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'

Subject: Re: Case Insensitive Hash but still preserve cases of original keys
Posted by chris_torrence@NOSPAM on Wed, 16 Jul 2014 21:28:22 GMT

View Forum Message <> Reply to Message

Hi all,

Just FYI, in IDL 8.4 the HASH and ORDEREDHASH will have a new FOLD_CASE keyword. This will make the Hash and OrderedHash be case insensitive but still preserve the case of the keys, which I believe is exactly what you want.

Cheers,
Chris

Subject: Re: Case Insensitive Hash but still preserve cases of original keys
Posted by SonicKenking on Thu, 17 Jul 2014 00:34:18 GMT

View Forum Message <> Reply to Message

On Thursday, July 17, 2014 7:28:22 AM UTC+10, Chris Torrence wrote:

> Hi all,
>
>
>
> Just FYI, in IDL 8.4 the HASH and ORDEREDHASH will have a new FOLD_CASE keyword. This will make the Hash and OrderedHash be case insensitive but still preserve the case of the keys, which I believe is exactly what you want.
>
>
>
> Cheers,
>
> Chris

Hi Chris,

That's a great news. Can't wait to see other new features in 8.4! Is there an estimated release date?

Thanks a lot for the response.

Cheers,
Yang

Subject: Re: Case Insensitive Hash but still preserve cases of original keys
Posted by chris_torrence@NOSPAM on Thu, 17 Jul 2014 16:26:30 GMT
View Forum Message <> Reply to Message

On Wednesday, July 16, 2014 6:34:18 PM UTC-6, SonicKenking wrote:
> On Thursday, July 17, 2014 7:28:22 AM UTC+10, Chris Torrence wrote:
>
>> Hi all,
>
>>
>
>>
>
>>
>
>> Just FYI, in IDL 8.4 the HASH and ORDEREDHASH will have a new FOLD_CASE keyword. This will make the Hash and OrderedHash be case insensitive but still preserve the case of the keys, which I believe is exactly what you want.
>
>>
>
>>
>

>>
>
>> Cheers,
>
>>
>
>> Chris
>
>
>
> Hi Chris,
>
>
>
> That's a great news. Can't wait to see other new features in 8.4! Is there an estimated release date?
>
>
>
> Thanks a lot for the response.
>
>
>
> Cheers,
>
> Yang

Should be right around November. Lots of cool features, including code coverage, big integers, lambda functions, and static methods for variables...
-Chris

Subject: Re: Case Insensitive Hash but still preserve cases of original keys
Posted by Paul Van Delst[1] on Thu, 17 Jul 2014 16:53:47 GMT

View Forum Message <> Reply to Message

lambda functions! 8o0

Code blocks in IDL?!?!

Goodness.

On 07/17/14 12:26, Chris Torrence wrote:
> On Wednesday, July 16, 2014 6:34:18 PM UTC-6, SonicKenking wrote:
>>
>> That's a great news. Can't wait to see other new features in 8.4!
>> Is there an estimated release date?
>>

>
> Should be right around November. Lots of cool features, including
> code coverage, big integers, lambda functions, and static methods for
> variables... -Chris
>

Subject: Re: Case Insensitive Hash but still preserve cases of original keys
Posted by Michael Galloy on Thu, 17 Jul 2014 21:36:16 GMT
View Forum Message <> Reply to Message

On 7/17/14, 10:26 AM, Chris Torrence wrote:
> Should be right around November. Lots of cool features, including
> code coverage, big integers, lambda functions, and static methods for
> variables... -Chris
>

Code coverage? Like which lines of a routine have been executed? That
would be an awesome feature for mgunit; I would be able to report back a
lot of useful information about the coverage of a test suite.

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation

Subject: Re: Case Insensitive Hash but still preserve cases of original keys
Posted by chris_torrence@NOSPAM on Thu, 17 Jul 2014 23:08:56 GMT
View Forum Message <> Reply to Message

On Thursday, July 17, 2014 3:36:16 PM UTC-6, Mike Galloy wrote:
>
>
> Code coverage? Like which lines of a routine have been executed? That
>
> would be an awesome feature for mgunit; I would be able to report back a
>
> lot of useful information about the coverage of a test suite.
>
>

Yep, that's exactly it. There's a standalone function you can call, and it's also integrated into the Workbench and the Profiler.

-Chris

Subject: Re: Case Insensitive Hash but still preserve cases of original keys
Posted by Fabzi on Fri, 18 Jul 2014 08:57:57 GMT
View Forum Message <> Reply to Message

On 17.07.2014 18:26, Chris Torrence wrote:
> Lots of cool features, including code coverage, big integers,
> lambda functions, and static methods for variables

Sounds like py**on but in much better because it's IDL ;-)

Code coverage surely is really great.

Are you willing to share a bit more? What will these static methods be
able to do?

Subject: Re: Case Insensitive Hash but still preserve cases of original keys
Posted by markb77 on Fri, 18 Jul 2014 14:48:58 GMT
View Forum Message <> Reply to Message

On Thursday, July 17, 2014 6:26:30 PM UTC+2, Chris Torrence wrote:

>
> Should be right around November. Lots of cool features, including code coverage, big integers, lambda functions, and static methods for variables...
>
> -Chris

cool!

+1 vote for parallel (multi-threaded) computing support which is compatible with the virtual machine

Subject: Re: Case Insensitive Hash but still preserve cases of original keys
Posted by Paul Van Delst[1] on Mon, 21 Jul 2014 16:38:01 GMT
View Forum Message <> Reply to Message

On 07/17/14 17:36, Michael Galloy wrote:
> On 7/17/14, 10:26 AM, Chris Torrence wrote:
>> Should be right around November. Lots of cool features, including
>> code coverage, big integers, lambda functions, and static methods for
>> variables... -Chris
>>

> Code coverage? Like which lines of a routine have been executed? That
> would be an awesome feature for mgunit; I would be able to report back a
> lot of useful information about the coverage of a test suite.

As an avid user of mgunit, that would be marvyplate. I am a lazy
unit-test writer... (downcast eyes) ...but isn't everyone? :o/

cheers,

paulv