Subject: Re: unloading a dlm...
Posted by Craig Markwardt on Mon, 10 Sep 2001 20:29:08 GMT
View Forum Message <> Reply to Message

Randall Skelton <rhskelto@atm.ox.ac.uk> writes:

> I suspect this is impossible, but does anyone know of a way to force IDL
> unload a DLM (without doing a reset_session).
>
> My problem is that I've quickly bolted a large Fortran model onto IDL and
> nearly every variable is in a common block (i.e. a global C structure).
> It amounts to me consuming an extra 50MB of RAM after this particular DLM
> is loaded :(It would be nice to reclaim this memory when I am done with
> the model...

Try .full_total_absolute_reset_session_for_good_really_now, ahh, umm,
or maybe just .full_reset_session.

But this may not help you if you want to do it programmatically. The
above incantation appears to be available only in recent versions of
IDL, and then only as an executive command at the command line, not in
procedures or functions.

Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: unloading a dlm...
Posted by Richard Younger on Mon, 10 Sep 2001 23:08:27 GMT
View Forum Message <> Reply to Message

Randall Skelton wrote:
>
> I suspect this is impossible, but does anyone know of a way to force IDL
> unload a DLM (without doing a reset_session).
>
> My problem is that I've quickly bolted a large Fortran model onto IDL and
> nearly every variable is in a common block (i.e. a global C structure).
> It amounts to me consuming an extra 50MB of RAM after this particular DLM
> is loaded :(It would be nice to reclaim this memory when I am done with
> the model...

I'm afraid you're just giving ammunition to the common block snobs

(myself among them). I can't help you unload a dlm, but I can think of a few poor alternatives. :-)

If you happen to be on an Intel platform and not developing for anyone else, RAM is dirt cheap compared with six months ago. Well less than US$100 will get you an extra 128 Megs of memory and you can let Moore's law absorb the extra 50 MB hit.

If that's not feasible, you can use the aforementioned .full_reset_session from the main level in a script (@-file).

Compiling as an executable and running with spawn, communicating with pipes or sockets and the like, would probably at least a couple steps backwards.

There might be an obscure option on your F** compiler to change the way it compiles common blocks, but since I haven't really used Fortran much, this is pure speculation on my part.

All of those alternatives are limited and clunky, if they exist at all.
I don't know any better way than to do the obvious (time consuming) thing and gain some quality time with your favorite Fortran compiler and search-and-replace tool.

Good luck,
Rich

--
Richard Younger

Subject: Re: unloading a dlm...
Posted by Randall Skelton on Tue, 11 Sep 2001 10:43:16 GMT
View Forum Message <> Reply to Message

Rats!

\begin{rant}

My *real* problem here is with this Fortran 77 model, which is still being actively developed using F77!? Fortran 77 if fine for algorithms, but for writing large applications, use Fortran 90 at the very least!

\end{rant}

My problem is that I've been told to use a particular atmospheric radiative transfer model written in F77. The program reads a bunch of input cards/files (shutter), and outputs some rather large files (data is

in 2-3 2 dimensional arrays). I need to run this model numerous times in an optimization/fitting process and a significant portion of the processing time is in the reading/writing of the files. My thought was that I would write a DLM that copies (or memory maps) between the Fortran data and IDL. This proved to be much easier than I thought it would be; however, the F77 routines don't pass around variables between subroutines (with the exception of error flags and the odd logical type). Rather, they use 30+ common blocks and, this being Fortran 77, each array in a block is fixed to a maximum size which makes the program rather bloated in memory. What I would like to be able to do is use a DLM to run the model, use mem-copy to copy the relevant portions of the data into IDL arrays (created in C), and then drop the bloated model from memory! I have a few objections to using a 'reset-session' as a programming call, not the least of which is that it will also destroy my newly created data arrays as well.

Thanks for all the suggestions and sorry about the rant... It looks as if I am back to using shell scripts and 'spawn.'

Cheers,
Randall

PS: If anyone has a radiative transfer model capable of producing high-resolution, atmospheric absorption, transmission, and emission spectra for occultation, nadir, and limb geometries from HITRAN data that is written in F90, C, C++, or IDL, please drop me an email...

On 10 Sep 2001, Craig Markwardt wrote:

> Try .full_total_absolute_reset_session_for_good_really_now, ahh, umm,
> or maybe just .full_reset_session.
>
> But this may not help you if you want to do it programmatically. The
> above incantation appears to be available only in recent versions of
> IDL, and then only as an executive command at the command line, not in
> procedures or functions.
>
> Craig
>
> --
> -- ----------
> Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
> Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
> -- ----------
>

Subject: Re: unloading a dlm...
Posted by Nigel Wade on Tue, 11 Sep 2001 12:05:05 GMT
View Forum Message <> Reply to Message

Randall Skelton wrote:
> Rats!
>
> \begin{rant}
>
> My *real* problem here is with this Fortran 77 model, which is still
> being actively developed using F77!? Fortran 77 if fine for algorithms,
> but for writing large applications, use Fortran 90 at the very least!
>
> \end{rant}
>
> My problem is that I've been told to use a particular atmospheric
> radiative transfer model written in F77. The program reads a bunch of
> input cards/files (shutter), and outputs some rather large files (data is
> in 2-3 2 dimensional arrays). I need to run this model numerous times in
> an optimization/fitting process and a significant portion of the
> processing time is in the reading/writing of the files. My thought was
> that I would write a DLM that copies (or memory maps) between the Fortran
> data and IDL. This proved to be much easier than I thought it would be;
> however, the F77 routines don't pass around variables between subroutines
> (with the exception of error flags and the odd logical type). Rather,
> they use 30+ common blocks and, this being Fortran 77, each array in a
> block is fixed to a maximum size which makes the program rather bloated in
> memory. What I would like to be able to do is use a DLM to run the model,
> use mem-copy to copy the relevant portions of the data into IDL arrays
> (created in C), and then drop the bloated model from memory! I have a few
> objections to using a 'reset-session' as a programming call, not the least
> of which is that it will also destroy my newly created data arrays as
> well.
>
> Thanks for all the suggestions and sorry about the rant... It looks as if
> I am back to using shell scripts and 'spawn.'
>
> Cheers,
> Randall
>
> PS: If anyone has a radiative transfer model capable of producing
> high-resolution, atmospheric absorption, transmission, and emission
> spectra for occultation, nadir, and limb geometries from HITRAN data that
> is written in F90, C, C++, or IDL, please drop me an email...
>

If this is running in a virtual memory system does it matter that the
data is still residing in virtual memory?

If you don't touch the pages which contain that data it will get swapped
out as other applications (or the same one) require physical memory.
Provided you have sufficient swap space to accommodate the necessary
swap pages you should not notice any difference to actually reducing the
size of the application. In many environments, even if you can remove
the DLM the amount of memory required by IDL won't reduce as the virtual
memory is not given back to the OS.

As an alternative, can you use the data directly from the common blocks
via IDL_ImportArray rather than creating a new array and copying the
contents? I've never tried using a FORTRAN DLM, but I would have thought
that if you can determine the address of the array that would be
sufficient. You may need to reverse the indexing as I think IDL accesses
arrays in C order rather than the FORTRAN order.
--

Nigel Wade, System Administrator, Space Plasma Physics Group,
University of Leicester, Leicester, LE1 7RH, UK
E-mail : nmw@ion.le.ac.uk
Phone : +44 (0)116 2523568, Fax : +44 (0)116 2523555

Subject: Re: unloading a dlm...
Posted by Paul van Delst on Wed, 12 Sep 2001 14:52:43 GMT
View Forum Message <> Reply to Message

Randall Skelton wrote:
>
> Rats!
>
> \begin{rant}
>
> My *real* problem here is with this Fortran 77 model, which is still
> being actively developed using F77!? Fortran 77 if fine for algorithms,
> but for writing large applications, use Fortran 90 at the very least!
>
> \end{rant}
>
> My problem is that I've been told to use a particular atmospheric
> radiative transfer model written in F77. The program reads a bunch of
> input cards/files (shutter), and outputs some rather large files (data is
> in 2-3 2 dimensional arrays). I need to run this model numerous times in
> an optimization/fitting process and a significant portion of the
> processing time is in the reading/writing of the files. My thought was
> that I would write a DLM that copies (or memory maps) between the Fortran
> data and IDL. This proved to be much easier than I thought it would be;
> however, the F77 routines don't pass around variables between subroutines

> (with the exception of error flags and the odd logical type). Rather,
> they use 30+ common blocks and, this being Fortran 77, each array in a
> block is fixed to a maximum size which makes the program rather bloated in
> memory. What I would like to be able to do is use a DLM to run the model,
> use mem-copy to copy the relevant portions of the data into IDL arrays
> (created in C), and then drop the bloated model from memory! I have a few
> objections to using a 'reset-session' as a programming call, not the least
> of which is that it will also destroy my newly created data arrays as
> well.
>
> Thanks for all the suggestions and sorry about the rant... It looks as if
> I am back to using shell scripts and 'spawn.'

Why a DLM? Why can't you run the LBL code as a regular executable in a shell script and then process the data in IDL via runtime use? I do exactly the same thing you do but avoided the soul-destroying task of trying to "join" the fortran and IDL. Once I processed about 10000 LBLRTM (which produced terabytes of output [not all at once]) runs using my (standalone) IDL code in ~ 6 days. Since your email address is in the UK you're probably using GENLN2, right?

paulv

> PS: If anyone has a radiative transfer model capable of producing
> high-resolution, atmospheric absorption, transmission, and emission
> spectra for occultation, nadir, and limb geometries from HITRAN data that
> is written in F90, C, C++, or IDL, please drop me an email...

The current crop of admittedly arcane f77 LBL RTE code may give you a headache to look at, but it has been verified up the wazoo. The task of rewriting that sort of code in F90 or, heaven forbid, C/C++/IDL/Matlab, would be a monumental task. Don't get me wrong, I would love a well designed f90 version of LBLRTM (for e.g. - I can't follow all the common block stuff neither...) but I wouldn't use it for operational tasks until it had been verified multiple times by various folks.

--
Paul van Delst Religious and cultural
CIMSS @ NOAA/NCEP purity is a fundamentalist
Ph: (301)763-8000 x7274 fantasy
Fax:(301)763-8545 V.S.Naipaul

Subject: Re: unloading a dlm...
Posted by Stein Vidar Hagfors H[1] on Mon, 17 Sep 2001 18:44:41 GMT
View Forum Message <> Reply to Message

Richard Younger <younger@ll.mit.edu> writes:

> Randall Skelton wrote:
>>

>> I suspect this is impossible, but does anyone know of a way to force IDL
>> unload a DLM (without doing a reset_session).
>>
>> My problem is that I've quickly bolted a large Fortran model onto IDL and
>> nearly every variable is in a common block (i.e. a global C structure).
>> It amounts to me consuming an extra 50MB of RAM after this particular DLM
>> is loaded :(It would be nice to reclaim this memory when I am done with
>> the model...
>
> I'm afraid you're just giving ammunition to the common block snobs
> (myself among them). I can't help you unload a dlm, but I can think of
> a few poor alternatives. :-)
>
> If you happen to be on an Intel platform and not developing for anyone
> else, RAM is dirt cheap compared with six months ago. Well less than
> US$100 will get you an extra 128 Megs of memory and you can let Moore's
> law absorb the extra 50 MB hit.
>
> If that's not feasible, you can use the aforementioned
> .full_reset_session from the main level in a script (@-file).

Guess he doesn't really want that: He needs to keep the data in IDL!

> Compiling as an executable and running with spawn, communicating with
> pipes or sockets and the like, would probably at least a couple steps
> backwards.

I think this may be a good candidate for Remote Procedure Calls, i.e. idlrpc,
with two front-ends: One accepting keyboard input, like the idlrpc example
program, and the other (the model) stuffing IDL with the data after doing the
calculations, then simply dying!

You'd at least avoid the writing/reading of data to disk... But if you're
talking about very large volumes, I'm not sure exactly how the RPC protocol
buffers things... (they might go via the disk anyway?). You should at least
use the IDL_RPCImportArray for creating variables to send over large stuff,
but there *will* be some overhead in the sending...

> There might be an obscure option on your F** compiler to change the way
> it compiles common blocks, but since I haven't really used Fortran much,
> this is pure speculation on my part.
>
> All of those alternatives are limited and clunky, if they exist at all.
> I don't know any better way than to do the obvious (time consuming)
> thing and gain some quality time with your favorite Fortran compiler and
> search-and-replace tool.

I would guess there are no ideal solutions to this, short of rewriting the

original code to avoid the static memory allocation (which I understand is not an option).

Come to think of it: If you are passing along the *majority* of the statically allocated data from F77 to IDL, then you might be better off with using Callable IDL - again using the IDL_ImportArray (note the lack of RPC in the routine name).

This way, your F77 program is linked together with IDL, you provide a main program (in C, I would hope) that calls IDL and F77 routines in some alternating fashion.. I guess that technically it's possible to "share memory" in arrays between the two processes (but be *very* careful with writing to those variables in IDL!!).. There's not automatic "reset" of the F77 routine, though, as a .full_reset_session would cause.

> Good luck,

Yes, indeed!

--

---------- ----------
Stein Vidar Hagfors Haugan
ESA SOHO SOC/European Space Agency Science Operations Coordinator for SOHO

NASA Goddard Space Flight Center, Email: shaugan@esa.nascom.nasa.gov
Mail Code 682.3, Bld. 26, Room G-1, Tel.: 1-301-286-9028/240-354-6066
Greenbelt, Maryland 20771, USA. Fax: 1-301-286-0264
---------- ----------