Subject: Re: LIST extensions
Posted by Paul Van Delst[1] on Wed, 15 Dec 2010 14:54:51 GMT
View Forum Message <> Reply to Message

Matt Haffner wrote:
> Also, note that LIST is a subclass of IDL_Container, which has
> a .Count() method, so .Length() may not be needed.

Excellent tip. I did not know that LIST was a subclass of IDL_Container. I will be sure to check that sort of thing with
other IDL objects in the future, i.e.

```
IDL> l=list(1,2,34,4)
IDL> help, l, /object
** Object class LIST, 2 direct superclasses, 4 known methods
   Superclasses:
      IDL_CONTAINER  <Direct>
      IDL_OBJECT  <Direct>
   Known Function Methods:
      IDL_CONTAINER::COUNT
      LIST::INIT
      LIST::_OVERLOADHELP
   Known Procedure Methods:
      LIST::ADD
IDL> help, l.count()
<Expression>    LONG      =           4
```

And the Get method worked even though the list contains no objects:

```
IDL> help, l.get(position=2)
<PtrHeapVar4>   LONG      =          34
```

As does the Move:

```
IDL> print, l
           1
           2
          34
           4
IDL> l.move,2,0
IDL> print, l
          34
           1
           2
           4
```

Tres cool!

I don't know why I should be surprised. But it would be nice if the superclasses were listed in the documentation.

Again, thanks for the info.

cheers,

paulv

> There is also
> a .Move method to rearrange items in a container. Unfortunately
> the .IsContained method doesn't seem to work for me on a LIST though
> (and is only for objects, in any case):
>
> IDL> z=g[50]
> IDL> print,g.IsContained(z)
> 0
> IDL> print,obj_valid(z, /get_heap)
> 1884104
> IDL> print,obj_valid(g[50], /get_heap)
> 1884104
> IDL> print, z eq g[50]
> 1
>
> mh
