Subject: Re: Automatic Compiliation of IDL Programs, Was: Lost Functions
Posted by hegde on Wed, 05 Nov 1997 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning (davidf@dfanning.com) wrote:

> do not understand how programs get compiled. Many of these
> people try to run IDL in the compile-link-run fashion of
> the good ol' Fortran days. :-)

Compile-link-run fashion is still the standard (best?) way.

> If you use the IDL executive command .Compile to compile
> a file, all the program modules in the file get compiled.

Major deficiency with IDL compiled state is that it doesn't even check whether calling function(/procedure)'s argument list matches with the compiled one. So if one makes a spelling error, the program will crash at runtime. If the software is of considerable size, chances are that one might miss that particular state during testing; whereas a simple check would have avoided that.

> But one of the wonderful features of IDL is the ability
> to have your programs compiled *automatically* when they
> are needed. This makes it possible to write IDL code that
> is not cluttered up with a bunch of INCLUDE statements, etc.
> You simply put all of your program files in some directory
> that is on the IDL path and you don't have to worry whether
> the files are pre-compiled or not when you write the next
> IDL program that uses one of these commands.

Isn't it similar to 'library path' in C/C++?. INCLUDE is not intended for including a huge source code anyway.

Say if one's display library contains 20 different files and if they were modified after compilation, to recompile one has to type 20 .Runs or quit the session and start all over again !

OR sometimes it is hard to predict which function will be called first. ex: string libraries for parsing/manipulating strings. The only solution then will be to keep 10-20 line codes in separate files generating an army of files.

> Automatic compilation doesn't solve *all* the problems
> you might run into. (Hence, the Forward_Function command.)
> But in my experience it solves a lot of them. And it sure
> beats having to compile all of your program modules before
> you can do anything in IDL. :-)

This feature might be good for running few things from IDL prompt. But as a programmer, I would like to better manage source code instead of pondering each time I call a function whether IDL would have compiled it before.

The arguments for 'separate files' appear to me as though telling that one could do something similar but not quite similar.

-M. Hegde