

---

Subject: Need advice on building a project  
Posted by [MKatz843](#) on Tue, 17 Dec 2002 21:43:56 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I'm learning how to build a project in IDL for use with a runtime licence on another machine. I've been able to build relatively small projects without any trouble. My problem is in building a really big application.

In my IDL working directories, I have hundreds and hundreds of .pro files, broken down into directories: functions, procedures, objects, etc. Only about 100-150 of those libraries will be required for the project at hand.

Is there an easy way to build just the relevant files into my project without having to hand pick them all individually or in small groups? Something like "Resolve All" but for building the project? What I'm looking for is a way where IDL can say, "you'll need this one, and that one, and this one. . . And then put them together for me."

I actually tried building the project with EVERY SINGLE file. That was a mistake.

I appreciate any advice from the gurus.

M. Katz

---

---

Subject: Re: Need advice on building a project  
Posted by [David Fanning](#) on Wed, 18 Dec 2002 06:42:36 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

M. Katz ([MKatz843@onebox.com](mailto:MKatz843@onebox.com)) writes:

- > I'm learning how to build a project in IDL for use with a runtime
- > licence on another machine. I've been able to build relatively small
- > projects without any trouble. My problem is in building a really big
- > application.
- >
- > In my IDL working directories, I have hundreds and hundreds of .pro
- > files, broken down into directories: functions, procedures, objects,
- > etc. Only about 100-150 of those libraries will be required for the
- > project at hand.
- >
- > Is there an easy way to build just the relevant files into my project
- > without having to hand pick them all individually or in small groups?
- > Something like "Resolve All" but for building the project? What I'm
- > looking for is a way where IDL can say, "you'll need this one, and

> that one, and this one. . . And then put them together for me."  
>  
> I actually tried building the project with EVERY SINGLE file. That was  
> a mistake.  
>  
> I appreciate any advice from the gurus.

I hate to even respond to this question, because it's just going to spur those IDLWAVE radicals on to new heights, but the IDL project interface could use some work. (I was going to say it "sucks", but I hear about it from my wife whenever I use language like that.)

Let's just say it's "peculiar".

I use it a lot, and once it is set up correctly it is amazingly helpful. But getting it set up can be a pain. I think the interface is clunky, and I guess it's non-intuitive too. (I suppose that is what you call it when you always find yourself doing something you didn't want to do.) It takes me two or three goes, usually, to get a file where I want it. (Maybe I'm just getting old.)

But the most annoying thing, to me, is changing the build order when you have a LOT of files. New files go at the bottom of the list, if you want them at the top of the list you have to drag them there, but the window doesn't scroll, so you have to drag them to the top of the \*visible\* window, where you drop them, and the window \*then\* scrolls all the way to the top, so you scroll down to find the file you just left closer to the top (somewhere!), and you continue on in this fashion until you decide a beer sounds like a real good idea or you manage to manhandle your file into the right place.

I have a rule that after I get 30 files in a project I write NO MORE FUNCTIONS, PERIOD. You can stick procedures anywhere in the build order, of course, without problems.

To say something nice about projects, changing the logical layout of folders, etc. is pretty easy. You can just drag and drop files from one virtual folder to another. You can't have sub-folders, of course, so if you think hierarchically, you are pretty much out of luck, as far as I know. But I'm a pretty shallow thinking kind of guy anyway, so this isn't much of a problem for me.

And one other thing. (I wasn't going to mention this because it is so weird.) I have a project that has somehow stored a record of phantom files! I close all my project editor windows,

save my project, exit. Now I come into IDL, open my project and build it. All my project files are compiled. Now, if I run the main program of my project by typing "main" from the IDL command line, all it perfect. But if I select Run Project from the menu or from the Run Project button, two of the file that just got compiled get re-compiled and they pop up in the editor windows!

I'm getting nowhere with Technical Support about this. But I thought it was worth mentioning. I'm guessing if the guys took a long look at this code, they might find one or two bugs yet.

Anyway, I go to all the trouble to copy files over into my project directories one at a time, as I add them to the project. Yes it's a pain. But that's what I do.

Now, let's hear what the IDLWAVE guys do. :-)

Cheers,

David

--

David W. Fanning, Ph.D.

Fanning Software Consulting, Inc.

Phone: 970-221-0438, E-mail: [david@dfanning.com](mailto:david@dfanning.com)

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

Toll-Free IDL Book Orders: 1-888-461-0155

---

Subject: Re: Need advice on building a project  
Posted by [Craig Markwardt](#) on Wed, 18 Dec 2002 07:08:44 GMT  
[View Forum Message](#) <> [Reply to Message](#)

David Fanning <[david@dfanning.com](mailto:david@dfanning.com)> writes:

> the IDL project interface could use some work. (I was going

IDLWAVE user ponders: "project interface?"

Craig

---

Subject: Re: Need advice on building a project  
Posted by [Pavel A. Romashkin](#) on Wed, 18 Dec 2002 16:48:27 GMT  
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

>

> I have a rule that after I get 30 files in a project I write

> NO MORE FUNCTIONS, PERIOD. You can stick procedures anywhere  
> in the build order, of course, without problems.

I am not quite clear on this, why? I never have any particular order and as long as they all get compiled why would I care?

Does it have something to do with setting or not the STRICT compiler option so that any parenthesis are considered a function call as opposed to brackets for arrays?

For what its worth, I just copy-paste a couple of lines like

Compile\_opt, IDL2, hidden

Error\_handler

as the first lines in all of my programs.

Cheers,

Pavel

---

Subject: Re: Need advice on building a project

Posted by [MKatz843](#) on Thu, 19 Dec 2002 17:50:05 GMT

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

Pavel,

Could you please say a few words about what these lines accomplish.

Compile\_opt, IDL2, hidden

Error\_handler

Thanks,

M. Katz

---

Subject: Re: Need advice on building a project

Posted by [MKatz843](#) on Thu, 19 Dec 2002 18:25:00 GMT

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

David, there may be hope.

> Anyway, I go to all the trouble to copy files over into my  
> project directories one at a time, as I add them to the  
> project. Yes it's a pain. But that's what I do.

Atle at RSI Tech support has given me some good advice.

I was able to build an IDL runtime application without adding each and every file to the project. Amen. I included only the main procedure, plus all of the object definition files, because IDL can't "see" that it needs those at compile-time. By selecting output as .sav, IDL's project built procedure automatically appended a RESOLVE\_ALL command

to the end of the compile. It appears that that takes care of all of the obvious links to other functions and procedures automatically.

So it seems there IS an easier way than adding every file.

Allow me to second Pavel's question about the importance of build-order that you mentioned.

Thanks again,

M. Katz

---

Subject: Re: Need advice on building a project  
Posted by [David Fanning](#) on Thu, 19 Dec 2002 19:11:27 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

M. Katz (MKatz843@onebox.com) writes:

> Atle at RSI Tech support has given me some good advice.  
> I was able to build an IDL runtime application without adding each and  
> every file to the project. Amen. I included only the main procedure,  
> plus all of the object definition files, because IDL can't "see" that  
> it needs those at compile-time. By selecting output as .sav, IDL's  
> project built procedure automatically appended a RESOLVE\_ALL command  
> to the end of the compile. It appears that that takes care of all of  
> the obvious links to other functions and procedures automatically.

Right. This certainly works with run-time files you wish to ship to colleagues who only work with that run-time application.

The problem I have is that the projects I work with are collaborative efforts between people at several locations (even across the world). To be sure what I see on my machine is what they see on their machine, we have to be careful about source code, too. Thus, it is important to me that each project file be in a particular location in the "project" directory structure.

This is quite easy to get out of whack, especially if you have a number of "releases", and results in all kinds of head scratching and muttering until I realize what is going on. :-(

> Allow me to second Pavel's question about the importance of  
> build-order that you mentioned.

Build order is almost never a problem in files I write.

After a long career writing IDL programs I sorta, mostly, know what I'm doing. The problem, really, is with all the legacy code we sometimes have to deal with. A lot of that code is not... well, very well written. (Probably the reason they have hired us in the first place, to be honest.)

Sometimes we just re-write the darn things the way they should be written, but sometimes we have to live with what we have. The result is a lot of trouble sometimes. :-)

Cheers,

David

--

David W. Fanning, Ph.D.

Fanning Software Consulting, Inc.

Phone: 970-221-0438, E-mail: [david@dfanning.com](mailto:david@dfanning.com)

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

Toll-Free IDL Book Orders: 1-888-461-0155

---

Subject: Re: Need advice on building a project

Posted by [Pavel A. Romashkin](#) on Fri, 20 Dec 2002 16:38:28 GMT

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

---

"M. Katz" wrote:

```
> Could you please say a few words about what these lines accomplish.  
>  
> Compile_opt, IDL2, hidden  
> Error_handler
```

Compile\_opt, IDL2, hidden takes care of the ambiguity that used to matter when both () and [] were used to subscript arrays. I think I recall complains about compiling problems and use of the Forward\_Function to avoid them. Setting STRICTRARR or IDL2 as COMPILE\_OPT lets IDL assume that any () indicate a function call, even if it is not compiled yet.

As far as for Error\_Handler, I used it to abbreviate for these lines:

```
Catch, Err_code ; Provide error handling.  
if (Err_code ne 0) then begin  
  print, 'Program error caught:'  
  help, /last_message  
  catch, /cancel  
  return  
endif
```

that I place in (almost) every program (or at least into event handlers) to provide feedback on any errors that might occur. I think there will be no feedback in runtime but at least the program won't crash. Often it is sufficient to handle only event handler errors because any nested errors will be traced back as well.

Cheers,  
Pavel

---

---

Subject: Re: Need advice on building a project  
Posted by [Robert Barnett](#) on Fri, 02 Jul 2004 03:21:31 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

For my first major project using IDL, I used version 5.4 and the idlde. I'm about to move onto a second project and I think that even idlde for version 6.0 is not terribly useful.

I want to know if anyone has tried this before:

Today I quickly wrote up a (perl) script that allows one to specify the build order in a heirarchical fashion.

Each directory contains a BuildOrder file which indicates what programs need to be built but also can contain dependancies on other BuildOrder files.

The aim of this is to package my programs so that any given group of programs can be compiled and tested as indepenantly as possible. Also a package (including it's build order) can be imported and exported without too much trouble.

I've only written a very simple script and there are no tools or documentation. I have only attached the code so that you might get some idea of what I'm getting at. Please do not expect it to actually work for you.

However, I'm sure that others must have explored similar methods for doing this. Does anyone know of someone who might have done this already?

--

nrb  
Robbie Barnett  
@  
Research Assistant  
wsahs

Nuclear Medicine & Ultrasound  
nsw  
Westmead Hospital  
gov  
Sydney Australia  
au  
+61 2 9845 7223

```
#!/usr/bin/perl
```

```
$debug = 0;  
$idlexec = 'idl';  
$filename = 'BuildOrder';
```

```
$r_commands = [];  
$pwd = $ENV{"PWD"};  
resolveOrder($r_commands,$filename,$pwd/);  
$debug && print join("\n",@$r_commands);
```

```
$startup = $ENV{"IDL_STARTUP"};  
if (defined($startup)) {  
    open(STARTUP,">$startup");  
    print STARTUP join("\n",@$r_commands) . "\n";  
    close(STARTUP);  
}
```

```
$savefile = $ARGV[0];  
if (defined($savefile)) {  
    open(IDL,'|' . $idlexec);  
    print IDL join("\n",@$r_commands) . "\n";  
    print IDL "SAVE, FILENAME='$savefile'\n";  
    print IDL "EXIT\n";  
    close(IDL);  
}
```

```
sub resolveOrder {  
    my ($r_commands,$filename,$directory) = @_;  
    if (-f $filename) {  
        $debug && print "Examining $filename\n";  
        open(ORDER,$filename);  
        my @lines = (<ORDER>);  
        close(ORDER);  
        $debug && print "$filename has " . scalar(@lines) . " lines\n";  
        foreach $line (@lines) {  
            $line =~ s/\s+$/;/;  
            $line = $line;  
            if (-f $directory . $line) {  
                push(@$r_commands,".COMPILE $line");  
            }  
        }  
    }  
}
```





files in a directory and tell the user to put the damn directory on their path. It's never failed yet. :-)

Cheers,

David

--

David Fanning, Ph.D.  
Fanning Software Consulting, Inc.  
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

---

---

Subject: Re: Need advice on building a project  
Posted by [Paul Van Delst\[1\]](#) on Fri, 02 Jul 2004 14:29:52 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Robert Barnett wrote:

>  
> For my first major project using IDL, I used version 5.4 and the idlde.  
> I'm about to move onto a second project and I think that even idlde for  
> version 6.0 is not terribly useful.  
>  
> I want to know if anyone has tried this before:  
>  
> Today I quickly wrote up a (perl) script that allows one to specify the  
> build order in a heirarchical fashion.

Forgive my denseness, but what do you mean by "build order"? Are you saying that your setup has problems finding other dependent .pro files that aren't in your current directory? If so, then all you should need to do is stick your entire IDL code heirarchy into you IDL\_PATH env variable.

Apologies if you already know this, but the way I do it is to put \*all\* my IDL code in an /idl subdirectory off my home directory. Then I do a  
export IDL\_PATH=\${IDL\_PATH}:+\${HOME}/idl  
in my .bashrc and every single file in every subdirectory is "visible" to IDL when you compile your .pro file of the moment.

> Each directory contains a BuildOrder file which indicates what programs  
> need to be built but also can contain dependancies on other BuildOrder  
> files.  
>  
> The aim of this is to package my programs so that any given group of  
> programs can be compiled and tested as indepenantly as possible. Also a  
> package (including it's build order) can be imported and exported  
> without too much trouble.

Now, the export issue is a different one (I reckon). I solve this using CVS. When I want to export some code (e.g. to distribute to users) I tag (cvs tag) all of the required files with a string (e.g., "Ootsma\_Plotter\_2-13" to indicate this is release 2.13 of my IDL application "oostma\_plotter.pro") I associate with that application. Then you simply do something like

```
cvs export -r Ootsma_Plotter_2-13 -d./ ${CVS_REPOSITORY}
```

to extract all the required .pro files into the current directory. And you can do this anywhere since cvs works across networks.

The only thing that can be a pain in the rear is tagging all the required files in the first place. But if you do it as you write your app (just like the documentation :o) then it's no biggie. Still, a perl script to do this task would be handy.

I'm not sure if this addresses exactly what you asked about. At any rate, I recommend CVS to cut down on versionitis -- although I'm always amazed at the resistance people have to using it... It's quite peculiar. Especially when I see people with multiple copies of the same file in different directories, each one possibly subtly modified in a way that may or not be noticed depending on which one is first in the IDL path search list.. Anyway...

paulv

---

Subject: Re: Need advice on building a project  
Posted by [David Fanning](#) on Sat, 03 Jul 2004 05:26:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Robbie Barnet writes:

> I didn't realise that IDL searched subdirectories, I was \*told\* that it  
> didn't do that. I'll have to look into what versions of IDL support  
> this.

My first experience with IDL was with IDL 1.0, and  
I'm pretty sure \*that\* one searched sub-directories. :-)

> Even though, I still have problems with stupid code which doesn't  
> keep the filename and procedure name the same.

People who write code like this should be taken  
out back and shot. (Apologies to many of my friends.)

> I'm also concerned with  
> accidentally having two files with the same name in a different directory.  
> Imagine two indepenant applications in your idl directory which use  
> different versions of the same file. The only way around this is to move  
> applications in and out of your idl directory.

Well, another way around this, frequently employed, as I understand it, is to rename one of the files. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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

---

Subject: Re: Need advice on building a project

Posted by [Robert Barnett](#) on Sat, 03 Jul 2004 17:15:06 GMT

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

---

Thanks, I think that you've nailed my two problems pretty square on the head.

I didn't realise that IDL searched subdirectories, I was \*told\* that it didn't do that. I'll have to look into what versions of IDL support this. Even though, I still have problems with stupid code which doesn't keep the filename and procedure name the same. I'm also concerned with accidentally having two files with the same name in a different directory. Imagine two indepenant applications in your idl directory which use different versions of the same file. The only way around this is to move applications in and out of your idl directory.

Thanks for recommending CVS, I certainly value the idea of explicit dependancies. I have to spend a lot of time reading other peoples code. I frequently come across a call to a procedure or function which I have no idea where it is and what other code that procedure or function requires. I think that I'm used to other languages such as perl and java where code is distributed in packages.

I think that the main problem with CVS is that I often have to jump through hoops to get new applications installed. Thankfully, I have a linux box hidden away in a room somewhere which I can use to serve files.

Just a quick question

How do/would you deploy a runtime IDL application if you were using CVS? You may have noticed that my script allows me to specify a .SAV file. When I do this my script compiles the files specified BuildOrder including their dependancies and subsementally saves them to the .sav file. Is there an easy way to do this with CVS?

Paul Van Delst wrote:

> Robert Barnett wrote:

>

>>

>> For my first major project using IDL, I used version 5.4 and the  
>> idlde. I'm about to move onto a second project and I think that even  
>> idlde for version 6.0 is not terribly useful.

>>

>> I want to know if anyone has tried this before:

>>

>> Today I quickly wrote up a (perl) script that allows one to specify  
>> the build order in a heirarchical fashion.

>

>

> Forgive my denseness, but what do you mean by "build order"? Are you  
> saying that your setup has problems finding other dependent .pro files  
> that aren't in your current directory? If so, then all you should need  
> to do is stick your entire IDL code heirarchy into you IDL\_PATH env  
> variable.

>

> Apologies if you already know this, but the way I do it is to put \*all\*  
> my IDL code in an /idl subdirectory off my home directory. Then I do a  
> export IDL\_PATH=\${IDL\_PATH}:+\${HOME}/idl  
> in my .bashrc and every single file in every subdirectory is "visible"  
> to IDL when you compile your .pro file of the moment.

>

>> Each directory contains a BuildOrder file which indicates what  
>> programs need to be built but also can contain dependancies on other  
>> BuildOrder files.

>>

>> The aim of this is to package my programs so that any given group of  
>> programs can be compiled and tested as indepenantly as possible. Also  
>> a package (including it's build order) can be imported and exported  
>> without too much trouble.

>

>

> Now, the export issue is a different one (I reckon). I solve this using  
> CVS. When I want to export some code (e.g. to distribute to users) I tag  
> (cvs tag) all of the required files with a string (e.g.,  
> "Ootsma\_Plotter\_2-13" to indicate this is release 2.13 of my IDL  
> application "oostma\_plotter.pro") I associate with that application.  
> Then you simply do something like  
> cvs export -r Ootsma\_Plotter\_2-13 -d./ \${CVS\_REPOSITORY}  
> to extract all the required .pro files into the current directory. And  
> you can do this anywhere since cvs works across networks.

>

> The only thing that can be a pain in the rear is tagging all the

> required files in the first place. But if you do it as you write your  
> app (just like the documentation :o) then it's no biggie. Still, a perl  
> script to do this task would be handy.  
>  
> I'm not sure if this addresses exactly what you asked about. At any  
> rate, I recommend CVS to cut down on versionitis -- although I'm always  
> amazed at the resistance people have to using it... It's quite peculiar.  
> Especially when I see people with multiple copies of the same file in  
> different directories, each one possibly subtly modified in a way that  
> may or not be noticed depending on which one is first in the IDL path  
> search list.. Anyway...  
>  
> paulv  
>

---

---

Subject: Re: Need advice on building a project  
Posted by [JD Smith](#) on Sun, 04 Jul 2004 08:59:42 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Fri, 02 Jul 2004 22:26:00 -0700, David Fanning wrote:

> Robbie Barnet writes:  
>  
>> I didn't realise that IDL searched subdirectories, I was \*told\* that it  
>> didn't do that. I'll have to look into what versions of IDL support  
>> this.  
>  
> My first experience with IDL was with IDL 1.0, and I'm pretty sure  
> \*that\* one searched sub-directories. :-)  
>  
>> Even though, I still have problems with stupid code which doesn't keep  
>> the filename and procedure name the same.  
>  
> People who write code like this should be taken out back and shot.  
> (Apologies to many of my friends.)  
>  
>> I'm also concerned with  
>> accidentally having two files with the same name in a different  
>> directory. Imagine two independent applications in your idl directory  
>> which use different versions of the same file. The only way around this  
>> is to move applications in and out of your idl directory.  
>  
> Well, another way around this, frequently employed, as I understand it,  
> is to rename one of the files. :-)

I've lately resorted to a little program that runs at startup and  
looks for a file "IDL\_IGNORE" in any of the directories on the search

path, and then prunes those from the path. This is very useful when you want to quickly eliminate a file from consideration.

```
-----  
pro prune_idl_ignore,RECURSIVE=rcrsv,VERBOSE=vb  
  dirs=strsplit(!PATH,path_sep(/SEARCH_PATH),/EXTRACT)  
  ps=path_sep()  
  sweep=file_test(dirs+ps+'IDL_IGNORE',/REGULAR)  
  good=where(sweep eq 0,ngood)  
  if ngood eq 0 then return  
  if keyword_set(rcrsv) then begin  
    for i=0,ngood-1 do begin  
      root=dirs[good[i]]  
      repeat begin  
        pos=strpos(root,ps,/REVERSE_SEARCH)  
        if pos ne -1 then root=strmid(root,0,pos)  
        endrep until pos eq -1 || file_test(root+ps+'IDL_IGNORE',/REGULAR)  
        if pos ne -1 then sweep[good[i]]=1  
      endfor  
    endfor  
  endfor  
  good=where(sweep eq 0,ngood)  
  if keyword_set(vb) then begin  
    bad=where(sweep,nbad)  
    print,'Pruning ',strtrim(nbad,2),' !PATH dirs:'  
    if nbad gt 0 then print,' '+transpose(dirs[bad])  
  endfor  
  if ngood eq 0 then return  
  !PATH=strjoin(dirs[good],path_sep(/SEARCH_PATH))  
end
```

---