Subject: accessing IDL internal (aka "built-in") routines
Posted by Tom Van Niel on Mon, 13 Jan 2014 05:11:44 GMT
View Forum Message <> Reply to Message

Hi All,

Has anybody figured out how to access internal IDL routines?

More specifically, I am trying to run MAP_PROJ_INIT in GDL, but get the following error:

“MAP_PROJ_INIT: Procedure not found: MAP_PROJ_GCTP_FORINIT”

MAP_PROJ_INIT.pro calls MAP_PROJ_GCTP_FORINIT on line 1195, which turns out to be an internal IDL code (which obviously has not been translated into GDL). I've tried things like "Routine_Info", and "which" to find out the location of MAP_PROJ_GCTP_FORINIT, but with no luck. I guess internal files are embedded in the executable.

If anybody has found a way to access internal routines or by chance has a copy of MAP_PROJ_GCTP_FORINIT, please let me know.

Regards,
Tom

Subject: Re: accessing IDL internal (aka "built-in") routines
Posted by Haje Korth on Mon, 13 Jan 2014 14:23:43 GMT
View Forum Message <> Reply to Message

There are a number of low-level routines of the map projection engine in a DLM. Without special skills, you have no change of getting at those, and you would probably be on soft legal grounds if you could. As people say, you get what you pay for. The your time and effort required for solving this problem would likely exceed the cost of an IDL license. Life's too short, use your time wisely!

On Monday, January 13, 2014 12:11:44 AM UTC-5, Tom Van Niel wrote:
> Hi All,
>
>
>
> Has anybody figured out how to access internal IDL routines?
>
>
>
> More specifically, I am trying to run MAP_PROJ_INIT in GDL, but get the following error:
>
>
>
> “MAP_PROJ_INIT: Procedure not found: MAP_PROJ_GCTP_FORINIT”
>

>
>
> MAP_PROJ_INIT.pro calls MAP_PROJ_GCTP_FORINIT on line 1195, which turns out to be an internal IDL code (which obviously has not been translated into GDL). I've tried things like "Routine_Info", and "which" to find out the location of MAP_PROJ_GCTP_FORINIT, but with no luck. I guess internal files are embedded in the executable.
>
>
>
> If anybody has found a way to access internal routines or by chance has a copy of MAP_PROJ_GCTP_FORINIT, please let me know.
>
>
>
> Regards,
>
> Tom

Subject: Re: accessing IDL internal (aka "built-in") routines
Posted by Haje Korth on Mon, 13 Jan 2014 14:30:10 GMT
View Forum Message <> Reply to Message

There are a number of low-level routines of the map projection engine in a DLM. Without special skills, you have no chance of getting at those, and you would probably be on soft legal grounds if you could. As people say, you get what you pay for. Your time and effort required for solving this problem would likely exceed the cost of an IDL license. Life's too short, use your time wisely!

On Monday, January 13, 2014 12:11:44 AM UTC-5, Tom Van Niel wrote:
> Hi All,
>
>
>
> Has anybody figured out how to access internal IDL routines?
>
>
>
> More specifically, I am trying to run MAP_PROJ_INIT in GDL, but get the following error:
>
>
>
> “MAP_PROJ_INIT: Procedure not found: MAP_PROJ_GCTP_FORINIT”
>
>
>
> MAP_PROJ_INIT.pro calls MAP_PROJ_GCTP_FORINIT on line 1195, which turns out to be an internal IDL code (which obviously has not been translated into GDL). I've tried things like "Routine_Info", and "which" to find out the location of MAP_PROJ_GCTP_FORINIT, but with no

luck. I guess internal files are embedded in the executable.
>
>
>
> If anybody has found a way to access internal routines or by chance has a copy of MAP_PROJ_GCTP_FORINIT, please let me know.
>
>
>
> Regards,
>
> Tom

Subject: Re: accessing IDL internal (aka "built-in") routines
Posted by luke.a.domanski on Mon, 13 Jan 2014 23:52:24 GMT
View Forum Message <> Reply to Message

On Tuesday, January 14, 2014 1:30:10 AM UTC+11, Haje Korth wrote:
> As people say, you get what you pay for. Your time and effort required for solving this problem would likely exceed the cost of an IDL license. Life's too short, use your time wisely!

Thanks Haje,

But unfortunately your reasoning is not entirely applicable in this case. We have plenty of IDL licenses, just not enough :). The built-in parallel capabilities of IDL are insufficient for addressing the size of the problem being solved, and processing using any sensible number of IDL licenses would still consume an in-feasible amount of time.

Short of solving this problem, or rewriting applicable portions of code using other languages or IDL functionality, there are few other options. Either of these options would still cost less in man hours, and be more valuable to us, than the cost of the number of IDL licences we would otherwise require.

Luke (a colleague of Tom's)

Subject: Re: accessing IDL internal (aka "built-in") routines
Posted by Haje Korth on Tue, 14 Jan 2014 01:37:44 GMT
View Forum Message <> Reply to Message

Hi Luke,
Sorry for jumping the gun. I frequently have discussions with folks going through extreme efforts to save a buck or two while valuing their time far too little. I certainly underestimated the size of your problem!

I still believe that my assessment regarding the internal routines is right, so you will likely have to

cook your own solution. I have no clue about your project, but have you checked whether GMT (The Generic Mapping Tool) available free at http://gmt.soest.hawaii.edu/ can help you?

Cheers,
Haje

On Monday, January 13, 2014 6:52:24 PM UTC-5, Luke Domanski wrote:
> On Tuesday, January 14, 2014 1:30:10 AM UTC+11, Haje Korth wrote:
>
>> As people say, you get what you pay for. Your time and effort required for solving this problem would likely exceed the cost of an IDL license. Life's too short, use your time wisely!
>
>
>
> Thanks Haje,
>
>
>
> But unfortunately your reasoning is not entirely applicable in this case. We have plenty of IDL licenses, just not enough :). The built-in parallel capabilities of IDL are insufficient for addressing the size of the problem being solved, and processing using any sensible number of IDL licenses would still consume an in-feasible amount of time.
>
>
>
> Short of solving this problem, or rewriting applicable portions of code using other languages or IDL functionality, there are few other options. Either of these options would still cost less in man hours, and be more valuable to us, than the cost of the number of IDL licences we would otherwise require.
>
>
>
> Luke (a colleague of Tom's)

Subject: Re: accessing IDL internal (aka "built-in") routines
Posted by luke.a.domanski on Tue, 14 Jan 2014 02:55:21 GMT
View Forum Message <> Reply to Message

On Tuesday, January 14, 2014 12:37:44 PM UTC+11, Haje Korth wrote:
> Hi Luke,
>
> Sorry for jumping the gun.

That's okay, thanks for the tip on GMT, we will look into it.

The code that calls MAP_PROJ_INIT is part a preprocessing step, and while fairly fast compared

to the main simulation/number crunching, has a high memory or data requirement when considering the scale of problem we will be addressing.

As a work around if we cant find an alternative, we will probably run the faster preprocessing code in parallel over fewer IDL instances/licenses, then run the more time consuming simulation over a far greater number of GDL instances. Each IDL instance will then produce preprocessed results for multiple GDL instances...we will just need to schedule things careful as not to exhaust all or disk space storing TBs of intermediate files.

Subject: Re: accessing IDL internal (aka "built-in") routines
Posted by Yngvar Larsen on Tue, 14 Jan 2014 12:32:12 GMT
View Forum Message <> Reply to Message

On Tuesday, 14 January 2014 00:52:24 UTC+1, Luke Domanski wrote:
> On Tuesday, January 14, 2014 1:30:10 AM UTC+11, Haje Korth wrote:

> But unfortunately your reasoning is not entirely applicable in this case. We have plenty of IDL licenses, just not enough :). The built-in parallel capabilities of IDL are insufficient for addressing the size of the problem being solved, and processing using any sensible number of IDL licenses would still consume an in-feasible amount of time.

I'm curious: since you claim to use several IDL licenses for one "parallel job", you are not using the "idl_idlbridge"? If not: why? Or rather: what limitation of idl_idlbridge is the bottleneck for you?

--
Yngvar

Subject: Re: accessing IDL internal (aka "built-in") routines
Posted by luke.a.domanski on Tue, 14 Jan 2014 21:25:34 GMT
View Forum Message <> Reply to Message

We are launching hundreds to thousands of IDL workers across a large cluster. idl_idlbridge will only serve our purpose on a single node or workstation.

We can use idl_idlbridge in conjunction with our implementation to get on node parallelism, but it would still result in hundreds of IDL licenses to achieve cross node parallelism. (Yes I am aware that there is a commercial IDL product for MPI programming, though I'm not sure this solve the problem either)

There was also a limitation that idl_idlbridge had for something that we wanted to do in the future...but this limitation is not coming to mind right now.

Subject: Re: accessing IDL internal (aka "built-in") routines

Posted by Michael Galloy on Mon, 20 Jan 2014 20:58:56 GMT
View Forum Message <> Reply to Message

On 1/14/14, 2:25 PM, Luke Domanski wrote:
> We are launching hundreds to thousands of IDL workers across a large
> cluster. idl_idlbridge will only serve our purpose on a single node
> or workstation.
>
> We can use idl_idlbridge in conjunction with our implementation to
> get on node parallelism, but it would still result in hundreds of IDL
> licenses to achieve cross node parallelism. (Yes I am aware that
> there is a commercial IDL product for MPI programming, though I'm not
> sure this solve the problem either)
>
> There was also a limitation that idl_idlbridge had for something that
> we wanted to do in the future...but this limitation is not coming to
> mind right now.
>

The only thing I can think to help in this case would be to make sure
you are using runtime licenses (not full developer licenses).

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

Subject: Re: accessing IDL internal (aka "built-in") routines
Posted by Tom Van Niel on Thu, 30 Jan 2014 06:08:50 GMT
View Forum Message <> Reply to Message

Hi Everyone,

Thanks Haje, Luke, and Mike for posting on this topic. The solution we came up with turns out to be quite simple. We installed GDAL and access 'gdalwarp' using 'spawn' from GDL, thus bypassing the need to use IDL internal routines for re-projecting altogether. For example from the GDL prompt to re-project a MODIS image from geographic projection (Lat/Lons) to GDA94 MGA Zone 55 (i.e., ~ UTM 55S):

```
GDL> infn = '/tmp/MOD09GA.hdf'
GDL> outfn = '/tmp/MOD09GA_MGA55.tif'
GDL> EPSG='EPSG:28355'
GDL> spawn, 'gdalwarp -t_srs '+ EPSG + ' '+infn + ' ' + outfn
```

Cheers,
Tom

Subject: Re: accessing IDL internal (aka "built-in") routines
Posted by Haje Korth on Thu, 30 Jan 2014 12:32:04 GMT
View Forum Message <> Reply to Message

Tom,
I am glad you found a solution. Thank you for sharing it with us.

Good luck,
Haje

On Thursday, January 30, 2014 1:08:50 AM UTC-5, mookie...@gmail.com wrote:
> Hi Everyone,
>
>
>
> Thanks Haje, Luke, and Mike for posting on this topic. The solution we came up with turns out to be quite simple. We installed GDAL and access 'gdalwarp' using 'spawn' from GDL, thus bypassing the need to use IDL internal routines for re-projecting altogether. For example from the GDL prompt to re-project a MODIS image from geographic projection (Lat/Lons) to GDA94 MGA Zone 55 (i.e., ~ UTM 55S):
>
>
>
> GDL> infn = '/tmp/MOD09GA.hdf'
>
> GDL> outfn = '/tmp/MOD09GA_MGA55.tif'
>
> GDL> EPSG='EPSG:28355'
>
> GDL> spawn, 'gdalwarp -t_srs '+ EPSG + ' '+infn + ' ' + outfn
>
>
>
> Cheers,
>
> Tom

Subject: Re: accessing IDL internal (aka "built-in") routines
Posted by David Fanning on Thu, 30 Jan 2014 13:09:45 GMT
View Forum Message <> Reply to Message

mookiethemet@gmail.com writes:

> The solution we came up with turns out to be quite simple. We
installed GDAL and access 'gdalwarp' using 'spawn' from GDL, thus
bypassing the need to use IDL internal routines for re-projecting
altogether.

Doh! Now, if we can just get someone to write a little interface and add
GDAL as a DLL we will be set. :-)

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: accessing IDL internal (aka "built-in") routines
Posted by luke.a.domanski on Mon, 03 Feb 2014 23:56:20 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Doh! Now, if we can just get someone to write a little interface and add
>
> GDAL as a DLL we will be set. :-)
>

Generating or writing a wrapper DLL to GDAL itself is probably going to be more straight forward, however, than looking at which MAP_PROJ_INIT functionality is failing in GDL, and then trying to replace all of that functionality transparently with calls to the GDAL library.

So it would be helpful if the development team (Joel Gales, Sylwester Arabas?) could review http://gnudatalanguage.cvs.sourceforge.net/viewvc/gnudatalan guage/gdl/MAP_INSTALL naming functionality that is confirmed to work or not work.

Given the statement in MAP_INSTALL:
81 NOTE: Map projection support in GDL is a work in progress. There are
82 BUGS!! These will be fixed as time permits. Additional projections
83 will also be added in the same manner. Bug reports and projection
84 requests are appreciated.

I will submit a bug report and see how we go.

Subject: Re: accessing IDL internal (aka "built-in") routines
Posted by luke.a.domanski on Tue, 04 Feb 2014 00:15:08 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Doh! Now, if we can just get someone to write a little interface and add
>
> GDAL as a DLL we will be set. :-)
>

Generating or writing a wrapper DLL to GDAL itself is probably going to be more straight forward, however, than looking at which MAP_PROJ_INIT functionality is failing in GDL, and then trying to replace all of that functionality transparently with calls to the GDAL library.

http://gnudatalanguage.cvs.sourceforge.net/viewvc/gnudatalan guage/gdl/MAP_INSTALL does have at least a partial list of functionality that is confirmed to work.

Given the statement in MAP_INSTALL:
```
81      NOTE: Map projection support in GDL is a work in progress.  There are
82      BUGS!!  These will be fixed as time permits.  Additional projections
83      will also be added in the same manner.  Bug reports and projection
84      requests are appreciated.
```

I will submit a bug report and see how we go.

Subject: Re: accessing IDL internal (aka "built-in") routines
Posted by luke.a.domanski on Tue, 04 Feb 2014 02:16:27 GMT
View Forum Message <> Reply to Message

FYI

Submitted the following https://sourceforge.net/p/gnudatalanguage/bugs/588/
