
Subject: Re: DLM heap variable access

Posted by [Jason Ferrara](#) on Mon, 06 Jul 2009 12:16:12 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Jun 27, 3:34 pm, pp <pp.pente...@gmail.com> wrote:

> I got tired of waiting for the ITTVIS folks to implement some more
> data structures in IDL. Coding a bunch of them (lists, maps, stacks)
> in IDL would be a fair amount of rewriting the wheel, and also
> inefficient, because of the way IDL's pointers and scalars work. So I
> decided that the nicest solution would be to have IDL objects as
> wrappers to C++ containers. It is simple enough to do it writing a DLM
> in the way Ronn Kling's book suggests, with the IDL object containing
> a (real) pointer to the C++ object, and wrapper methods to call the C++
> methods.

With Slither you can use Python's container classes from IDL. I realize it's a bit extreme to use a method that requires a Python install plus a Slither license just to get some containers, but it does work rather well.

```
IDL> pb=pyimport("__builtin__")
% Loaded DLM: SLITHER.
IDL> l=pb->list()
IDL> l->append, 6
IDL> l->append, 3
IDL> l->append, 23
IDL> print, pb->len(l)
      3
IDL> print, l->__getitem__(1)
      3
IDL> print, l->__getitem__(0)
      6
IDL> print, l->__getslice__(1,3)
      3      23
IDL> print, pytoidl(l)
      6      3      23
IDL> print, l->pop()
      23
IDL> print, l->pop()
      3
IDL>
IDL>
IDL>
IDL> d=pb->dict()
IDL> d->__setitem__, "something",2
IDL> d->__setitem__, "otherthing",3
IDL> d->__setitem__, "nothing",6
IDL> print, d->__getitem__("otherthing")
```

3

```
IDL> print, d->keys()
```

```
nothing otherthing something
```

```
IDL> print, d->values()
```

```
6    3    2
```
