

---

Subject: On-the-fly compilation of routines  
Posted by [gsever](#) on Thu, 06 Nov 2008 04:16:18 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hello,

I would like to know whether it is possible to compile a routine while I am already running my main IDL program?

To illustrate more this, I will explain my intention a bit more. We have a powerful IDL-based data analysis tool in our atmospheric sciences department to analyze airborne acquired data. I am currently working on a spectrum plot routine of the this tool-suite to make small modification and learning purposes. I successfully debug the routines from within the IDL Workbench, but whenever I make a change on the program or wanted to set a new breakpoint I have to restart the program and compile all the routines. My question is again, is there a way to update my current routine in a way to see the changes are going to be reflected without restarting and compiling everything?

Another point is, can we compile only the recently changed routines in the beginning of each compilation process?

I appreciate your comments on these issues.

Thank you again.

---

---

Subject: Re: On-the-fly compilation of routines  
Posted by [R.G. Stockwell](#) on Thu, 06 Nov 2008 17:23:53 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

> "gsever" <gokhansever@gmail.com> wrote in message  
> news:f2ca21f0-e43b-49f7-b13d-98b164701b4f@w24g2000prd.google groups.com...  
> So my understanding is IDL interpreter itself can not sense the change  
> and re-compile the necessary procedure throughout the flow of the  
> program. Every time I make a modification on the source-code I have to  
> re-compile the routine on the console or command shell to reflect this  
> change.

> Once again, in the same IDL session, when I first launch the program  
> 168 routines are compiled, after I make an addition the source code of  
> one file, and re-run the program all 168 routines are re-compiled  
> again. This is controversial to Reimer Bauer's yes answer.

You only need to recompile the program you changed.

Cheers,  
bob

---

---

Subject: Re: On-the-fly compilation of routines  
Posted by [David Fanning](#) on Thu, 06 Nov 2008 17:34:20 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

gsever writes:

> Once again, in the same IDL session, when I first launch the program  
> 168 routines are compiled, after I make an addition the source code of  
> one file, and re-run the program all 168 routines are re-compiled  
> again. This is controversial to Reimer Bauer's yes answer.

Yes, it suggests a poorly written IDL program, I'm afraid. :-(

Cheers,

David

--

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

---

---

Subject: Re: On-the-fly compilation of routines  
Posted by [gsever](#) on Thu, 06 Nov 2008 18:26:12 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

The routines have been structured like Mr. Bauer said in his first reply. I would like to hear your comments about how we can improve the organization of our program.

One of my confusion has not been resolved yet. For example on a modern embedded C compiler, throughout the debugging process if I make a change on the source code and save the file, and try to observe the same function by executing a command or function, the compiler warns me to re-compile the code. While doing this, it only compiles the recently changed file (I mean as far as I understand the compiler or linker attaches a file stamp or flag them, so next time it doesn't re-compile unchanged files. This means a significant compiling speed improvement and this is something I haven't observed in IDL.

On Nov 6, 11:34 am, David Fanning <n...@dfanning.com> wrote:

> gsever writes:  
>> Once again, in the same IDL session, when I first launch the program  
>> 168 routines are compiled, after I make an addition the source code of

>> one file, and re-run the program all 168 routines are re-compiled  
>> again. This is controversial to Reimer Bauer's yes answer.  
>  
> Yes, it suggests a poorly written IDL program, I'm afraid. :-(  
>  
> Cheers,  
>  
> David  
>  
> --  
> David Fanning, Ph.D.  
> Fanning Software Consulting, Inc.  
> Coyote's Guide to IDL Programming:<http://www.dfanning.com/>  
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

---

---

Subject: Re: On-the-fly compilation of routines  
Posted by [David Fanning](#) on Thu, 06 Nov 2008 18:43:06 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

gsever writes:

> The routines have been structured like Mr. Bauer said in his first  
> reply. I would like to hear your comments about how we can improve the  
> organization of our program.  
>  
> One of my confusion has not been resolved yet. For example on a modern  
> embedded C compiler, throughout the debugging process if I make a  
> change on the source code and save the file, and try to observe the  
> same function by executing a command or function, the compiler warns  
> me to re-compile the code. While doing this, it only compiles the  
> recently changed file (I mean as far as I understand the compiler or  
> linker attaches a file stamp or flag them, so next time it doesn't re-  
> compile unchanged files. This means a significant compiling speed  
> improvement and this is something I haven't observed in IDL.

No, because it is typically not necessary to every recompile everything, except when you start IDL. You are working in the Workbench, so when you finish editing a file, and wish to re-compile just that file, reach up and click the "Compile" button. (It looks like a set of gears. Don't ask me.) Just that one file you are working on gets compiled. On my Windows machine CNTL-F8 does the same thing.

Cheers,

David

--

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

---

---

Subject: Re: On-the-fly compilation of routines  
Posted by [sboLin42](#) on Thu, 06 Nov 2008 19:03:50 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

There is also an option to recompile a file every time you save it.  
Windows->Preferences->IDL->Editor->Enable compile on save

-Scott

---

---

Subject: Re: On-the-fly compilation of routines  
Posted by [gsever](#) on Fri, 07 Nov 2008 01:15:16 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Thank you Scott. This the option that I was looking for :)

Now I have been looking ways to our program not to load all 168  
modules in the entree state. I understood that IDL will load necessary  
modules throughout the program flow.

Actually there is an alternative way to launch the program by using  
already compiled binary sav image, which my system shows 1 second  
speed improvement for each different version of calling the program.  
When I run the program without any input arguments it my computer  
takes about 2 second to compile all the routines. I am sure this could  
be improved more.

On Nov 6, 1:03 pm, skydome <[sboli...@gmail.com](mailto:sboli...@gmail.com)> wrote:

> There is also an option to recompile a file every time you save it.  
> Windows->Preferences->IDL->Editor->Enable compile on save  
>  
> -Scott

---

---

Subject: Re: On-the-fly compilation of routines  
Posted by [R.Bauer](#) on Fri, 07 Nov 2008 08:26:01 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

gsever schrieb:

> Thank you Scott. This the option that I was looking for :)  
>

> Now I have been looking ways to our program not to load all 168  
> modules in the entree state. I understood that IDL will load necessary  
> modules throughout the program flow.  
>  
> Actually there is an alternative way to launch the program by using  
> already compiled binary sav image, which my system shows 1 second  
> speed improvement for each different version of calling the program.  
> When I run the program without any input arguments it my computer  
> takes about 2 second to compile all the routines. I am sure this could  
> be improved more.

on which platform do you program?

>  
> On Nov 6, 1:03 pm, skydome <sboli...@gmail.com> wrote:  
>> There is also an option to recompile a file every time you save it.  
>> Windows->Preferences->IDL->Editor->Enable compile on save  
>>  
>> -Scott  
>

---

Subject: Re: On-the-fly compilation of routines  
Posted by [R.Bauer](#) on Fri, 07 Nov 2008 08:44:36 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Andrew Cool schrieb:  
>> mean is a idl source function. You can open it by .edit mean on the idl input line.  
>  
> Well I'll be!!  
>  
> 17 years programming in IDL and I never knew this.  
>  
> What else don't I know? I'll go ask my wife - she'll tell me... ;-)  
>  
> Andrew  
>

Hi Andrew!

On the other hand I know you know things I don't know, that's why I love  
since several years this group.

cheers

Reimar

---

---

Subject: Re: On-the-fly compilation of routines  
Posted by [David Fanning](#) on Fri, 07 Nov 2008 11:37:19 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Reimar Bauer writes:

> Andrew Cool schrieb:  
>>> mean is a idl source function. You can open it by .edit mean on the idl input line.  
>>  
>> Well I'll be!!  
>>  
>> 17 years programming in IDL and I never knew this.  
>>  
>> What else don't I know? I'll go ask my wife - she'll tell me... ;-)  
>>  
>> Andrew  
>>  
>  
>  
> Hi Andrew!  
>  
> On the other hand I know you know things I don't know, that's why I love  
> since several years this group.

This reminds me of a discussion we had several weeks ago that I forgot to follow up on. Someone was trying to compile a large program project with a script that did a .COMPILE on his files. This worked fine in some earlier version of IDL, but had the effect of opening hundreds of edit windows in later versions.

I happened to ask about this when I was around some ITTVIS types who were likely to know the answer. It turns out that .COMPILE is now the equivalent to the command sequence .RUN/.EDIT. And that what the person needed to do was replace all his .COMPILE commands with .RUN.

This sort of turns the .COMPILE command back on its head, since I originally argued for it on the basis that no one outside of RSI could figure out why .RUN only compiled and didn't run anything, and that is why we needed a .COMPILE command.

Anyway, I suppose when your institutional memory leaves the company, it's easy to forget why you did things originally. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

---

---

Subject: Re: On-the-fly compilation of routines

Posted by [R.Bauer](#) on Fri, 07 Nov 2008 13:03:36 GMT

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

---

David Fanning schrieb:

> Reimar Bauer writes:

>

>> Andrew Cool schrieb:

>>>> mean is a idl source function. You can open it by .edit mean on the idl input line.

>>> Well I'll be!!

>>>

>>> 17 years programming in IDL and I never knew this.

>>>

>>> What else don't I know? I'll go ask my wife - she'll tell me... ;-)

>>>

>>> Andrew

>>>

>>

>> Hi Andrew!

>>

>> On the other hand I know you know things I don't know, that's why I love

>> since several years this group.

>

> This reminds me of a discussion we had several weeks ago that

> I forgot to follow up on. Someone was trying to compile a large

> program project with a script that did a .COMPILE on his files.

> This worked fine in some earlier version of IDL, but had the

> effect of opening hundreds of edit windows in later versions.

>

> I happened to ask about this when I was around some ITTVIS types

> who were likely to know the answer. It turns out that .COMPILE

> is now the equivalent to the command sequence .RUN/.EDIT. And

> that what the person needed to do was replace all his .COMPILE

> commands with .RUN.

>

> This sort of turns the .COMPILE command back on its head,

> since I originally argued for it on the basis that no one

> outside of RSI could figure out why .RUN only compiled and

> didn't run anything, and that is why we needed a .COMPILE  
> command.  
>  
> Anyway, I suppose when your institutional memory leaves the  
> company, it's easy to forget why you did things originally. :-)  
>  
> Cheers,  
>  
> David

indeed that is a stupid misbehaviour of idl 7.0 again. luckily it does not crash the idl shell (because of a missing edit window).

In the past the .run dot command was used to compile and execute main programs. There was no difference between the shell and the ide.

Now it behaves this way only if you don't use the ide :(

Reimar

---

---

Subject: Re: On-the-fly compilation of routines  
Posted by [Kenneth P. Bowman](#) on Fri, 07 Nov 2008 14:41:12 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

In article <MPG.237dd652953c0f9d98a510@news.giganews.com>, David Fanning <news@dfanning.com> wrote:

> This reminds me of a discussion we had several weeks ago that  
> I forgot to follow up on. Someone was trying to compile a large  
> program project with a script that did a .COMPILE on his files.  
> This worked fine in some earlier version of IDL, but had the  
> effect of opening hundreds of edit windows in later versions.  
>  
> I happened to ask about this when I was around some ITTVIS types  
> who were likely to know the answer. It turns out that .COMPILE  
> is now the equivalent to the command sequence .RUN/.EDIT. And  
> that what the person needed to do was replace all his .COMPILE  
> commands with .RUN.  
>  
> This sort of turns the .COMPILE command back on its head,  
> since I originally argued for it on the basis that no one  
> outside of RSI could figure out why .RUN only compiled and  
> didn't run anything, and that is why we needed a .COMPILE  
> command.  
>  
> Anyway, I suppose when your institutional memory leaves the



> company, it's easy to forget why you did things originally. :-)  
>  
> Cheers,  
>  
> David

In true unix fashion, I always `.r` my programs to recompile them. Why type all of those extra characters. ;-)  
(`.com` will work, but `.c` is not unique).

Cheers, Ken

---

Subject: Re: On-the-fly compilation of routines  
Posted by [Foldy Lajos](#) on Fri, 07 Nov 2008 19:19:52 GMT  
[View Forum Message](#) <> [Reply to Message](#)

On Fri, 7 Nov 2008, Kenneth P. Bowman wrote:

> In true unix fashion, I always `.r` my programs to recompile them. Why type  
> all of those extra characters. ;-)  
(`.com` will work, but `.c` is not unique).

`.r` is not unique either (`.run/.reset` :-)

regards,  
lajos

---

Subject: Re: On-the-fly compilation of routines  
Posted by [Kenneth P. Bowman](#) on Fri, 07 Nov 2008 20:53:52 GMT  
[View Forum Message](#) <> [Reply to Message](#)

In article <Pine.LNX.4.64.0811072016520.27772@lxserv1.kfki.hu>,  
FOLDY Lajos <foldy@rmki.kfki.hu> wrote:

> On Fri, 7 Nov 2008, Kenneth P. Bowman wrote:  
>  
>> In true unix fashion, I always `.r` my programs to recompile them. Why type  
>> all of those extra characters. ;-)  
(`.com` will work, but `.c` is not unique).  
>  
> `.r` is not unique either (`.run/.reset` :-)  
>  
> regards,  
> lajos

But `.r` expands to `.run`, while `.c` expands to `.continue`. I'm sure there's a logical reason. :-)

Ken

---

---

Subject: Re: On-the-fly compilation of routines  
Posted by [R.Bauer](#) on Mon, 10 Nov 2008 09:08:06 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Kenneth P. Bowman schrieb:

> In article <Pine.LNX.4.64.0811072016520.27772@lxserv1.kfki.hu>,  
> FOLDY Lajos <foldy@rmki.kfki.hu> wrote:  
>  
>> On Fri, 7 Nov 2008, Kenneth P. Bowman wrote:  
>>  
>>> In true unix fashion, I always .r my programs to recompile them. Why type  
>>> all of those extra characters. ;-) (.com will work, but .c is not unique).  
>> .r is not unique either (.run/.reset :-)  
>>  
>> regards,  
>> lajos  
>  
> But .r expands to .run, while .c expands to .continue. I'm sure there's a  
> logical reason. :-)  
>  
> Ken

no it is a bug. ;)

At least if you see it in comparison to the keyword logic.

cheers  
Reimar

---