Subject: Re: Best way to share 'enum' between various functions
Posted by David Fanning on Sat, 24 Sep 2011 13:05:47 GMT
View Forum Message <> Reply to Message

Robin Wilson writes:

> I have a number of constant definitions that I'm using to do what an
> enum would do in other languages:
>
> MARITIME = 1
> CLOUD = 2
> URBAN = 3
> ...
>
> I want these definitions to be available to a lot of different functions
> and procedures (all in different files) in my code - they are used a lot
> in IF statements and various assignments.
>
> Obviously I don't want to replicate the definitions in loads of
> different files, as if I ever change them (or add extra ones) then it
> will get *very* confusing.
>
> What's the best way to do this?

Many of the people I know who do this have a list
of system variables that they create in a start-up
file. Like programs themselves, you have to keep
one eye open for naming conflicts, but it seems
to work reasonably well.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thue. ("Perhaps thos speakest truth.")

Subject: Re: Best way to share 'enum' between various functions
Posted by Brian Wolven on Mon, 26 Sep 2011 16:40:49 GMT
View Forum Message <> Reply to Message

Another method is to define a structure, each tag of which corresponds to one of the constant

definitions you may need. You can write a function that simply defines the structure and returns it to the calling procedure. In use it would look something like this:

```
my_con = my_constants()

some_variable = my_con.maritime
other variable = my_con.urban
```

You can also easily "browse" your definitions:

help,/struct,my_constants()
