System.Collections.Generic.IDictionary<TKey, TValue> Interface

```
[ILAsm]
.class interface public abstract IDictionary`2<TKey,TValue> implements
System.Collections.Generic.ICollection`1<valuetype
System.Collections.Generic.KeyValuePair`2<!0,!1>>,
System.Collections.Generic.IEnumerable`1<valuetype
System.Collections.Generic.KeyValuePair`2<!0,!1>>

[C#]
public interface IDictionary<TKey,TValue>:
ICollection<KeyValuePair<TKey,TValue>>,
IEnumerable<KeyValuePair<TKey,TValue>>
```

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.Generic.ICollection<KeyValuePair<TKey,TValue>>**
- **System.Collections.Generic.IEnumerable<KeyValuePair<TKey,TValue>>**

Summary

Represents a generic collection of key/value pairs.

Library: BCL

Description

This interface class is the base interface for generic collections of key/value pairs. The implementing class must have a method for comparing keys.

Each element is a key/value pair stored in a key value pair object.

Each pair must have a non-`null` key unique according to the comparison method of the class implementing this interface. The value can be `null` and need not be unique. The `System.Collections.Generic.IDictionary<TKey,TValue>` interface allows the contained keys and values to be enumerated, but it does not imply any particular sort order.

Some implementations of this interface might permit null keys, and some might not. A dictionary implementation that prohibits null keys shall throw `System.ArgumentNullException` whenever a method or indexer is called with a null key.

IDictionary<TKey,TValue>.Add(TKey, TValue) Method

```
[ILAsm]
.method public hidebysig virtual abstract void Add(!0 key, !1 value)

[C#]
void Add(TKey key, TValue value)
```

Summary

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

Parameters

Parameter	Description
key	The TKey to use as the key of the entry to add.
value	The TValue to use as the value of the entry to add.

Description

You can also use the `System.Collections.Generic.IDictionary<TKey,TValue>.Item(TKey)` property to add new elements by setting the value of a key that does not exist in the dictionary. However, if the specified key already exists in the dictionary, setting the `System.Collections.Generic.IDictionary<TKey,TValue>.Item(TKey)` property overwrites the old value. In contrast, the `System.Collections.Generic.IDictionary<TKey,TValue>.Add(TKey,TValue)` method does not modify existing elements.

Implementations can vary in how they determine equality of objects.

Exceptions

Exception	Condition
System.ArgumentException	An entry with the same key already exists in the current instance.

System.NotSupportedException	The current instance is read-only.

1
2

3

IDictionary<TKey,TValue>.ContainsKey(TKey) Method

```
[ILAsm]
.method public hidebysig virtual abstract bool ContainsKey(!0 key)

[C#]
bool ContainsKey(TKey key)
```

Summary

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

Parameters

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

Return Value

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

Description

Implementations can vary in how they determine equality of objects.

IDictionary<TKey,TValue>.Remove(TKey) Method

```
[ILAsm]
.method public hidebysig virtual abstract bool Remove(!0 key)

[C#]
bool Remove(TKey key)
```

Summary

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

Parameters

Parameter	Description
key	The key of the entry to remove.

Return Value

`true` if the element is successfully removed; otherwise, `false`.[*Note:* This method also returns `false` if *key* was not found.

]

Description

Implementations can vary in how they determine equality of objects.

Exceptions

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

1

IDictionary<TKey,TValue>.Item Property

```
[ILAsm]
.property !1 Item[!0 key] { public hidebysig virtual abstract specialname
!1 get_Item(!0 key) public hidebysig virtual abstract specialname void
set_Item(!0 key, !1 value) }

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

Summary

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

Parameters

Parameter	Description
key	The key of the element to get or set.

Property Value

The value associated with the given key.

Description

This property provides the ability to access a specific element in the collection.

You can also use the `System.Collections.Generic.IDictionary<TKey,TValue>.Item(TKey)` property to add new elements by setting the value of a key that does not exist in the dictionary. However, if the specified key already exists in the dictionary, setting the `System.Collections.Generic.IDictionary<TKey,TValue>.Item(TKey)` property overwrites the old value. In contrast, the `System.Collections.Generic.IDictionary<TKey,TValue>.Add(TKey,TValue)` method does not modify existing elements.

Implementations can vary in how they determine equality of objects.

Exceptions

Exception	Condition
System.ArgumentException	The property is read but *key* is not found.
System.NotSupportedException	The property is set and the current instance is read-only.

1
2
3

IDictionary<TKey,TValue>.Keys Property

```
[ILAsm]
.property class System.Collections.Generic.ICollection`1<!0> Keys { public
hidebysig virtual abstract specialname class
System.Collections.Generic.ICollection`1<!0> get_Keys() }

[C#]
ICollection<TKey> Keys { get; }
```

Summary

Gets a collection containing the keys of the current instance.

Property Value

A collection containing the keys of the current instance.

Description

This property is read-only.

The order of the keys in the returned `System.Collections.Generic.ICollection<TKey>` is unspecified, but it is guaranteed to be the same order as the corresponding values in the collection returned by the `System.Collections.Generic.IDictionary<TKey,TValue>.Values` property.

IDictionary<TKey,TValue>.Values Property

```
[ILAsm]
.property class System.Collections.Generic.ICollection`1<!1> Values {
public hidebysig virtual abstract specialname class
System.Collections.Generic.ICollection`1<!1> get_Values() }

[C#]
ICollection<TValue> Values { get; }
```

Summary

Gets a collection containing the values in the current instance.

Property Value

A collection containing the values in the current instance.

Description

This property is read-only.

The order of the values in the returned `System.Collections.Generic.ICollection<TKey>` is unspecified, but it is guaranteed to be the same order as the corresponding keys in the collection returned by the `System.Collections.Generic.IDictionary<TKey,TValue>.Keys` property.