

System.Collections.IDictionary Interface

```
[ILAsm]
.class interface public abstract IDictionary implements
System.Collections ICollection, System.Collections.IEnumerable

[C#]
public interface IDictionary: ICollection, IEnumerable
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 2.0.x.x
- *Attributes:*
 - CLSCompliantAttribute(true)

Type Attributes:

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

Implements:

- **System.Collections.ICollection**
- **System.Collections.IEnumerable**

Summary

Implemented by classes that support collections of associated keys and values (i.e. dictionaries).

Library: BCL

Description

[*Note:* Each key-value pair must have a unique non-null key, but the value of an association can be any object reference, including a null reference. The `System.Collections.IDictionary` interface allows the contained keys and values to be enumerated, but it does not imply any particular sort order.

`System.Collections.IDictionary` implementations fall into three categories: read-only, fixed-size, variable-size. A read-only implementation cannot be modified. A fixed-size implementation does not allow the addition or removal of elements, but it allows the modification of existing elements. A variable-size implementation allows the addition, removal and modification of elements.

]

I Dictionary.Add(System.Object, System.Object) Method

```
[ILAsm]  
.method public hidebysig virtual abstract void Add(object key, object  
value)  
  
[C#]  
void Add(object key, object value)
```

Summary

Adds an entry with the provided key and value to the current instance.

Parameters

Parameter	Description
<i>key</i>	The System.Object to use as the key of the entry to add.
<i>value</i>	The System.Object to use as the value of the entry to add.

Description

If the specified key already exists in the current instance, this method throws a `System.ArgumentException` exception but does not modify the associated value.

Behaviors

As described above.

Exceptions

Exception	Condition
System.ArgumentNullException	<i>key</i> 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.

I Dictionary.Clear() Method

```
[ILAsm]  
.method public hidebysig virtual abstract void Clear()  
  
[C#]  
void Clear()
```

Summary

Removes all key and value pairs from the current instance.

Behaviors

As described above.

Exceptions

Exception	Condition
System.NotSupportedException	The System.Collections.IDictionary is read-only.

I Dictionary.Contains(System.Object) Method

```
[ILAsm]  
.method public hidebysig virtual abstract bool Contains(object key)  
  
[C#]  
bool Contains(object key)
```

Summary

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

Parameters

Parameter	Description
<i>key</i>	The key to locate in the System.Collections.IDictionary.

Return Value

true if the System.Collections.IDictionary contains an entry with the key;
otherwise, false.

Behaviors

As described above.

Exceptions

Exception	Condition
System.ArgumentNullException	<i>key</i> is null.

1 IDictionary.GetEnumerator() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual abstract class  
4 System.Collections.IDictionaryEnumerator GetEnumerator()  
  
5 [C#]  
6 IDictionaryEnumerator GetEnumerator()
```

7 Summary

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

9 Return Value

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

11 Description

12 [Note: For detailed information regarding the use of an enumerator, see
13 System.Collections.IEnumerator.]
14
15

16 Behaviors

17 As described above.
18
19

I Dictionary.Remove(System.Object) Method

```
[ILAsm]  
.method public hidebysig virtual abstract void Remove(object key)  
  
[C#]  
void Remove(object key)
```

Summary

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

Parameters

Parameter	Description
<i>key</i>	The key of the entry to remove.

Behaviors

If *key* is not found in the current instance, no exception is thrown and the current instance remains unchanged.

Exceptions

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

I Dictionary.IsFixedSize Property

```
[ILAsm]  
.property bool IsFixedSize { public hidebysig virtual abstract specialname  
bool get_IsFixedSize() }  
  
[C#]  
bool IsFixedSize { get; }
```

Summary

Gets a value 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 read-only.

[*Note:* A collection with a fixed size does not allow the addition or removal of elements, but does allow the modification of existing elements.]

Behaviors

As described above.

Default

The default of this property is false.

How and When to Override

Override this method, setting the value as true, to prevent the addition and removal of the elements in the current instance.

1 I Dictionary.IsReadOnly Property

```
2 [ILAsm]  
3 .property bool IsReadOnly { public hidebysig virtual abstract specialname  
4 bool get_IsReadOnly() }  
  
5 [C#]  
6 bool IsReadOnly { get; }
```

7 Summary

8 Gets a value indicating whether the current instance is read-only.

9 Property Value

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

11 Description

12 This property is read-only.

13
14 [Note: A collection that is read-only does not allow the addition, removal, or
15 modification of elements.]
16
17

18 Behaviors

19 As described above.
20

21 Default

22 The default of this property is false.
23

24 How and When to Override

25 Override this method, setting the value as true, to prevent the addition, removal, and
26 modification of the elements in the current instance.
27
28

Dictionary.Item Property

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

[C#]
object this[object key] { get; set; }
```

Summary

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

Parameters

Parameter	Description
<i>key</i>	The key of the element to get or set.

Property Value

The element with the specified key.

Description

[*Note:* This property provides the ability to access a specific element in the collection by using the following syntax: `myCollection[index].`]

Behaviors

When setting this property, if the specified key already exists in the current instance, the value is required to be replaced; otherwise, a new element is required to be created.

Exceptions

Exception	Condition
System.ArgumentNullException	<i>key</i> is null.
System.NotSupportedException	The property is set and the current instance is read-only.

	The property is set, <i>key</i> does not exist in the collection, and the current instance has a fixed size.
--	--

1

2

I Dictionary.Keys Property

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

[C#]
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.

1 Dictionary.Values Property

```
2 [ILAsm]  
3 .property class System.Collections.ICollection Values { public hideby sig  
4 virtual abstract specialname class System.Collections.ICollection  
5 get_Values() }  
  
6 [C#]  
7 ICollection Values { get; }
```

8 Summary

9 Gets a System.Collections.ICollection containing the values in the current instance.

10 Property Value

11 A System.Collections.ICollection containing the values in the current instance.

12 Description

13 This property is read-only.

14 Behaviors

15 As described above.