
Subject: Re: Case Insensitive Hash but still preserve cases of original keys

Posted by [SonicKenking](#) on Wed, 16 Jul 2014 02:44:02 GMT

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

On Tuesday, July 15, 2014 10:32:30 PM UTC+10, fawltyl...@gmail.com wrote:

> On Tuesday, July 15, 2014 1:11:06 PM UTC+2, SonicKenking wrote:

>

>

>

>> What I am confused is how PRINT gets the value for the key "X"? It seems to me that it should call something like self["X"], i.e. SpecialHash::_overloadBracketsRightSide. However it is not the case and this can be verified by setting a breakpoint as well.

>

>

>

> There is an undocumented function hash::get:

>

>

>

> IDL> print, (hash(0,0))[0]

>

> 0

>

> IDL> .comp hash_get

>

> % Compiled module: HASH::GET.

>

> IDL> print, (hash(0,0))[0]

>

> HASH::GET called: 0

>

> 0

>

> IDL>

>

>

>

> hash_get.pro is:

>

>

>

> function hash::get, key, _extra=ext

>

> print, 'HASH::GET called: ', key

>

> return, 0

>

> end

>
>
>
> PRINT probably calls self.get(...), so you can try to add a SpecialHash::Get function.
>
>
>
> regards,
>
> Lajos

Thanks Lajos. Override the Get() function solves the problem!

Inspired by your answer, I dug a bit more into the HASH internals. So HASH sometimes works off the internal data structure (e.g. TABLE_DATA) without consulting the public APIs. This creates problem for subclasses that alter the behaviors of public APIs.

One more example is that the FOREACH loop of SpecialHash display the key 'x' in lowercase, while it really should be in uppercase.

```
IDL> foreach v,h, k do print, k, v  
x 1337
```

My guess is that Hash's foreach implementation gets the keys directly from the internal self.TABLE_DATA structure. The keys() function hence never gets called and doesn't have the chance to provide the correct case of key 'X'
