Subject: Re: Self-compiling procedures
Posted by Vince Hradil on Sat, 16 Feb 2008 21:51:12 GMT
View Forum Message <> Reply to Message

On Feb 16, 12:47 pm, maye <kmichael....@googlemail.com> wrote:
> Hi!
> In terms of self-compiling I'm a beginner, so please excuse my
> ignorance if this is written somewhere in the manual, I couldn't find
> it easily at least:
> So far I put every procedure I consider to be useful enough for more
> than one task in its extra file, so that if it's called, IDL compiles
> it automatically, having the procedure and file name identical.
> But after a while, the number of files I have keeps growing.
> I wonder if there's another elegant way to keep things auto-compilable
> without me having to compile some kind of library before I start
> working on a task using my "library" of useful procedures.
> Am I missing something simple?
> Thanks for any hints and a nice weekend!
> Best regards,
> Michael

I think you've got it. Just as long as they are in the path, you can
just:
1- write a new procedure(s) that needs some of the library functions.
2- compile the new procedure(s)
3- resolve_all

I think that's the "most elegant" - the compilation is really fast,
anyways.

Subject: Re: Self-compiling procedures
Posted by Spon on Sat, 16 Feb 2008 22:05:47 GMT
View Forum Message <> Reply to Message

On Feb 16, 6:47 pm, maye <kmichael....@googlemail.com> wrote:
> Hi!
> In terms of self-compiling I'm a beginner, so please excuse my
> ignorance if this is written somewhere in the manual, I couldn't find
> it easily at least:
> So far I put every procedure I consider to be useful enough for more
> than one task in its extra file, so that if it's called, IDL compiles
> it automatically, having the procedure and file name identical.
> But after a while, the number of files I have keeps growing.
> I wonder if there's another elegant way to keep things auto-compilable
> without me having to compile some kind of library before I start
> working on a task using my "library" of useful procedures.
> Am I missing something simple?

> Thanks for any hints and a nice weekend!
> Best regards,
> Michael

You should add your library directory to your !PATH - File-
> Preferences->Path.
Then you can call every procedure/function in the library without ever
having to compile it first. In fact, any directory that has any .pro
file in it is probably better off in !PATH. Top level directories,
anyway - thankfully IDL searches recursively.

Take care,
Chris

Subject: Re: Self-compiling procedures
Posted by maye on Sat, 16 Feb 2008 22:18:21 GMT
View Forum Message <> Reply to Message

On Feb 16, 11:05 pm, Spon <christoph.b...@gmail.com> wrote:
> On Feb 16, 6:47 pm, maye <kmichael....@googlemail.com> wrote:
>
>> Hi!
>> In terms of self-compiling I'm a beginner, so please excuse my
>> ignorance if this is written somewhere in the manual, I couldn't find
>> it easily at least:
>> So far I put every procedure I consider to be useful enough for more
>> than one task in its extra file, so that if it's called, IDL compiles
>> it automatically, having the procedure and file name identical.
>> But after a while, the number of files I have keeps growing.
>> I wonder if there's another elegant way to keep things auto-compilable
>> without me having to compile some kind of library before I start
>> working on a task using my "library" of useful procedures.
>> Am I missing something simple?
>> Thanks for any hints and a nice weekend!
>> Best regards,
>> Michael
>
> You should add your library directory to your !PATH - File->Preferences->Path.
>
> Then you can call every procedure/function in the library without ever
> having to compile it first. In fact, any directory that has any .pro
> file in it is probably better off in !PATH. Top level directories,
> anyway - thankfully IDL searches recursively.
>
> Take care,
> Chris

Thanks so much for your fast answers, and I'm very sorry, I was too
imprecise with my question, silly me.
I am actually looking for a way to group similar procedures into one
file to reduce the amount of library files, it's just harder to
maintain.
Is there a way to keep things still self-compilable when I put several
procedures into one file?

Cheers,
Michael

Subject: Re: Self-compiling procedures
Posted by Spon on Sat, 16 Feb 2008 22:30:09 GMT
View Forum Message <> Reply to Message

On Feb 16, 10:18 pm, maye <kmichael....@googlemail.com> wrote:
> On Feb 16, 11:05 pm, Spon <christoph.b...@gmail.com> wrote:
>
>
>
>> On Feb 16, 6:47 pm, maye <kmichael....@googlemail.com> wrote:
>
>>> Hi!
>>> In terms of self-compiling I'm a beginner, so please excuse my
>>> ignorance if this is written somewhere in the manual, I couldn't find
>>> it easily at least:
>>> So far I put every procedure I consider to be useful enough for more
>>> than one task in its extra file, so that if it's called, IDL compiles
>>> it automatically, having the procedure and file name identical.
>>> But after a while, the number of files I have keeps growing.
>>> I wonder if there's another elegant way to keep things auto-compilable
>>> without me having to compile some kind of library before I start
>>> working on a task using my "library" of useful procedures.
>>> Am I missing something simple?
>>> Thanks for any hints and a nice weekend!
>>> Best regards,
>>> Michael
>
>> You should add your library directory to your !PATH - File->Preferences->Path.
>
>> Then you can call every procedure/function in the library without ever
>> having to compile it first. In fact, any directory that has any .pro
>> file in it is probably better off in !PATH. Top level directories,
>> anyway - thankfully IDL searches recursively.
>
>> Take care,
>> Chris

>
> Thanks so much for your fast answers, and I'm very sorry, I was too
> imprecise with my question, silly me.
> I am actually looking for a way to group similar procedures into one
> file to reduce the amount of library files, it's just harder to
> maintain.
> Is there a way to keep things still self-compilable when I put several
> procedures into one file?
>
> Cheers,
> Michael

Short answer - no.

Here, have a read of this, particularly the second half:

http://www.dfanning.com/tips/namefiles.html

Good luck,
Chris

Subject: Re: Self-compiling procedures
Posted by Vince Hradil on Sat, 16 Feb 2008 22:45:24 GMT

View Forum Message <> Reply to Message

On Feb 16, 4:18 pm, maye <kmichael....@googlemail.com> wrote:
> On Feb 16, 11:05 pm, Spon <christoph.b...@gmail.com> wrote:
>
>> On Feb 16, 6:47 pm, maye <kmichael....@googlemail.com> wrote:
>
>>> Hi!
>>> In terms of self-compiling I'm a beginner, so please excuse my
>>> ignorance if this is written somewhere in the manual, I couldn't find
>>> it easily at least:
>>> So far I put every procedure I consider to be useful enough for more
>>> than one task in its extra file, so that if it's called, IDL compiles
>>> it automatically, having the procedure and file name identical.
>>> But after a while, the number of files I have keeps growing.
>>> I wonder if there's another elegant way to keep things auto-compilable
>>> without me having to compile some kind of library before I start
>>> working on a task using my "library" of useful procedures.
>>> Am I missing something simple?
>>> Thanks for any hints and a nice weekend!
>>> Best regards,
>>> Michael
>
>> You should add your library directory to your !PATH - File->Preferences->Path.

>
>> Then you can call every procedure/function in the library without ever
>> having to compile it first. In fact, any directory that has any .pro
>> file in it is probably better off in !PATH. Top level directories,
>> anyway - thankfully IDL searches recursively.
>
>> Take care,
>> Chris
>
> Thanks so much for your fast answers, and I'm very sorry, I was too
> imprecise with my question, silly me.
> I am actually looking for a way to group similar procedures into one
> file to reduce the amount of library files, it's just harder to
> maintain.
> Is there a way to keep things still self-compilable when I put several
> procedures into one file?
>
> Cheers,
> Michael

I started out putting several procedures in on file and found it
harder to maintain that way. With each file a separate procedure I
know exactly where to look. That said, of course it makes sense to
group procedures and functions in one file, if they are only used by
the "main" function (the one with the same name as the file). If they
are going to be used by themselves, then I break them out as separate
files.

Subject: Re: Self-compiling procedures
Posted by maye on Sat, 16 Feb 2008 23:50:32 GMT
View Forum Message <> Reply to Message

On Feb 16, 11:30 pm, Spon <christoph.b...@gmail.com> wrote:
> On Feb 16, 10:18 pm, maye <kmichael....@googlemail.com> wrote:
>
>
>
>> On Feb 16, 11:05 pm, Spon <christoph.b...@gmail.com> wrote:
>
>>> On Feb 16, 6:47 pm, maye <kmichael....@googlemail.com> wrote:
>
>>>> Hi!
>>>> In terms of self-compiling I'm a beginner, so please excuse my
>>>> ignorance if this is written somewhere in the manual, I couldn't find
>>>> it easily at least:
>>>> So far I put every procedure I consider to be useful enough for more
>>>> than one task in its extra file, so that if it's called, IDL compiles

>>>> it automatically, having the procedure and file name identical.
>>>> But after a while, the number of files I have keeps growing.
>>>> I wonder if there's another elegant way to keep things auto-compilable
>>>> without me having to compile some kind of library before I start
>>>> working on a task using my "library" of useful procedures.
>>>> Am I missing something simple?
>>>> Thanks for any hints and a nice weekend!
>>>> Best regards,
>>>> Michael
>
>>> You should add your library directory to your !PATH - File->Preferences->Path.
>
>>> Then you can call every procedure/function in the library without ever
>>> having to compile it first. In fact, any directory that has any .pro
>>> file in it is probably better off in !PATH. Top level directories,
>>> anyway - thankfully IDL searches recursively.
>
>>> Take care,
>>> Chris
>
>> Thanks so much for your fast answers, and I'm very sorry, I was too
>> imprecise with my question, silly me.
>> I am actually looking for a way to group similar procedures into one
>> file to reduce the amount of library files, it's just harder to
>> maintain.
>> Is there a way to keep things still self-compilable when I put several
>> procedures into one file?
>
>> Cheers,
>> Michael
>
> Short answer - no.
>
> Here, have a read of this, particularly the second half:
>
> http://www.dfanning.com/tips/namefiles.html
>
> Good luck,
> Chris

Thx will do.
And I got an idea: I could at least sort things by subfolders. As IDL
parses recursively - as it was mentioned before - I can put related
procedures together.
Thanks for your hints.
Cheers,
Michael

Subject: Re: Self-compiling procedures
Posted by Maarten[1] on Mon, 18 Feb 2008 08:55:19 GMT
View Forum Message <> Reply to Message

On Feb 16, 11:50 pm, Michael Aye <kmichael....@googlemail.com> wrote:
> On Feb 16, 11:30 pm, Spon <christoph.b...@gmail.com> wrote:
>> On Feb 16, 10:18 pm, maye <kmichael....@googlemail.com> wrote:
>>> Thanks so much for your fast answers, and I'm very sorry, I was too
>>> imprecise with my question, silly me.
>>> I am actually looking for a way to group similar procedures into one
>>> file to reduce the amount of library files, it's just harder to
>>> maintain.
>>> Is there a way to keep things still self-compilable when I put several
>>> procedures into one file?
>> Short answer - no.
>> Here, have a read of this, particularly the second half:
>> http://www.dfanning.com/tips/namefiles.html

> And I got an idea: I could at least sort things by subfolders. As IDL
> parses recursively - as it was mentioned before - I can put related
> procedures together.
> Thanks for your hints.

If the functions are truly related, you may want to investigate the
use of object programming. The IDL manual on the subject is quite
horrible - it assumes you want to use their object graphics and write
user interfaces in IDL (yuck). But as far as grouping related
function, it is a powerful concept. Be prepared to dive into pointers
though.

Maarten

Subject: Re: Self-compiling procedures
Posted by maye on Tue, 19 Feb 2008 09:29:27 GMT
View Forum Message <> Reply to Message

On Feb 18, 9:55 am, Maarten <maarten.sn...@knmi.nl> wrote:
> On Feb 16, 11:50 pm, Michael Aye <kmichael....@googlemail.com> wrote:
>
>
>
>> On Feb 16, 11:30 pm, Spon <christoph.b...@gmail.com> wrote:
>>> On Feb 16, 10:18 pm, maye <kmichael....@googlemail.com> wrote:
>>>> Thanks so much for your fast answers, and I'm very sorry, I was too
>>>> imprecise with my question, silly me.
>>>> I am actually looking for a way to group similar procedures into one
>>>> file to reduce the amount of library files, it's just harder to

>>>> maintain.
>>>> Is there a way to keep things still self-compilable when I put several
>>>> procedures into one file?
>>> Short answer - no.
>>> Here, have a read of this, particularly the second half:
>>> http://www.dfanning.com/tips/namefiles.html
>> And I got an idea: I could at least sort things by subfolders. As IDL
>> parses recursively - as it was mentioned before - I can put related
>> procedures together.
>> Thanks for your hints.
>
> If the functions are truly related, you may want to investigate the
> use of object programming. The IDL manual on the subject is quite
> horrible - it assumes you want to use their object graphics and write
> user interfaces in IDL (yuck). But as far as grouping related
> function, it is a powerful concept. Be prepared to dive into pointers
> though.
>
> Maarten

Yes, saw the object stuff already, but wondered, as you said, if
that's only there for object graphics or if there's more use to it. So
it's a good point.
Which reminds me of another question since IDL 7.0: What's ITT's idea
now for graphical UI's?
And are there actually ways to do the UI e.g. with QT Designer or with
WXWindowx/WxPython and put it on top of IDL routines? I guess, then
one has to start with all that importing stuff and doing it in C/C++
using some libraries for import?

Regards,
Michael