
Subject: Re: IDL 6.0

Posted by [JD Smith](#) on Fri, 28 Mar 2003 16:59:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Fri, 28 Mar 2003 08:51:37 -0700, Michael A. Miller wrote:

>>>> >> "Michael" == Michael A Miller <mmiller3@iupui.edu> writes:

>

>> That and access to command line arguments like every other

>> language on the planet would go a long way to streamlining my IDL

>> use.

>

> To clarify my point a bit - what I'm getting at is that much of what we

> do here with IDL is interactive, so the "I" in IDL is really great. Even

> more of what we do is along the lines of "take what we've developed

> interactively and repeat it over and over and over ad infinitum." The

> cumbersome methods that are needed to pass parameters to IDL codes from

> outside IDL has resulted in some really ugly kludges around here.

>

> One source of that seems to be that many of us new-old-schoolers

> (troglodytes that is :-)) tend to want to string a collection of programs

> together in sequence with a script. To do that with IDL, every tool we

> develop with IDL needs to have an additional layer of code wrapped

> around it to handle the fact that IDL has no access to command line

> arguments. That's something that we (well, ok, I) got used to back in

> the 80's and it is hard to give up. Even though IDL has lots of

> wonderful features, that "I" for interactive makes some harder to get at

> than I'd like.

>

> Ok, I'll get off that soap box now ;-)

The problem is the IDL startup time is long enough that interactive use is stressed much more than batch use. I.e. the standard response would be that you should string together your bits of code in an *IDL* script/batch-file/\$MAIN\$-level-routine.

If you have other types of code you need to interact with in IDL, call it from there (as opposed to visa versa). This can be awkward if IDL is really just a small component of a large process, but it's really the most convenient way.

JD
