
Subject: Re: Storing !NULL in struct

Posted by [lecacheux.alain](#) on Fri, 15 Mar 2013 13:31:18 GMT

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

Le vendredi 15 mars 2013 09:10:45 UTC+1, Tom Grydeland a écrit :

>> On 3/13/13 2:17 am, Tom Grydeland wrote:

>

>>> And it would be useful _to me_ if I could put a !NULL in a struct.

>

>>> I'm using the struct for UVALUES in widgets, and it would be nice if

>

>>> they allowed _any_ well-defined IDL value.

>

>

>

> On Wednesday, March 13, 2013 2:46:17 PM UTC, Mike Galloy wrote:

>

>> But !null is undefined:

>

>

>

> Not so undefined that I cannot assign it to a variable, or return it from a function, or use it as a good 'missing value' indicator in most cases. It is also pretty damn useful if you want to build an array by parts (although, as Bob pointed out, I could use a LIST for that last case).

>

>

>

> See, I was trying to stash away an arbitrary value (provided by the user for their later retrieval) under a given tag in a structure, and I didn't want to restrict what values they were allowed to use.

>

>

>

> So instead of using a struct { internal: <whatever>, public: <your value here> }, I thought I could use a List(<whatever>, <your value here>), but that didn't work either, since I'm not allowed to assign to a struct that lives in a List.

>

>

>

> 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: $MAIN$
>
>
>
> :-(
>
>
>
>> Mike
>
>
>
> --T
```

Are'nt you mixing three different concepts: undefined value, unassigned value and missing data ?
In IDL, you can use : !Null, New_Ptr() and !Values.F_NAN, respectively.

alx.
