Subject: Re: Are user-defined private methods possible?
Posted by Oliver on Fri, 10 Jul 2009 08:33:48 GMT
View Forum Message <> Reply to Message

cool - I'm always amazed about the creativity of people when it comes
to circumvent all those IDL annoyances.

For my lib I used the much simpler approach of defining a 'magic'
number, something along the lines of

```
pro myClass::privateMethod, arg1, arg2, magic
  if( magic ne 148 ) then message, 'private calls only...'
  ; ...
end
```

This is obviously no protection at all against someone who on purpose
wants
to call this method, but then its his/her own fault and I would
consider this
bad style and a programming error.

Speaking about mimicking OO concepts in IDL: what also annoys me is
the absence of static methods and class variables.
For the former I use the convention

```
pro myClass_staticMethodName
```

i.e. by mixing "classic" IDL and object orientation. Ugly but it
works.
For the latter I use commons:

```
common myClassStaticVars initializedFlag, var1, var2, var3
```

The problem with this approach is that variables cannot be initialized
here.
So I define a method

```
pro myClass_initializeStaticVars
```

which I either call at program start or (if there're not too many)
in the constructor and all 'static methods' of the class.
InitializedFlag
is set to '1' during the first call so repeated calls have no effect.

Cheers
Oliver