# System.Collections.Generic.Dictionary<TKey, TValue>.KeyCollection Class

```
[ILAsm]
.class sealed nested public serializable
System.Collections.Generic.Dictionary`2/KeyCollection<TKey,TValue> extends
System.Object implements System.Collections.Generic.ICollection`1<!0>,
System.Collections.Generic.IEnumerable`1<!0>,
System.Collections.ICollection, System.Collections.IEnumerable

[C#]
public sealed class Dictionary<TKey,TValue>.KeyCollection:
ICollection<TKey>, IEnumerable<TKey>, 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)

## Implements:

- **System.Collections.ICollection**
- **System.Collections.IEnumerable**
- **System.Collections.Generic.ICollection<TKey>**
- **System.Collections.Generic.IEnumerable<TKey>**

## Summary

Represents a read-only collection of keys in a dictionary.

## Inherits From: System.Object

**Library:** BCL

**Thread Safety:** Static members of this type are thread-safe. Any instance members are not guaranteed to be thread-safe.

## Description

The `System.Collections.Generic.Dictionary<TKey,TValue>.Keys` property returns an instance of this type, containing all the keys in that dictionary. The order of the keys in the key collection is unspecified, but it is the same order as the associated values in the value collection returned by the `System.Collections.Generic.Dictionary<TKey,TValue>.Values` property.

If the underlying dictionary is modified, or the value of any key in that dictionary is modified, the behavior of the key collection is unspecified.

# Dictionary<TKey,TValue>.KeyCollection(System.Collections.Generic.Dictionary<TKey,TValue>) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(class
System.Collections.Generic.Dictionary`2<!2,!3> dictionary)

[C#]
public KeyCollection(Dictionary<TKey,TValue> dictionary)
```

## Summary

Initializes a new key collection to reflect the keys from the specified dictionary.

## Parameters

| Parameter | Description |
|---|---|
| *dictionary* | The dictionary whose keys are to be reflected in the key collection. |

## Exceptions

| Exception | Condition |
|---|---|
| **System.ArgumentNullException** | *dictionary* is `null`. |

# Dictionary<TKey,TValue>.KeyCollection.CopyTo(TKey[], System.Int32) Method

[ILAsm]

```
.method public hidebysig void CopyTo(!2[] array, int32 arrayIndex)
```

[C#]

```
public void CopyTo(TKey[] array, int arrayIndex)
```

## Summary

Copies the elements of the key collection to a `System.Array`, starting at a particular index.

## Parameters

| Parameter | Description |
|---|---|
| *array* | A one-dimensional, zero-based `System.Array` that is the destination of the elements copied from the key collection. |
| *arrayIndex* | The zero-based index in *array* at which copying begins. |

## Description

The elements are copied onto the array in the same order in which the enumerator iterates through the key collection.

## Exceptions

| Exception | Condition |
|---|---|
| **System.ArgumentException** | *array* is multidimensional.<br>-or-<br>*array* does not have zero-based indexing.<br>-or-<br>*arrayIndex* is greater than the length of *array*.<br>-or- |

| | |
|---|---|
| | The number of elements in the key collection is greater than the available space from *arrayIndex* to the end of the destination *array*.<br>-or-<br>Some element of the key collection is not assignable to the element type of the array. |
| **System.ArgumentNullException** | *array* is `null`. |
| **System.ArgumentOutOfRangeException** | *arrayIndex* < 0. |

# Dictionary<TKey,TValue>.KeyCollection.GetEnumerator() Method

```
[ILAsm]
.method public hidebysig virtual class
System.Collections.Generic.Dictionary`2/KeyCollection/Enumerator<!0,!1>
GetEnumerator()

[C#]
public Dictionary<TKey,TValue>.KeyCollection.Enumerator GetEnumerator()
```

## Summary

Returns an enumerator that can be used to iterate over the key collection.

## Return Value

An enumerator for the key collection.

## Description

If the underlying dictionary is modified, or the value of any key in that dictionary is modified, the behavior of the enumerator is unspecified.

## Usage

For a detailed description regarding the use of an enumerator, see `System.Collections.Generic.IEnumerator<TKey>`.

# Dictionary<TKey,TValue>.KeyCollection.System.Collections.Generic.ICollection<TKey>.Add (TKey) Method

```
[ILAsm]
.method private hidebysig virtual final void
System.Collections.Generic.ICollection`1<TKey>.Add(!0 item)

[C#]
void ICollection<TKey>.Add(TKey item)
```

## Summary

This method is implemented to support the `System.Collections.Generic.ICollection<TKey>` interface. This method throws an exception of type `System.NotSupportedException`.

# Dictionary<TKey,TValue>.KeyCollection.System.Collections.Generic.ICollection<TKey>.Clear() Method

[ILAsm]

```
.method private hidebysig virtual final void
System.Collections.Generic.ICollection`1<TKey>.Clear()
```

[C#]

```
void ICollection<TKey>.Clear()
```

## Summary

This method is implemented to support the `System.Collections.Generic.ICollection<TKey>` interface. This method throws an exception of type `System.NotSupportedException`.

# Dictionary<TKey,TValue>.KeyCollection.System.Collections.Generic.ICollection<TKey>.Contains(TKey) Method

```
[ILAsm]
.method private hidebysig virtual final bool
System.Collections.Generic.ICollection`1<TKey>.Contains(!0 item)

[C#]
bool ICollection<TKey>.Contains(TKey item)
```

## Summary

This method is implemented to support the `System.Collections.Generic.ICollection<TKey>` interface.

# Dictionary<TKey,TValue>.KeyCollection.System.Collections.Generic.ICollection<TKey>.Remove(TKey) Method

```
[ILAsm]
.method private hidebysig virtual final bool
System.Collections.Generic.ICollection`1<TKey>.Remove(!0 item)

[C#]
bool ICollection<TKey>.Remove(TKey item)
```

## Summary

This method is implemented to support the `System.Collections.Generic.ICollection<TKey>` interface. This method throws an exception of type `System.NotSupportedException`.

# Dictionary<TKey,TValue>.KeyCollection.System.Collections.Generic.IEnumerable<TKey>.GetEnumerator() Method

```
[ILAsm]
.method private hidebysig virtual final class
System.Collections.Generic.IEnumerator`1<TKey>
System.Collections.Generic.IEnumerable`1<TKey>.GetEnumerator()

[C#]
IEnumerator<TKey> IEnumerable<TKey>.GetEnumerator()
```

## Summary

This method is implemented to support the `System.Collections.Generic.IEnumerable<TKey>` interface.

## Description

If the underlying dictionary is modified, or the value of any key in that dictionary is modified, the behavior of the enumerator is unspecified.

# Dictionary<TKey,TValue>.KeyCollection.System.Collections.ICollection.CopyTo(System.Array, System.Int32) Method

[ILAsm]

```
.method private hidebysig virtual final void
System.Collections.ICollection.CopyTo(class System.Array array, int32
index)
```

[C#]

```
void ICollection.CopyTo(Array array, int index)
```

## Summary

This method is implemented to support the `System.Collections.ICollection` interface.

# Dictionary<TKey,TValue>.KeyCollection.System.Collections.IEnumerable.GetEnumerator() Method

[ILAsm]

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

[C#]

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

## Summary

This method is implemented to support the `System.Collections.IEnumerable` interface.

## Description

If the underlying dictionary is modified, or the value of any key in that dictionary is modified, the behavior of the enumerator is unspecified.

# Dictionary<TKey,TValue>.KeyCollection.Count Property

```
[ILAsm]
.property int32 Count { public hidebysig specialname instance int32
get_Count () }

[C#]
public int Count { get; }
```

## Summary

Gets the number of elements contained in the key collection.

## Property Value

The number of elements in the key collection.

## Description

This property is read-only.

# Dictionary<TKey,TValue>.KeyCollection.System.Collections.Generic.ICollection<TKey>.IsReadOnly Property

[ILAsm]

```
.property bool System.Collections.Generic.ICollection`1<TKey>.IsReadOnly {
private hidebysig virtual final specialname bool get_IsReadOnly() }
```

[C#]

```
bool ICollection<TKey>.IsReadOnly { get; }
```

## Summary

This read-only property is implemented to support the `System.Collections.Generic.ICollection<TKey>` interface. This property returns true.

# Dictionary<TKey,TValue>.KeyCollection.System.Collections.ICollection.IsSynchronized Property

[ILAsm]

```
.property bool System.Collections.ICollection.IsSynchronized { private
hidebysig virtual final specialname bool get_IsSynchronized() }
```

[C#]

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

## Summary

This read-only property is implemented to support the `System.Collections.ICollection` interface. This property returns false.

# Dictionary<TKey,TValue>.KeyCollection.System.Collections.ICollection.SyncRoot Property

[ILAsm]

```
.property object System.Collections.ICollection.SyncRoot { private
hidebysig virtual final specialname object get_SyncRoot() }
```

[C#]

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

## Summary

This property is implemented to support the `System.Collections.ICollection` interface.

## Description

This read-only property returns the `SyncRoot` property of the underlying dictionary.