Subject: Re: IDL 7 on LINUX Question
Posted by Doug Edmundson on Sat, 01 Dec 2007 01:55:11 GMT

View Forum Message <> Reply to Message

Vince Hradil wrote:
> On Nov 30, 3:55 pm, Doug Edmundson <do...@ittvis.com> wrote:
>> David Fanning wrote:
>>> Doug Edmundson writes:
>>>> Vince is correct. The IDL Workbench is parsing all the PRO code on your
>>>> path, in the current directory, in your open projects, and editors. If
>>>> you're similar to our ENVI developers, it can take a while to do all
>>>> that! I've already taken quite a bit of ribbing on this, so the plan is
>>>> to speed it up.
>>> Given this, and given that I have several big libraries that
>>> I always use: JHUAPL, ASTRO, MARKWART, COYOTE, etc., is it
>>> better to make each of these "projects" and always open
>>> them. Or, should I just add these directories to my path
>>> and not bother to make them projects?
>>> Cheers,
>>> David
>> David,
>>
>> I think a lot of people will be asking this.
>>
>> I'd probably have a project for each of them and leave them open all the
>> time. One benefit is that you'll have all the nice searching
>> capabilities of Eclipse, such as the "search" dialog. But that's not
>> all! You also get "Open Resource" (see the "Navigate" menu). That's
>> one of the IDE team's favorites. Doing cmd+shift+R pops up the "Open
>> Resource" dialog. Type in a partial name of a file and voila... a list
>> of matching files appears like magic. This is a fast way to hunt down
>> files.
>>
>> The "Open Declaration" feature doesn't require projects and provides a
>> fast way to hunt down routines.
>>
>> If you like to manually manage your path, those new projects should
>> probably have the "Update IDL Path preference..." option turned off when
>> you create them. That property can always be changed later (right click
>> on a project, bring up its properties and select "IDL Project
>> Properties"). For those totally new to the IDL Workbench, you can take
>> a look at your path preference in "Preferences > IDL > Paths".
>>
>> So, projects can be pretty benign. You don't have to use them to build
>> SAV files or anything like that, but they do facilitate a lot of the
>> workbench functionality.
>>
>> Thanks for the good question.

>>
>> Doug
>
> A couple of questions/comments
> 1- how does IDL-Wave for Emacs do this. Doesn't it build a "library"
> file that keeps track of things like keywords and completions, etc.
> Then that file can be quickly searched. Of course, if you change your
> library, you have to re-scan the files, but I find this not too
> inconvenient.

I'll defer to JD on that one. I will say that it set the bar pretty high.

> 2- I'm having trouble figuring out exactly what the "Update IDL Path
> preference..." option is for. I have a library directory with several
> subdirectories containing various external libraries (like David's and
> Craig's, etc.). These are always in my IDL_PATH. Then I used to
> direct IDL to "Change directory on open" so that I could resolve_all
> and find the other files that might be needed, since they are in the
> same directory. These are specific routines - if they get generalized
> and "library worthy", I would move them to my subdirectory in the
> IDL_PATH directory. How can I best use the combination of IDL_PATH
> and Projects to emulate this?

First, a little overview...

One of the design goals of the IDL Workbench was to make it easy for new
and casual users. The idea is that they should not have to worry about
the path unless they really need to. To achieve that goal we created
the "Update IDL Path Preference" property for projects. When set, the
IDL Workbench will add the project's directory to the path preference
(with a '+' out front). The workbench will re-expand the path as
directories are added to the project. It also takes care of refreshing
the path cache as PRO files are added. When the property is not set,
the IDL Workbench does nothing to the path preference (and hence path).
 Each project has its own personal setting.

This should be nice for new users, casual users and even experienced
users who don't like to manually manage their path. Those that are
power users will probably want to turn that setting off on most
projects, if not all. The "New Project" dialog remembers your last
choice, so if you know that you never want path management, just unset
it the first time, don't worry about it again and the IDL Workbench will
never muck with your path.

If you have the "Update IDL Path Preference" property turned on, one can
monitor what the workbench is doing by looking at "Preferences > IDL >
Paths". One can also change the order, as in the old IDE. Note that
when projects are being managed, they are added to the end of your path.

So, closing and then re-opening a project will move it to the end. The conclusion to all this is that if order matters then the property should probably be turned off.

To answer your question more directly, if you turn off path management for all projects then things are no different than before. If order is not important then automatic path management might be a nice convenience. (You'll just have to trust the IDL Workbench!) ;-) I'd recommend using projects regardless of path management decisions because they provide other useful functionality. The "Open Resource" feature really does rock!

Here's a scenario where managed projects might really help. Say you have a new version of a library and want to test it out. So you wind up have two projects, one pointing at the old and one for the new. If both projects were managed, closing one would automatically remove it from the path. Opening the other would add it. So, toggling back and forth between projects would be easier than manually managing the path.

I hope all this has made sense. We had many a debate among ourselves as to the best way to achieve that ease-of-use design goal. Ultimately we decided that the IDL Workbench should pretty much do what a user would likely do... add new directories to the path preference. Not good for everyone, hopefully good for most, and it can be disabled.

Thanks for the good question.

Doug