

Re: Hashtable Contains() – byte arrays as keys –

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2004-09/0484.html>

From: mikeb (mailbox.google_at_nospam.mailnull.com)

Date: 09/01/04

Date: Wed, 01 Sep 2004 09:55:56 -0700

Rakesh Rajan wrote:

> *But that would again mean he will have to iterate and compare each entries*
> *right? Wonder whether there are any other efficient implementations?*

Assuming I understand his problem (that he needs the byte arrays to perform a 'deep' compare of the contents of the arrays), then yes – these interface implementations would need to generate a hash value that was dependent on the contents of the byte array and to compare the entries of the arrays.

The check for equality could certainly have some short cuts so that a member-by-member comparison might not always be needed:

- 1) if there's reference equality, the arrays are by definition equal
- 2) if the array lengths are different, then the arrays must not be equal.

The hashcode might be cacheable if the byte arrays are not changed once they are created. Indeed, once the byte arrays are added to the hashtable, it would be a bug to modify them in such a way that the hashcode (or equality test) would render different results.

There may be other techniques that can improve the efficiency (particularly if unsafe code can be used). However, my main point was that there's a way for the OP to get the behavior he needs without too much work.

Performance is of no consequence if things don't even work correctly in the first place.

>
>
> *"mikeb" wrote:*
>
>
>> *Joseph Lee wrote:*
>>

microsoft.public.dotnet.languages.csharp: Re: Hashtable Contains() – byte arrays as keys –

>>>ic, i will look into it. Thanks

>>

>>You can create your Hashtable with custom Hashcode provider and comparer.

>>

>>Look at the Hashtable(IHashCodeProvider, IComparer) constructor.

>>

>>With a Hashtable created with that constructor you can provide your own

>>methods for comparing the byte arrays instead of using the default

>>methods provided by the Array class.

>>

>>

>>>Joey

>>>

>>>"Rakesh Rajan" <RakeshRajan@discussions.microsoft.com> wrote in message

>>>news:8719FC6F-94E0-4217-8BC9-838FEEB1D665@microsoft.com...

>>>

>>>

>>>>As Bob said, it's the reference that is being tested and not the contents.

>>>

>>>If

>>>

>>>

>>>>I remember right, searching in the lines of overriding Equals and

>>>

>>>GetHashCode

>>>

>>>

>>>>might help. Sorry, I can't recollect them now...

>>>>

>>>>

>>>>"Joseph Lee" wrote:

>>>>

>>>>

>>>>

>>>>>Hi All,

>>>>>

>>>>>I am having problem when i am using hashtable to keep an array of bytes

>>>>>value as keys.

>>>>>

>>>>>Take a look at the code snippet below

>>>>>

>>>>>-----

>>>>>

>>>>>ASCIIEncoding asciiEncoder = new ASCIIEncoding();

>>>>>byte[] bArray = asciiEncoder.GetBytes("Test");

>>>>>

>>>>>Hashtable ht = new Hashtable();

>>>>>ht.Add(bArray, "Some value");

>>>>>

>>>>>byte[] bArrayNew = asciiEncoder.GetBytes("Test");

>>>>>

Re: Hashtable Contains() – byte arrays as keys –

microsoft.public.dotnet.languages.csharp: Re: Hashtable Contains() – byte arrays as keys –

>>*mikeb*

>>

--

mikeb