Subject: Re: Automatic Compiliation of IDL Programs, Was: Lost Functions
Posted by Stein Vidar Hagfors H on Thu, 06 Nov 1997 08:00:00 GMT
View Forum Message <> Reply to Message

M. Hegde wrote:
[..]
> 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.

On the other hand, with IDL being an interpreting language, it allows
dynamically created (or modified) programs (which are compiled only
when *called* during runtime, not when *referenced* during compilation
of another procedure).

In fact I do have some programs which writes procedures "on the fly",
compiling them as they are needed - obviously this could cause
problems if syntax checking was performed at compile-time (not
very hard to fix, though).

But it would certainly be nice to have a few extra tools to manage
large IDL software collections - e.g., checking for number of
parameters,
legality of keywords, etc..

> 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 !

Actually, you can type e.g., ".run prog1 prog2 prog3 prog4 prog5"....

Usually, when working on a large set of routines in one "session",
I put such statements into a file, e.g., "c.pro", and then I simply
type "@c" when I need to recompile...

(And when working with the idl-shell mode, my fingers seem to
have a will of their own, typing ^C-^D-^C (saving and recompiling)
faster than I can think)

> 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.

I do agree with you, I just haven't experienced IDL's standard
"compile-when-needed" approach as a problem. I agree that you may

get a silly number of tiny procedures, but it's always possible
to put them into one file (let's say, "stringlib.pro") and then
have a procedure at the end of that called STRINGLIB. If the user
wishes to use the string library - have the statement "stringlib"
in the startup file.

And let a stand-alone program (IDL procedure or otherwise) do the
syntax/parameter checking.

Regards,

Stein Vidar
