Subject: Re: Pre-processor-like Capability?
Posted by thompson on Tue, 11 Jun 1996 07:00:00 GMT
View Forum Message <> Reply to Message

"Bruce E. Thomason" <brucet@hsonline.net> writes:

> Does anyone know of a IDL/PV-Wave capability that is analogous to
> C's preprocessor?

> I am particularly interested in efficiently establishing symbolic
> references (in the coding process) to constant values for use across
> many routines (as #define does w/ the C pre-processor), but to have
> these values efficiently accessed at run-time.

> The two alternative's I've come up with are:
> 1. Declaring variables via an "include" (@xxx) file. This has the down
> side of potentially including declarations to variables that are not
> used in a routine, as well as accessing variables in what (I expect) is
> a less efficient manner than accessing a literal value.

> 2. Using system variables, which I also expect is less efficient that
> accessing a literal value.

> I would prefer a language intrinsic approach, but might have to settle
> for writing/using a (language independent) preprocessor. Does anyone
> have a recommendation in this regard?

I suggest you follow your second option above. Use system parameters, and set the actual values in the IDL startup file.

Considering the way that IDL works, I don't think that there is any significant performance difference in using a literal value versus storing the value in a system variable. In some sense, IDL uses a "preprocessor". The first time that you call a procedure, it is read and converted into a format that IDL uses to actually run it. I'm not sure in detail what happens to literal values embedded in the code, but expect that they're converted to temporary parameters which are accessed in a very similar way as other parameters.

Bill Thompson