Subject: Re: Solve memory problems
Posted by Michael Galloy on Wed, 14 Jan 2009 17:55:36 GMT
View Forum Message <> Reply to Message

On Jan 14, 9:23 am, Craig Markwardt <cbmarkwa...@gmail.com> wrote:
> On Jan 14, 9:03 am, David Fanning <n...@dfanning.com> wrote:
>
>> Craig Markwardt writes:
>>> Uh, like using any other high-level language that doesn't force you to
>>> free your own variables?
>
>> My goodness, people, whatever happened to craftsmanship?
>> I feel like I'm working with a bunch of Wal-Mart furniture
>> builders here. Quick, easy, cheap. :-(
>
> Uh, I wasn't defending the use of HEAP_GC, but it's monumentally silly
> that the IDL language designers had the choice to implement automatic
> freeing of dangling pointers ("garbage collection") and did not [*].
> I mean, would you really enjoy the "privilege" of freeing every
> *regular* variable before returning from each IDL procedure? Of
> course not. The IDL runtime has enough information to know *exactly*
> when a pointer becomes dangling, so why not use that information?
>
> I think I understand craftmanship -- I hope my public code speaks for
> itself. But I don't think that has anything to do with masochistic
> worship at the alter of POINTER_FREE.
>
> Snark-Craig
>
> [*] - I understand that for debugging purposes, automatic garbage
> collection may be a nuisance. It would be straightforward to disable
> it with a system variable.

I must say that garbage collection is one of the features I really appreciate when I use Python. Garbage collection is now even in OS X's objective C.

That said, IDL does not have the same garbage collection algorithms. HEAP_GC is *slow* when you have a lot of heap variables. I would support IDL adding real garbage collection, but what is currently in IDL is a debugging tool only.

Mike
--
www.michaelgalloy.com
Tech-X Corporation
Associate Research Scientist