Subject: Re: are there any s/w eng tools for IDL
Posted by Tim Patterson on Mon, 24 Feb 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Judith Bachman wrote:
>
> I'm fairly new to IDL programming. I'm finding that IDL
> does it's job well, but it doesn't help me or the rest of my team
> do ours very well!
>
> As experienced C/C++ programmers we really miss a
> compiler that can warn that we've messed up a calling sequence or
> done something that's probably dumb as far as data typing goes.
> We are finding that we're spending a lot of time doing "desk
> checking" to catch things that a complier catches. Does anyone
> have a "lint" like program for IDL or are we going to have to
> learn to be VERY careful when we code? Does anyone have
> recommended coding standards that might help. We're using a
> "Hungarian notation" derivative to help keep data typing under
> control - that's been a help.
>
> Thanks in advance for any suggestions that folks might have.
> Judith Bachman
> Judith.Bachman@gsfc.nasa.gov

There's a useful IDL mode for emacs which is worth getting.

There's also the IDLTOOL (type 'idltool' at the unix shell prompt)
which is a very, very basic "debug" tool which is ok for
simple routines, but isn't anything to get too excited
over. Basically, it just has a GUI to the same functions
such as HELP and BREAKPOINT that you can use via the IDL shell.

Until RSI introduce type-checking and other such features,
all you can do is try and be as thorough as possible about
employing coding standards. It is very easy to run up a
few modules in IDL which is great for prototyping, but
can be a real nightmare for projects that are under more
rigorous control. Perhaps the OO stuff in IDL 5.0 will
allow better software engineering prectices to be introduced.
Until then I find using structures to gather up like-objects
can be very useful, as it minimises the chances of mistyping
a variable name and introducing a new variable at run-time!

Tim