
Subject: More problems with Build Project, IDLDE vs IDLRT question
Posted by j.coenia@gmail.com on Tue, 22 Dec 2009 22:39:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

IDL 7.0: Based on advice from an earlier thread, I now import every single possibly needed library into my project before I build, to try to resolve dependencies. Build time went from 7s to over 1 min. Fine, but now when I try to run from the built .sav file, I get the following message:

The following error was encountered: Keyword MYKEYWORD not allowed in call to: MYFUNCTION.

That's bull. The first keyword for MYFUNCTION is in fact MYKEYWORD. The source for MYFUNCTION is right there in the project now, and there is no other myfunction.pro anywhere on the entire computer (I searched), so I can't see how it's getting confused.

Again, the same exact application works fine if I just run it in IDLDE, and has worked fine for years! It seems almost impossible to build a binary from a huge old application like this, written over several years by several different programmers coming in and out of a research lab. What's really annoying is the program works perfectly fine in the development environment, so there's nothing wrong with the code.

Is there any way to just create a shortcut to a script to tell IDLDE to run the program *as if* it's a standalone program, without the development environment ever popping up in a window at all? That would be fine with me, since our computers all have IDL licenses. (I'm exaggerating, but it feels like I could re-write this whole thing in Python in the time it's taking me to build a .sav file from working code!)

Thanks for any suggestions.

Subject: Re: More problems with Build Project, IDLDE vs IDLRT question
Posted by j.coenia@gmail.com on Wed, 23 Dec 2009 18:54:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

> IDL> Resolve_Routine, 'myfunction', /EITHER

In a new IDL session, or after a full reset, the above command works just fine! And then Routine_Filepath('myfunction', /EITHER) also will work just fine. However, if I then just build the project that includes myfunction, and issue the exact same Routine_Filepath command, it fails

Subject: Re: More problems with Build Project, IDLDE vs IDLRT question
Posted by [David Fanning](#) on Wed, 23 Dec 2009 19:01:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

j.coenia@gmail.com writes:

> In a new IDL session, or after a full reset, the above command works
> just fine! And then Routine_Filepath('myfunction', /EITHER) also will
> work just fine. However, if I then just build the project that
> includes myfunction, and issue the exact same Routine_Filepath
> command, it fails

Is "myfunction" perhaps being evoked in the program
from within a Call_Function call?

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: More problems with Build Project, IDLDE vs IDLRT question
Posted by j.coenia@gmail.com on Wed, 23 Dec 2009 20:57:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

> Is "myfunction" perhaps being evoked in the program
> from within a Call_Function call?

Definitely not. "myfunction" is actually a user-defined compound
widget replacement for widget_button, or a "my_widget_button". The
only time it is ever called is when the gui is being set up.

I think the button replacement was necessary for two small reasons:
(1) widget_buttons can't generate tooltips when they are not
sensitive, which is perhaps the time when you are most curious about
what the button might do, and (2) widget button exclusive bases can't
exclude each others' buttons, because pushbutton events are not
generated by buttons in exclusive bases. At least, that's what it
looks like to me -- all your excluding buttons have to be contiguous
in the same base: you can't push a button in one exclusive base that
immediately pops up a button in another, like proper radio buttons

would.
