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

Subject: Re: Pre-processor-like Capability?
Posted by dave on Thu, 13 Jun 1996 07:00:00 GMT
View Forum Message <> Reply to Message

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

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

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

I create small functions at the beginning of the source that return the desired values:

```
FUNCTION magic_number
  return 'boing'
END

PRO whatever, x, y, z
  . . .
  IF a NE magic_number THEN BEGIN
    print, "Bad Magic"
    return, 0
  ENDIF
  . . .
END
```

This is portable, can be included in one source file, and used in another, and is almost as simple as a #define. It's a good idea to use a FORWARD_FUNCTION declaration to declare all such functions at the beginning of a procedure, in case the definition is in another file.

Performace won't suffer unless your code is highly looped, in which case you chould probably be linking in a C for FORTRAN module anyway.

Good luck,

David.
--
David Fenyes University of Texas Medical School
dave@msrad72.med.uth.tmc.edu Dept. of Radiology

Subject: Re: Pre-processor-like Capability?
Posted by dave on Sat, 15 Jun 1996 07:00:00 GMT
View Forum Message <> Reply to Message

From: dave@msrad71.med.uth.tmc.edu (David Fenyes)
Subject: Re: Pre-processor-like Capability?
Newsgroups: comp.lang.idl-pvwave
Date: 13 Jun 1996 13:36:50 -0500
Organization: UTHSC-Houston, Dept of Radiology
Path: news.uth.tmc.edu!usenet
Lines: 44
Message-ID: <x6afy78sal.fsf@msrad71.med.uth.tmc.edu>
References: <31BD4280.351B@hsonline.net>
NNTP-Posting-Host: msrad71.med.uth.tmc.edu
Mime-Version: 1.0
Content-Type: text/plain;charset=US-ASCII
Content-Transfer-Encoding: 7bit
In-reply-to: "Bruce E. Thomason"'s message of Tue, 11 Jun 1996 09:55:12 +0000
X-Newsreader: Gnus v5.1

Some typos:

```
>  FUNCTION magic_number
>     return 'boing'
>  END

>  PRO whatever, x, y, z

     forward_function magic_number

>     . . .
>     IF a NE magic_number THEN BEGIN
            ^ magic_number()

>        print, "Bad Magic"
>        return, 0
>     ENDIF
>     . . .
>  END
```

David.--

David Fenyes University of Texas Medical School
dave@msrad72.med.uth.tmc.edu Dept. of Radiology