
Subject: Re: idl2matlab translate-o-matic
Posted by [John Pezaris](#) on Fri, 25 Feb 2000 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

davidf@dfanning.com (David Fanning) writes:

- > David McClain (dmcclain@azstarnet.com) writes:
- >
- > I'm sorry, but I think this **completely** misses the point.
- > Cleaning up variables is one thing, but checking for **every**
- > pointer reference at the end of every program module that
- > exits would bring just about any program--never mind IDL--
- > to a complete stand-still. It shouldn't be done. I applaud
- > the folks at RSI for dismissing the idea out of hand.

Your're correct in the letter of what your wrote, but dead wrong in spirit. Doing a full GC at every program module -- FULL, that is, chasing down and verifying every reference -- would not be a good thing. No garbage collector I've known of has done this. But, if you look at the beautiful and efficient GCs written since 1978 or so, you'll see that they impose perhaps 1-2% run-time overhead. I've spent many years working on a system (a LispM derivative called the TI Explorer) which was written entirely (yes, **entirely**, from basic OS on up) in Lisp and had a GC running constantly. One nice thing about these system was called the "modeline" which displayed the currently active module: GC was hardly ever displayed there. More modern programming systems, such as Caml, Scheme, and the like, have very, very good GCs, and, frankly, I'd rather be using a system like that than worry about needing to explicitly allocate and deallocate objects, as in C/C++, etc., and having to trace down obscure memory leaks.

So, to summarize, GCs are not inherently bad, and, rather the opposite, are quite good. You might want to re-examine your position on the subject. I don't know anything about IDL, but, if what you say about RSI is true, then they might want to re-think their position as well.

Cheers,

- pz.

--

John Pezaris
pz@caltech.edu
