Subject: Re: IDL 8.0 compile_opt changes
Posted by penteado on Thu, 24 Dec 2009 06:13:25 GMT
View Forum Message <> Reply to Message

On Dec 24, 2:48 am, David Fanning <n...@dfanning.com> wrote:
> prof...@gmail.com writes:
>> As an old fogey who still has working code from 1982 (version 2, when
>> we used VMS and our disk quota was 5 kiloblocks), I much prefer
>> the .pro/.prx solution. If I have to dig into that old code to change
>> parentheses to brackets, specify that 16 bit associated variables are
>> 16 bits, remove all VMS-specific keywords, and everything else that is
>> now obsolete, I'll probably feel compelled to rewrite 28 years worth
>> of code to make it look and run better, and my productivity will drop
>> to or below zero... You think I can improve code without breaking it?
>
> I think most people could probably put a "compile_opt idl1" into their
> code without breaking it.

If he does that, he will break the code for use with older versions, since the option idl1 does not exist in them. As probably the upgrade will not happen at the same time on all 4 machines he uses, he would need to keep two sets of files, one to use with IDL 8 with the idl1 option, and the other, unchanged, to use with IDL <8. In my experience, it is common for people to use the same code on many different computers (often even all from the same filesystem), and they do not have control over when upgrades are installed in some, or all of those computers.

This is one example of why I was saying that a new extension should be used. It is the best way to keep old code working while making the needed changes. Even those who want to retrofit their code will not want to keep two copies of each file (with and without the idl1 option), and may not have the time to immediately stop everything to find and replace all the () in the old files.

Even in Fortran, when people wanted to get rid of the old horrible fixed source format the choice was made to use a new extension. So the new compilers would know to treat the .f files with the old horrible standard, and the .f90 files with the new one. Fortran is a language with a strong lack of good choices in its design, but in that particular decision was a good one. Sure, it made a lot of people not even noticing that there was something new, even to this day (if you are not familiar with Fortran, a lot of people still use the 1977 standard as if it is the only one). But a lot of people would have kept using old compilers if the new ones did not work with unaltered old code.

As I said before, IDL currently has issues that already make its

upgrade too difficult. It should not have a new obstacle. Keep in mind that the most frequent posters in this newsgroup are likely to be better informed of IDL's changes and better ways to write code, and more willing to make changes, than most IDL users. If idl2 becomes default, that majority of users may simply not understand why their code stopped working, and so keep using the older versions. And most of the well informed users know how inconvenient it is to keep limiting what they use in their new code because some if its users are still with an old version of IDL. And even if those people do not even know it, we can send them our files in the new standard and their relatively new compilers will understand it.

It reminds me of the (true) case of the man who, after living well for most of life while blind, recovered his vision in his fifties through surgery. His family then thought that he should immediately live as a normal person, without ever using his cane or any of the other aids he used when blind. Since he did not know how to make sense of the images, he could no longer function, got stressed, depressed, sick and died shortly afterwards. This story is told in Oliver Sacks' book "An Anthropologist on Mars", and was somewhat altered into the story of the movie "At First Sight", with Val Kilmer and Mira Sorvino. My point is that even if change is for the better, forced instant change with no adaptation can have severe bad side effects.

In short, breaking compatibility with old code would be much worse than a new extension, both for those who agree that the new way if doing things is necessary, and for those who do not even now there is a new way. I say this from experience of how people in the academic sector write their code, and of trying to get them to stop suffering from the old ways.