTORE procedure? How does it make the loaded variables available in memory after the execution? It behaves like a batch file that takes some input (filename, keywords) or as a procedure with a silent STOP before the END.

Such functions can be useful for other people. Do you plan to make them publicly available?

Subject: Re: RESTORE-like function using H5
Posted by Michael Galloy on Sun, 11 Dec 2016 18:41:55 GMT
View Forum Message <> Reply to Message

On 12/10/16 2:21 AM, Nikola Vitas wrote:
> Many thanks for the reply, Michael. It's the first time I'm using any
> of the scope functions. They seem to be my new favorites of IDL.

A word of warning: these routines are useful for some things (like command line use or "Import to commandline" functionality), but overuse can make code difficult to deal with.

For example, creating variables by restoring a Save file can be useful on the command line, but in a procedure/function it can be difficult to determine where a variable came from. Compare using RESTORE:

```
  restore, 'mysavefile.sav'
  ; my_var pops into existence without any mention of its name
```

with my library routine to deal with Save files:

```
  my_var = mg_save_getdata('mysavefile.sav', 'my_var')
```

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)

Subject: Re: RESTORE-like function using H5
Posted by Nikola on Mon, 12 Dec 2016 15:37:17 GMT
View Forum Message <> Reply to Message

On Sunday, December 11, 2016 at 6:42:00 PM UTC, Michael Galloy wrote:

> A word of warning: these routines are useful for some things (like
> command line use or "Import to commandline" functionality), but overuse
> can make code difficult to deal with.
>

Thanks for the warning, Michael! I'll keep it in mind.

> Such functions can be useful for other people. Do you plan to make them publicly available?

Sergey, if I make them reliable enough, I will certainly make them public. But don't expect anything fancy. The only motivation is to combine easy-to-use of save/restore in IDL with a file format that I can easily read from MPI/Fortran.