Subject: Re: Storing !NULL in struct
Posted by lecacheux.alain on Mon, 18 Mar 2013 09:27:31 GMT

View Forum Message <> Reply to Message

Le lundi 18 mars 2013 10:00:58 UTC+1, Tom Grydeland a écrit :

```
>  On Friday, March 15, 2013 8:40:56 PM UTC, Chris Torrence wrote:
>
>>  Hi all,
>
>
>
>>  Not to stir the pot some more [...]
>
>
>
>  Not at all, it is good to hear the rationale behind decisions such as this one.
>
>
>
>>  The deal-killer was that IDL structures are supposed to map directly to C structures. [...]
>
>
>
>  I'll buy that.
>
>
>
>>  As Mike suggests, perhaps HASH is the way to go.
>
>
>
>  Either of HASH or LIST would be perfectly fine, if I were able to even assign to already-known fields of structs stored inside them:
>
>
>
>  IDL> h = hash('f', {t:0})
>
>  IDL> help, h
>
>  H               HASH  <ID=198  NELEMENTS=1>
>
>  IDL> print, h['f'].t
>
>             0
>
>  IDL> h['f'].t = 1
>
```

```
>  % Attempt to store into an expression: Structure reference.
>
>  % Execution halted at: $MAIN$
>
>  IDL> c = list({t:0})
>
>  IDL> print, c[0].t
>
>            0
>
>  IDL> c[0].t = 1
>
>  % Attempt to store into an expression: Structure reference.
>
>  % Execution halted at: $MAIN$
>
>
>
>
>
>  It's not that I cannot imagine a way of working around this, but it seems to defeat the purpose of providing high-level data structures.
>
>
>
>>  Chris
>
>
>
>  --T
```

```
IDL> h = hash('f', {t:0})
IDL> help, h
H               HASH  <ID=1  NELEMENTS=1>
IDL> print, h['f'].t
         0
IDL> h['f'] = {t:1}
IDL> print, h['f'].t
         1
```

alx.