System.Collections.Hashtable Class

[ILASM]

```
.class public serializable Hashtable extends System.Object
implements System.ICloneable,
System.Collections.ICollection,
System.Collections.IDictionary,
System.Collections.IEnumerable
```

[C#]

```
public class Hashtable: ICloneable, ICollection,
IDictionary, IEnumerable
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 1.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

Type Attributes:

- DefaultMemberAttribute("Item") [*Note:* This attribute requires the RuntimeInfrastructure library.]

Implements:

- **System.Collections.IDictionary**
- **System.Collections.ICollection**
- **System.Collections.IEnumerable**
- **System.ICloneable**

Summary

Represents a hash table.

Inherits From: System.Object

Library: BCL

Thread Safety: This class is safe for multiple readers and a single writer.

Description

A **System.Collections.Hashtable** represents a dictionary with a constant lookup time that contains entries of associated keys and values. The type of each entry in a **System.Collections.Hashtable** is

System.Collections.DictionaryEntry. A statement that exposes each element in the collection is required to iterate over this type. [*Note:* See example.]

Objects used as keys in a **System.Collections.Hashtable** is required to either implement both **System.Object.GetHashCode** and **System.Object.Equals** or neither. Furthermore, for a particular key, these methods are required to produce the same results when called with the same parameters while that key exists in a particular **System.Collections.Hashtable**. Keys cannot be mutated while they are used in the table.

Every key in a **System.Collections.Hashtable** is required to be unique compared to every other key in the table. An object that implements **System.Collections.IComparer** can determine whether two keys are unequal. The default comparer for a key is the key's implementation of **System.Object.Equals**.

Each value in a **System.Collections.Hashtable** is required to provide its own hash function, which can be accessed by calling **System.Collections.Hashtable.GetHash**. Alternatively, if an object that implements **System.Collections.IHashCodeProvider** is passed to a **System.Collections.Hashtable** constructor, the custom hash function provided by that object is used for every value in the table.

[*Note:* The default capacity (i.e. the default number of entries that can be contained) of a **System.Collections.Hashtable** is zero.

When an entry is added to the **System.Collections.Hashtable**, the entry is placed into a bucket based on the hash code obtained from the **System.Collections.IHashCodeProvider** implementation of the table, or the **System.Object.GetHashCode** if no specific **System.Collections.IHashCodeProvider** was provided. Subsequent lookups of the key use the hash code of the key to search in only one particular bucket, substantially reducing the number of key comparisons required to find an entry.

As entries are added to a **System.Collections.Hashtable**, and the maximum capacity of the table is reached, the number of buckets in the table is automatically increased to the smallest prime number that is larger than twice the current number of buckets.

A **System.Collections.Hashtable** can safely support one writer and multiple readers concurrently. To support multiple writers, all operations are required to be done through the wrapper returned by the **System.Collections.Hashtable.Synchronized** method.]

Example

The following example shows how to iterate over the elements of a **System.Collections.Hashtable**.

[C#]

```
foreach (DictionaryEntry myEntry in myHashtable)
```

Hashtable() Constructor

```
[ILASM]
public rtspecialname specialname instance void .ctor()

[C#]
public Hashtable()
```

Summary

Constructs and initializes a new instance of the **System.Collections.Hashtable** class.

Description

The new instance is initialized with the default capacity, **System.Collections.IHashCodeProvider**, and **System.Collections.IComparer**.

Hashtable(System.Int32) Constructor

```
[ILASM]
public rtspecialname specialname instance void .ctor(int32
capacity)

[C#]
public Hashtable(int capacity)
```

Summary

Constructs and initializes a new instance of the **System.Collections.Hashtable** class with the specified initial capacity.

Parameters

Parameter	Description
capacity	A **System.Int32** that specifies the number of entries that the new **System.Collections.Hashtable** instance can initially contain.

Description

The new instance is initialized with the default **System.Collections.IHashCodeProvider** and **System.Collections.IComparer**.

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*capacity* < 0.

Hashtable(System.Collections.IHashCodeProvider, System.Collections.IComparer) Constructor

```
[ILASM]
public rtspecialname specialname instance void .ctor(class
System.Collections.IHashCodeProvider hcp, class
System.Collections.IComparer comparer)

[C#]
public Hashtable(IHashCodeProvider hcp, IComparer comparer)
```

Summary

Constructs and initializes a new instance of the **System.Collections.Hashtable** class with the specified **System.Collections.IHashCodeProvider** and the specified **System.Collections.IComparer**.

Parameters

Parameter	Description
hcp	The **System.Collections.IHashCodeProvider** that supplies the hash codes for all keys in the **System.Collections.Hashtable**; or, **null** to use the default hash code provider.
comparer	The **System.Collections.IComparer** to use to determine whether two keys are equal; or, **null** to use the default comparer.

Description

The new instance is initialized with the default capacity.

Hashtable(System.Int32, System.Collections.IHashCodeProvider, System.Collections.IComparer) Constructor

```
[ILASM]
public rtspecialname specialname instance void .ctor(int32
capacity, class System.Collections.IHashCodeProvider hcp,
class System.Collections.IComparer comparer)

[C#]
public Hashtable(int capacity, IHashCodeProvider hcp,
IComparer comparer)
```

Summary

Constructs and initializes a new instance of the **System.Collections.Hashtable** class with the specified initial capacity, the specified **System.Collections.IHashCodeProvider**, and the specified **System.Collections.IComparer**.

Parameters

Parameter	Description
capacity	A **System.Int32** that specifies the number of entries that the new **System.Collections.Hashtable** instance can initially contain.
hcp	The **System.Collections.IHashCodeProvider** that supplies the hash codes for all keys in the **System.Collections.Hashtable**; or, **null** to use the default hash code provider.
comparer	The **System.Collections.IComparer** to use to determine whether two keys are equal, or **null** to use the default comparer.

Hashtable(System.Collections.IDictionary) Constructor

```
[ILASM]
public rtspecialname specialname instance void .ctor(class
System.Collections.IDictionary d)

[C#]
public Hashtable(IDictionary d)
```

Summary

Constructs and initializes a new instance of the **System.Collections.Hashtable** class using the values of the specified **System.Collections.IDictionary**.

Parameters

Parameter	Description
d	The **System.Collections.IDictionary** used to initialize the elements of the new instance.

Description

The initial capacity of the new instance is set to the number of entries in *d*. The new instance is initialized with the default **System.Collections.IHashCodeProvider** and **System.Collections.IComparer**.

Exceptions

Exception	Condition
System.ArgumentNullException	*d* is **null**.

Hashtable(System.Collections.IDictionary, System.Collections.IHashCodeProvider, System.Collections.IComparer) Constructor

```
[ILASM]
public rtspecialname specialname instance void .ctor(class
System.Collections.IDictionary d, class
System.Collections.IHashCodeProvider hcp, class
System.Collections.IComparer comparer)

[C#]
public Hashtable(IDictionary d, IHashCodeProvider hcp,
IComparer comparer)
```

Summary

Constructs and initializes a new instance of the **System.Collections.Hashtable** class using the values of the specified **System.Collections.IDictionary**, the specified **System.Collections.IHashCodeProvider**, and the specified **System.Collections.IComparer**.

Parameters

Parameter	Description
d	The **System.Collections.IDictionary** used to initialize the elements of the new instance.
hcp	The **System.Collections.IHashCodeProvider** that supplies the hash codes for all keys in the new instance; or, **null** to use the default hash code provider.
comparer	The **System.Collections.IComparer** to use to determine whether two keys are equal in the new instance, or **null** to use the default comparer.

Description

The initial capacity of the new instance is set to the number of entries in *d*.

Exceptions

Exception	Condition

System.ArgumentNullException	*d* is **null**.

Hashtable.Add(System.Object, System.Object) Method

```
[ILASM]
.method public hidebysig virtual void Add(object key,
object value)

[C#]
public virtual void Add(object key, object value)
```

Summary

Adds an entry with the specified key and value into the current instance.

Parameters

Parameter	Description
key	The key of the entry to add.
value	The value of the entry to add.

Exceptions

Exception	Condition
System.ArgumentNullException	*key* is **null**.
System.ArgumentException	An entry with the same key already exists in the current instance.
System.NotSupportedException	The current instance is read-only or has a fixed size.

Hashtable.Clear() Method

[ILASM]

```
.method public hidebysig virtual void Clear()
```

[C#]

```
public virtual void Clear()
```

Summary

Removes all entries from the current instance.

Description

[*Note:* This method is implemented to support the **System.Collections.IDictionary** interface.]

Behaviors

As described above.

Default

The value of each key and value in the current instance is set to **null**. The **System.Collections.Hashtable.Count** property of the current instance is set to zero. The capacity of the current instance remains unchanged.

If the current instance is empty, it remains unchanged and no exception is thrown.

Exceptions

Exception	Condition
System.NotSupportedException	The current instance is read-only.

Hashtable.Clone() Method

[ILASM]

```
.method public hidebysig virtual object Clone()
```

[C#]

```
public virtual object Clone()
```

Summary

Creates a **System.Object** that is a copy of the current instance.

Return Value

A **System.Object** that is a copy of the current instance.

Description

[*Note:* This method is implemented to support the **System.ICloneable** interface.]

Behaviors

As described above.

Default

This method creates a new **System.Collections.Hashtable** instance is initialized with the same count, **System.Collections.IHashCodeProvider** implementation, and **System.Collections.IComparer** implementation as the current instance. The references to the objects contained by the current instance are copied to the new instance.

Hashtable.Contains(System.Object) Method

```
[ILASM]
.method public hidebysig virtual bool Contains(object key)

[C#]
public virtual bool Contains(object key)
```

Summary

Determines whether the current instance contains the specified key.

Parameters

Parameter	Description
key	The key to locate in the current instance.

Return Value

true if the current instance contains *key*; otherwise, **false**.

Description

[*Note:* This method is implemented to support the **System.Collections.IDictionary** interface.]

Behaviors

As described above.

Default

This method is equivalent to **System.Collections.Hashtable.ContainsKey**.

[*Note:* For the default implementation, this method has a constant (O(1)) lookup time.]

Exceptions

Exception	Condition
System.ArgumentNullException	*key* is **null**.

Hashtable.ContainsKey(System.Object) Method

```
[ILASM]
.method public hidebysig virtual bool ContainsKey(object key)

[C#]
public virtual bool ContainsKey(object key)
```

Summary

Determines whether the current instance contains an entry with the specified key.

Parameters

Parameter	Description
key	The key of the entry to locate in the current instance.

Return Value

true if the current instance contains an entry with *key*; otherwise, **false**.

Description

Behaviors

As described above.

Default

This method uses **System.Collections.Hashtable.KeyEquals** to compare *key* to the keys in the current instance.

[*Note:* For the default implementation, this method has a constant (O(1)) lookup time.]

Exceptions

Exception	Condition
System.ArgumentNullException	*key* is **null**.

Hashtable.ContainsValue(System.Object) Method

```
[ILASM]
.method public hidebysig virtual bool ContainsValue(object
value)

[C#]
public virtual bool ContainsValue(object value)
```

Summary

Determines whether the current instance contains an entry with the specified value.

Parameters

Parameter	Description
value	The value to locate in the current instance.

Return Value

true if the current instance contains an entry with *value*; otherwise, **false**.

Description

[*Note:* This method is implemented to support the **System.Collections.IDictionary** interface.]

Behaviors

As described above.

Default

This method is equivalent to **System.Collections.Hashtable.ContainsKey**.

[*Note:* For the default implementation, this method has a constant (O(1)) lookup time.]

Hashtable.CopyTo(System.Array, System.Int32) Method

```
[ILASM]
.method public hidebysig virtual void CopyTo(class
System.Array array, int32 arrayIndex)

[C#]
public virtual void CopyTo(Array array, int arrayIndex)
```

Summary

Copies the entries of the current instance to a one-dimensional **System.Array** starting at the specified index.

Parameters

Parameter	Description
array	The one-dimensional, zero-indexed **System.Array** that is the destination of the objects copied from the current instance.
arrayIndex	A **System.Int32** that specifies the zero-based index in *array* at which copying begins. This value is between 0 and *array*.Count minus the **System.Collections.Hashtable.Count** of the current index, inclusive.

Behaviors

As described above.

Default

The **System.Collections.DictionaryEntry** elements in the current instance are copied to the **System.Array** in the same order in which they are contained the current instance. If **System.Collections.DictionaryEntry** is not assignment-compatible with the type of *array*, a **System.InvalidCastException** is thrown. If an exception is thrown while copying, the state of the current instance is undefined.

Exceptions

Exception	Condition
System.ArgumentNullException	*array* is **null**.
System.ArgumentOutOfRangeException	*arrayIndex* < 0.

System.ArgumentException	*array* has more than one dimension. *arrayIndex* > *array*.Count - The **System.Collections.Hashtable.Count** of the current instance.
System.InvalidCastException	The type of the current instance is not assignment-compatible with the type of *array*.

Hashtable.GetEnumerator() Method

[ILASM]

```
.method public hidebysig virtual class
System.Collections.IDictionaryEnumerator GetEnumerator()
```

[C#]

```
public virtual IDictionaryEnumerator GetEnumerator()
```

Summary

Returns a **System.Collections.IDictionaryEnumerator** for the current instance.

Return Value

A **System.Collections.IDictionaryEnumerator** for the current instance.

Description

If the elements of the current instance are modified while an enumeration is in progress, a call to **System.Collections.IEnumerator.MoveNext** or **System.Collections.IEnumerator.Current** throws **System.InvalidOperationException**.

[*Note:* For detailed information regarding the use of an enumerator, see **System.Collections.IEnumerator**.

This property is implemented to support the **System.Collections.IList** interface.]

Behaviors

As described above.

Hashtable.GetHash(System.Object) Method

```
[ILASM]
.method family hidebysig virtual int32 GetHash(object key)

[C#]
protected virtual int GetHash(object key)
```

Summary

Generates a hash code for the specified key in the current instance.

Parameters

Parameter	Description
key	The **System.Object** whose hash code is to be generated.

Return Value

A **System.Int32** containing the hash code for *key*.

Description

This method is accessible only through this class or a derived class.

Behaviors

As described above.

Default

If the current instance was instantiated with a specific **System.Collections.IHashCodeProvider** implementation, this method uses that hash code provider; otherwise, it uses the **System.Object.GetHashCode** implementation of *key*.

Exceptions

Exception	Condition
System.ArgumentNullException	*key* is **null**.

Hashtable.KeyEquals(System.Object, System.Object) Method

```
[ILASM]
.method family hidebysig virtual bool KeyEquals(object
item, object key)

[C#]
protected virtual bool KeyEquals(object item, object key)
```

Summary

Determines whether the specified **System.Object** and the specified key in the current instance represent the same value.

Parameters

Parameter	Description
item	The **System.Object** to compare with *key*.
key	The key in the current instance to compare with *item*.

Return Value

true if *item* and *key* represent the same value; otherwise, **false**.

Description

This method is accessible only through this class or a derived class.

Behaviors

As described above.

Default

If the current instance was initialized with a specified **System.Collections.IComparer** implementation, this method uses that implementation to perform the comparison; otherwise, the **System.Object.Equals** implementation of *item* is used.

Exceptions

Exception	Condition

System.ArgumentNullException	*item* is **null**. -or- *key* is **null**.

Hashtable.Remove(System.Object) Method

```
[ILASM]
.method public hidebysig virtual void Remove(object key)

[C#]
public virtual void Remove(object key)
```

Summary

Removes the entry with the specified key from the current instance.

Parameters

Parameter	Description
key	The key of the entry to remove.

Description

[*Note:* This method is implemented to support the **System.Collections.IDictionary** interface.]

Behaviors

As described above.

Default

This method uses the **System.Object.Equals** implementation of *key* to locate it in the current instance. If *key* is found in the current instance, the values of both *key* and its associated value are set to **null**. If *key* is not found in the current instance, no exception is thrown and the current instance remains unchanged.

Exceptions

Exception	Condition
System.ArgumentNullException	*key* is **null**.
System.NotSupportedException	The current instance is read-only or has a fixed size.

Hashtable.Synchronized(System.Collections.Hashtable) Method

```
[ILASM]
.method public hidebysig static class
System.Collections.Hashtable Synchronized(class
System.Collections.Hashtable table)

[C#]
public static Hashtable Synchronized(Hashtable table)
```

Summary

Returns a synchronized (thread-safe) wrapper for the specified **System.Collections.Hashtable**.

Parameters

Parameter	Description
table	The **System.Collections.Hashtable** to synchronize.

Return Value

A synchronized (thread-safe) wrapper for *table*.

Description

This method returns a new **System.Collections.Hashtable** instance that contains values equal to the values of *table*, and provides synchronized access to those values.

If more than one thread is to write to a **System.Collections.Hashtable** concurrently, all write operations are required to be done through this wrapper.

[*Note:* A **System.Collections.Hashtable** can safely support one writer and multiple readers concurrently.]

Exceptions

Exception	Condition
System.ArgumentNullException	*table* is **null**.

Hashtable.System.Collections.IEnumerable.GetEnumerator() Method

[ILASM]

```
.method private final hidebysig virtual class
System.Collections.IEnumerator
System.Collections.IEnumerable.GetEnumerator()
```

[C#]

```
IEnumerator IEnumerable.GetEnumerator()
```

Summary

Implemented to support the **System.Collections.IEnumerable** interface. [Note: For more information, see **System.Collections.IEnumerable.GetEnumerator**.]

Hashtable.Count Property

[ILASM]

```
.property int32 ICollection.Count { public hidebysig
virtual abstract specialname int32 get_ICollection.Count()
}
```

[C#]

```
int ICollection.Count { get; }
```

Summary

Implemented to support the **System.Collections.ICollection** interface. [Note: For more information, see **System.Collections.ICollection.Count**.]

Hashtable.Count Property

[ILASM]

```
.property int32 Count { public hidebysig virtual
specialname int32 get_Count() }
```

[C#]

```
public virtual int Count { get; }
```

Summary

Gets the number of key-and-value pairs contained in the current instance.

Property Value

A **System.Int32** that specifies the number of key-and-value pairs contained in the current instance.

Description

This property is read-only.

Behaviors

As described above.

Hashtable.IsFixedSize Property

[ILASM]

```
.property bool IDictionary.IsFixedSize { public hidebysig
virtual abstract specialname bool
get_IDictionary.IsFixedSize() }
```

[C#]

```
bool IDictionary.IsFixedSize { get; }
```

Summary

Implemented to support the **System.Collections.IDictionary** interface. [Note: For more information, see **System.Collections.IDictionary.IsFixedSize**.]

Hashtable.IsFixedSize Property

[ILASM]

```
.property bool IsFixedSize { public hidebysig virtual
specialname bool get_IsFixedSize() }
```

[C#]

```
public virtual bool IsFixedSize { get; }
```

Summary

Gets a **System.Boolean** indicating whether the current instance has a fixed size.

Property Value

true if the current instance has a fixed size; otherwise, **false**.

Description

This property is a read-only.

[*Note:* Elements can be modified in, but not added to or removed from a **System.Collections.Hashtable** with a fixed size.]

Behaviors

As described above.

Default

The default value of this property is **false**.

How and When to Override

Override this property, setting it to **true**, to prevent addition or removal of entries in the current instance.

Hashtable.IsReadOnly Property

[ILASM]

```
.property bool IDictionary.IsReadOnly { public hidebysig
virtual abstract specialname bool
get_IDictionary.IsReadOnly() }
```

[C#]

```
bool IDictionary.IsReadOnly { get; }
```

Summary

Implemented to support the **System.Collections.IDictionary** interface. [Note: For more information, see **System.Collections.IDictionary.IsReadOnly**.]

Hashtable.IsReadOnly Property

[ILASM]

```
.property bool IsReadOnly { public hidebysig virtual
specialname bool get_IsReadOnly() }
```

[C#]

```
public virtual bool IsReadOnly { get; }
```

Summary

Gets a **System.Boolean** value indicating whether the current instance is read-only.

Property Value

true if the current instance is read-only; otherwise, **false**.

Description

This property is read-only.

[*Note:* Elements cannot be modified in, added to, or removed from a **System.Collections.Hashtable** that is read-only.]

Behaviors

As described above.

Default

The default value of this property is **false**.

How and When to Override

Override this property, setting it to **true**, in order to prevent the addition, removal, or modification of entries in the current instance.

Hashtable.IsSynchronized Property

[ILASM]

```
.property bool ICollection.IsSynchronized { public
hidebysig virtual abstract specialname bool
get_ICollection.IsSynchronized() }
```

[C#]

```
bool ICollection.IsSynchronized { get; }
```

Summary

Implemented to support the **System.Collections.ICollection** interface. [Note: For more information, see **System.Collections.ICollection.IsSynchronized**.]

Hashtable.IsSynchronized Property

[ILASM]

```
.property bool IsSynchronized { public hidebysig virtual
specialname bool get_IsSynchronized() }
```

[C#]

```
public virtual bool IsSynchronized { get; }
```

Summary

Gets a **System.Boolean** value indicating whether access to the current instance is synchronized (thread-safe).

Property Value

true if access to the current instance is synchronized (thread-safe); otherwise, **false**.

Description

This property is read-only.

[*Note:* This property is implemented to support the **System.Collections.ICollection** interface.

For more information regarding synchronization of access to a **System.Collections.Hashtable**, see **System.Collections.Hashtable.Synchronized**.]

Behaviors

As described above.

Default

The default value of this property is **false**.

How and When to Override

Override this property, setting it to **true**, if thread-safety can be guaranteed for the current instance. In order to obtain this safety, use **System.Collections.Hashtable.SyncRoot** or **System.Collections.Hashtable.Synchronized**.

Hashtable.Item Property

[ILASM]

```
.property object Item[object key] { public hidebysig
virtual specialname object get_Item(object key) public
hidebysig virtual specialname void set_Item(object key,
object value) }
```

[C#]

```
public virtual object this[object key] { get; set; }
```

Summary

Gets or sets the value in the current instance that is associated with the specified key.

Parameters

Parameter	Description
key	The key whose value to get or set.

Property Value

The value in the current instance that is associated with *key*. If *key* is not contained in the current instance, attempting to get it returns **null**, and attempting to set it creates a new entry using *key*.

Description

[*Note:* This property provides the ability to access a specific element in the current instance using the following notation: `myCollection[key]`.]

Behaviors

As described above.

Default

If this property is being set and *key* is already contained in the current instance, the value associated with the old key is replaced.

Exceptions

Exception	Condition

System.ArgumentNullException	*key* is **null**.
System.NotSupportedException	The property is being set and the current instance is read-only. The property is being set, *key* is not contained in the current instance, and the current instance has a fixed size.

Hashtable.Keys Property

```
[ILASM]
.property class System.Collections.ICollection
IDictionary.Keys { public hidebysig virtual abstract
specialname class System.Collections.ICollection
get_IDictionary.Keys() }
```

```
[C#]
ICollection IDictionary.Keys { get; }
```

Summary

Implemented to support the **System.Collections.IDictionary** interface. [Note: For more information, see **System.Collections.IDictionary.Keys**.]

Hashtable.Keys Property

[ILASM]

```
.property class System.Collections.ICollection Keys {
public hidebysig virtual specialname class
System.Collections.ICollection get_Keys() }
```

[C#]

```
public virtual ICollection Keys { get; }
```

Summary

Gets a **System.Collections.ICollection** containing the keys of the current instance.

Property Value

A **System.Collections.ICollection** containing the keys of the current instance.

Description

This property is read-only.

Behaviors

As described above.

Default

The order of the keys in the **System.Collections.ICollection** is unspecified, but it is the same order as the associated values in the **System.Collections.ICollection** returned by the **System.Collections.Hashtable.Values** method.

The returned **System.Collections.ICollection** is a reference to the current instance, not a static copy. Therefore, changes to the current instance continue to be reflected in the **System.Collections.ICollection**.

Hashtable.SyncRoot Property

[ILASM]

```
.property object ICollection.SyncRoot { public hidebysig
virtual abstract specialname object
get_ICollection.SyncRoot() }
```

[C#]

```
object ICollection.SyncRoot { get; }
```

Summary

Implemented to support the **System.Collections.ICollection** interface. [Note: For more information, see **System.Collections.ICollection.SyncRoot**.]

Hashtable.SyncRoot Property

[ILASM]

```
.property object SyncRoot { public hidebysig virtual
specialname object get_SyncRoot() }
```

[C#]

```
public virtual object SyncRoot { get; }
```

Summary

Gets a **System.Object** that can be used to synchronize access to the current instance.

Property Value

A **System.Object** that can be used to synchronize access to the current instance.

Description

This property is read-only.

A thread is required to perform synchronized operations only on the **System.Collections.Hashtable.SyncRoot** of a **System.Collections.Hashtable**, not directly on the table itself. This maintains proper synchronization with any other threads concurrently modifying the table.

[*Note:* This property is implemented to support the **System.Collections.ICollection** interface.]

Behaviors

As described above.

Default

This method returns a reference to the current instance.

How and When to Override

Override this property to return an object on which to lock when implementing a collection that wraps another collection (using a subset of it, for example). This is useful when providing synchronized access through two or more wrapper collections to the same underlying collection. Typically, this property returns a reference to the current instance.

Usage

Use this property to obtain a **System.Object** that can be used to synchronize access to the current instance.

Hashtable.Values Property

[ILASM]

```
.property class System.Collections.ICollection
IDictionary.Values { public hidebysig virtual abstract
specialname class System.Collections.ICollection
get_IDictionary.Values() }
```

[C#]

```
ICollection IDictionary.Values { get; }
```

Summary

Implemented to support the **System.Collections.IDictionary** interface. [Note: For more information, see **System.Collections.IDictionary.Values**.]

Hashtable.Values Property

[ILASM]

```
.property class System.Collections.ICollection Values {
public hidebysig virtual specialname class
System.Collections.ICollection get_Values() }
```

[C#]

```
public virtual ICollection Values { get; }
```

Summary

Gets a **System.Collections.ICollection** containing the values of the current instance.

Property Value

A **System.Collections.ICollection** containing the values of the current instance.

Description

This property is read-only.

Behaviors

As described above.

Default

The order of the values in the **System.Collections.ICollection** is unspecified, but it is the same order as the associated keys in the **System.Collections.ICollection** returned by the **System.Collections.Hashtable.Keys** method.

The returned **System.Collections.ICollection** is a reference to the current instance, not a static copy. Therefore, changes to the current instance continue to be reflected in the **System.Collections.ICollection**.