System.Collections.Generic.ISet<T> Interface

[ILAsm]

```
.class public interface abstract ISet<T> implements
System.Collections.Generic.ICollection`1<!0>,
System.Collections.Generic.IEnumerable`1<!0>,
System.Collections.IEnumerable
```

[C#]

```
public interface ISet<T>: System.Collections.Generic.ICollection<T>,
System.Collections.Generic.IEnumerable<T>
```

Assembly Info:

- *Name:* System
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 4.0.0.0
- *Attributes:*
    - CLSCompliantAttribute(true)

Implements:

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

Summary

Provides the base interface for the abstraction of sets.

Library: BCL

Description

This interface provides methods for implementing sets, which are collections that have unique elements and specific operations.

[*Note:* The definition of uniqueness is defined by a class implementing this interface.

]

ISet<T>.Add(T) Method

[ILAsm]

```
.method public hidebysig newslot abstract virtual instance bool Add(!0
item) cil managed
```

[C#]

```
public bool Add (T item)
```

Summary

Adds an element to the current set and returns a value to indicate if the element was successfully added.

Parameters

Parameter	**Description**
item	The element to add to the set.

Return Value

`true` if the element is added to the set; `false` if the element is already in the set.

ISet<T>.ExceptWith(System.Collections.Generic.IEnumerable<T>) Method

```
[ILAsm]
.method public hidebysig newslot abstract virtual instance void
ExceptWith(class System.Collections.Generic.IEnumerable`1<!0> other) cil
managed

[C#]
public void ExceptWith (System.Collections.Generic.IEnumerable<T> other)
```

Summary

Removes all elements from the current set that are in the specified collection *other*

Parameters

Parameter	**Description**
other	The collection of items to remove from the set.

Description

This method is an O(*n*) operation, where *n* is the number of elements in the *other* parameter.

Exceptions

Exception	**Condition**
System.ArgumentNullException	*other* is `null`.

ISet<T>.IntersectWith(System.Collections.Generic.IEnumerable<T>) Method

```
[ILAsm]
.method public hidebysig newslot abstract virtual instance void
IntersectWith(class System.Collections.Generic.IEnumerable`1<!0> other)
cil managed

[C#]
public void IntersectWith (System.Collections.Generic.IEnumerable<T>
other)
```

Summary

Removes all elements from the current set that are not in the specified collection *other*

Parameters

Parameter	**Description**
other	The collection to compare to the current set.

Description

This method ignores any duplicate elements in *other*.

Exceptions

Exception	**Condition**
System.ArgumentNullException	*other* is `null`.

ISet<T>.IsProperSubsetOf(System.Collections.Generic.IEnumerable<T>) Method

```
[ILAsm]
.method public hidebysig newslot abstract virtual instance bool
IsProperSubsetOf(class System.Collections.Generic.IEnumerable`1<!0> other)
cil managed

[C#]
public bool IsProperSubsetOf (System.Collections.Generic.IEnumerable<T>
other)
```

Summary

Determines whether the current set is a property (strict) subset of a specified collection.

Parameters

Parameter	Description
other	The collection to compare to the current set.

Return Value

`true` if the current set is a correct subset of *other*; otherwise, `false`.

Description

If the current set is a proper subset of *other*, *other* must have at least one element that the current set does not have.

An empty set is a correct subset of any other collection. Therefore, this method returns `true` if the current set is empty, unless the *other* parameter is also an empty set.

Exceptions

Exception	Condition
System.ArgumentNullException	*other* is `null`.

ISet<T>.IsProperSupersetOf(System.Collections.Generic.IEnumerable<T>) Method

```
[ILAsm]
.method public hidebysig newslot abstract virtual instance bool
IsProperSupersetOf(class System.Collections.Generic.IEnumerable`1<!0>
other) cil managed

[C#]
public bool IsProperSupersetOf (System.Collections.Generic.IEnumerable<T>
other)
```

Summary

Determines whether the current set is a correct superset of a specified collection.

Parameters

Parameter	Description
other	The collection to compare to the current set.

Return Value

`true` if the current set is a correct superset of *other*; otherwise, `false`.

Description

If the current set is considered a proper superset of *other*, the current set must have at least one element that *other* does not have.

Exceptions

Exception	Condition
System.ArgumentNullException	*other* is `null`.

ISet<T>.IsSubsetOf(System.Collections.Generic.IEnumerable<T>) Method

```
[ILAsm]
.method public hidebysig newslot abstract virtual instance bool
IsSubsetOf(class System.Collections.Generic.IEnumerable`1<!0> other) cil
managed

[C#]
public bool IsSubsetOf (System.Collections.Generic.IEnumerable<T> other)
```

Summary

Determines whether a set is a subset of a specified collection.

Parameters

Parameter	Description
other	The collection to compare to the current set.

Return Value

`true` if the current set is a subset of *other*; otherwise, `false`.

Description

If *other* contains the same elements as the current set, the current set is still considered a subset of other.

This method always returns `false` if the current set has elements that are not in *other*.

Exceptions

Exception	Condition
System.ArgumentNullException	*other* is `null`.

ISet<T>.IsSupersetOf(System.Collections.Generic.IEnumerable<T>) Method

```
[ILAsm]
.method public hidebysig newslot abstract virtual instance bool
IsSupersetOf(class System.Collections.Generic.IEnumerable`1<!0> other) cil
managed

[C#]
public bool IsSupersetOf (System.Collections.Generic.IEnumerable<T> other)
```

Summary

Determines whether the current set is a superset of a specified collection.

Parameters

Parameter	Description
other	The collection to compare to the current set.

Return Value

`true` if the current set is a superset of *other*; otherwise, `false`.

Description

If *other* contains the same elements as the current set, the current set is still considered a superset of *other*.

This method always returns `false` if the current set has fewer elements than *other*.

Exceptions

Exception	Condition
System.ArgumentNullException	*other* is `null`.

ISet<T>.Overlaps(System.Collections.Generic.IEnumerable<T>) Method

```
[ILAsm]
.method public hidebysig newslot abstract virtual instance bool
Overlaps(class System.Collections.Generic.IEnumerable`1<!0> other) cil
managed

[C#]
public bool Overlaps (System.Collections.Generic.IEnumerable<T> other)
```

Summary

Determines whether the current set overlaps with the specified collection.

Parameters

Parameter	Description
other	The collection to compare to the current set.

Return Value

`true` if the current set and *other* share at least one common element; otherwise, `false`.

Description

Any duplicate elements in *other* are ignored.

Exceptions

Exception	Condition
System.ArgumentNullException	*other* is `null`.

ISet<T>.SetEquals(System.Collections.Generic.IEnumerable<T>) Method

```
[ILAsm]
.method public hidebysig newslot abstract virtual instance bool
SetEquals(class System.Collections.Generic.IEnumerable`1<!0> other) cil
managed

[C#]
public bool SetEquals (System.Collections.Generic.IEnumerable<T> other)
```

Summary

Determines whether the current set and the specified collection contain the same elements.

Parameters

Parameter	**Description**
other	The collection to compare to the current set.

Return Value

`true` if the current set is equal to *other*; otherwise, false.

Description

This method ignores the order of elements and any duplicate elements in *other*.

Exceptions

Exception	**Condition**
System.ArgumentNullException	*other* is `null`.

ISet<T>.SymmetricExceptWith(System.Collections.Generic.IEnumerable<T>) Method

```
[ILAsm]
.method public hidebysig newslot abstract virtual instance void
SymmetricExceptWith(class System.Collections.Generic.IEnumerable`1<!0>
other) cil managed

[C#]
public void SymmetricExceptWith (System.Collections.Generic.IEnumerable<T>
other)
```

Summary

Modifies the current set so that it contains only elements that are present either in the current set or in the specified collection, but not both.

Parameters

Parameter	Description
other	The collection to compare to the current set.

Description

Any duplicate elements in *other* are ignored.

Exceptions

Exception	Condition
System.ArgumentNullException	*other* is `null`.

ISet<T>.UnionWith(System.Collections.Generic.IEnumerable<T>) Method

```
[ILAsm]
.method public hidebysig newslot abstract virtual instance void
UnionWith(class System.Collections.Generic.IEnumerable`1<!0> other) cil
managed

[C#]
public void UnionWith (System.Collections.Generic.IEnumerable<T> other)
```

Summary

Modifies the current set so that it contains all elements that are present in both the current set and in the specified collection.

Parameters

Parameter	Description
other	The collection to compare to the current set.

Description

Any duplicate elements in *other* are ignored.

Exceptions

Exception	Condition
System.ArgumentNullException	*other* is `null`.