Subject: Re: Self-compiling procedures
Posted by Maarten[1] on Tue, 19 Feb 2008 09:56:51 GMT
View Forum Message <> Reply to Message

On Feb 19, 9:29 am, Michael Aye <kmichael....@googlemail.com> wrote:
> On Feb 18, 9:55 am, Maarten <maarten.sn...@knmi.nl> wrote:
>> If the functions are truly related, you may want to investigate the
>> use of object programming. The IDL manual on the subject is quite
>> horrible - it assumes you want to use their object graphics and write
>> user interfaces in IDL (yuck). But as far as grouping related
>> function, it is a powerful concept. Be prepared to dive into pointers
>> though.
>
> Yes, saw the object stuff already, but wondered, as you said, if
> that's only there for object graphics or if there's more use to it. So
> it's a good point.

There are a few analysis objects, and yes: objects are useful outside of the graphics and UI context. Since the objects are bolted onto the structures, you also inherit (no pun intended) the limitation of named structures: arrays have to be pointers, or you won't be able to resize them.

> Which reminds me of another question since IDL 7.0: What's ITT's idea
> now for graphical UI's?
> And are there actually ways to do the UI e.g. with QT Designer or with
> WXWindowx/WxPython and put it on top of IDL routines?

IDL has its own UI toolkit (probably based on something more or less native). I /never/ use IDL for user interfaces, the 'I' in IDL is pointless to me at this moment, and I don't think that will change. I would say: just don't bother with IDL user interfaces, especially if you already know Python and WxPython. Those are much cleaner, more modern, and far less frustrating. And some algorithms are just plain easier to write and read back - if you need to obtain a fair speed. 'for' loops don't kill Python like they do IDL.

Maarten