Subject: Re: IDL 8.0 compile_opt changes
Posted by penteado on Thu, 24 Dec 2009 16:49:26 GMT
View Forum Message <> Reply to Message

On Dec 24, 11:07 am, wlandsman <wlands...@gmail.com> wrote:
> Maybe it would be better then to have a flag (or separate icon) when
> starting up IDL, e.g.
>
> idl -classic
>
> that would recognize the parentheses indexing (and thus *not*
> recognize the new method syntax). This would ensure that users
> could run their old code, but also encourage them to take the 20
> minutes to automatically convert their code to use square bracket
> indices. (Note that IDL currently knows when () is used as an index
> and when it is used for a function in a pre-8.0 procedure -- since it
> knows the names of all variables -- so the conversion can certainly be
> automated.) --Wayne

I would not say that the conversion can be automated. The current criteria to decide whether something is a function or variable depend not only on the defined variables, but also on the known functions, at the time the reference is encountered. The trouble happens if there are both a function and a variable with the same name. If the reference is found before the function is compiled, it is taken to be a variable. If the function happens to have been compiled, the reference is taken to be a function. So the conversion of a file does not depend only on its contents, but also on the environment when the file is read. And as many other topics here indicate, routine name resolution is fraught with difficulties.

What could be automated is to add the compile opt to, say, all routines found in a directory. The trouble is that those files would no longer work on older versions because of the idl1, so it is not a good solution either. Having a switch or a preference making the interpreter work the old way is better than making idl2 the default all the time. But it has the problem that one would not be able to run both old and new code during the same session, so it would still pose an obstacle to change, though a smaller one. Which is why I still see a new extension as a better choice.