Subject: Do we need PTR_FREE anymore?
Posted by wlandsman on Thu, 29 Jul 2010 01:23:57 GMT
View Forum Message <> Reply to Message

I am trying to come to terms with the automatic garbage collection implemented in IDL 8.0. Does this mean that we never have to cleanup our pointer variables with PTR_FREE anymore? I find this a little unsettling -- like being told that there is no longer any need to brush your teeth.

The help files give one obscure example where automatic garbage collection fails -- when "two objects or pointers refer to each other, but no other object or pointer refers to either". Is there ever any other reason (e.g. performance issues) to explicitly call PTR_FREE anymore?

--Wayne

Subject: Re: Do we need PTR_FREE anymore?
Posted by James[2] on Fri, 30 Jul 2010 17:07:11 GMT
View Forum Message <> Reply to Message

Wlandsman, that example doesn't sound particularly obscure to me -- it seems like it could come up frequently when dealing with linked data structures.

It's a little unusual that IDL has added garbage collection but kept manual memory management intact. Most languages only offer one or the other. It's pretty unusual for a language as high-level as IDL to have manual memory management at all.

Subject: Re: Do we need PTR_FREE anymore?
Posted by wlandsman on Fri, 30 Jul 2010 17:29:30 GMT
View Forum Message <> Reply to Message

On Jul 30, 1:07 pm, James <donje...@gmail.com> wrote:
> Wlandsman, that example doesn't sound particularly obscure to me -- it
> seems like it could come up frequently when dealing with linked data
> structures.

I notice that early versions of Python had the same limitation, but since 2.2 (I think) it can handle the case where 2 pointers reference each other.
>
> It's a little unusual that IDL has added garbage collection but kept

> manual memory management intact. Most languages only offer one or the
> other. It's pretty unusual for a language as high-level as IDL to
> have manual memory management at all.

The manual memory management was introduced back in 1997, and they seem to have made a strong effort to keep backwards compatibility. --
Wayne

Subject: Re: Do we need PTR_FREE anymore?
Posted by Michael Galloy on Fri, 30 Jul 2010 17:29:30 GMT
View Forum Message <> Reply to Message

On 7/30/10 11:07 am, James wrote:
> Wlandsman, that example doesn't sound particularly obscure to me -- it
> seems like it could come up frequently when dealing with linked data
> structures.
>
> It's a little unusual that IDL has added garbage collection but kept
> manual memory management intact. Most languages only offer one or the
> other. It's pretty unusual for a language as high-level as IDL to
> have manual memory management at all.

But IDL has to continue to support manual memory management or else break backward compatibility. I would imagine that PTR_FREE and OBJ_DESTROY will be around for a long time.

Mike
--
www.michaelgalloy.com
Research Mathematician
Tech-X Corporation

Subject: Re: Do we need PTR_FREE anymore?
Posted by Adam Lefkoff on Fri, 30 Jul 2010 19:12:06 GMT
View Forum Message <> Reply to Message

On 7/30/2010 11:29 AM, Michael Galloy wrote:
> On 7/30/10 11:07 am, James wrote:
>> Wlandsman, that example doesn't sound particularly obscure to me -- it
>> seems like it could come up frequently when dealing with linked data
>> structures.
>>
>> It's a little unusual that IDL has added garbage collection but kept
>> manual memory management intact. Most languages only offer one or the
>> other. It's pretty unusual for a language as high-level as IDL to

>> have manual memory management at all.
>
> But IDL has to continue to support manual memory management or else
> break backward compatibility. I would imagine that PTR_FREE and
> OBJ_DESTROY will be around for a long time.
>
> Mike

HANDLE_FREE still works too!