
Subject: Re: HASH with case insensitive keys versus a new DICT class
Posted by chris_torrence@NOSPAM on Thu, 13 Dec 2012 03:10:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wednesday, December 12, 2012 12:09:34 PM UTC-7, Craig Markwardt wrote:

> On Wednesday, December 12, 2012 1:05:08 AM UTC-5, Chris Torrence wrote:

>

>> Okay, I'll bite.

>

>>

>

>>

>

>>

>

>> I have a specific use case for this, where using a "dynamic" structure would be very helpful. Users have also been asking for dynamic, extendable structures for years.

>

>>

>

>>

>

>>

>

>> One more data point: It would take less than a day to make this update. Adding a new widget system would take more than a day.

>

>

>

> I would go at it from the other end and make the structure access operator "." overloadable from within the IDL language. Once that key language feature is added, then your particular use case of HASH as the backing store can be wired up with a few lines of IDL code, and you can choose whatever idiosyncracies you wish.

>

>

>

> Craig

Hi Craig,

Actually, it is already overloadable. As long as your class inherits from IDL_Object, then using a "." will call GetProperty (if it is on the right-hand-side of the equals), or SetProperty (if it is on the left-hand-side).

And you're right, it was indeed just a couple of lines of code. The trickiness comes in because the IDL parser has already made the "field" name uppercase by the time it reaches the internal C structure code. I really don't want to mess with the IDL parser.

-Chris
