Range<T> Members

Namespace: IPS7Lnk
Assemblies: IPS7LnkNet.Advanced.dll
The Range<T> struct defines the following members.

Constructors

Range(T, T)

Initializes a new instance of the Range´1 struct with the specified `start` and `end` values.

C#

```
public Range(T start, T end)
```

Parameters

`start` T

The lower bound of the region.

`end` T

The upper bound of the region.

Exceptions

ArgumentNullException

Either `start` or `end` is a null reference (Nothing in Visual Basic).

ArgumentOutOfRangeException

The `start` value is lower or equal to the `end`.

Fields

Empty

Represents a Range´1 structure with its properties left uninitialized.

C#

```
public static readonly Range<T> Empty
```

Field Value
Range`1

Properties

End

Gets the upper bound of the range.

C#

```
public T End { get; }
```

Property Value

T

The upper bound of the range.

IsEmpty

Tests whether all properties of this Range´1 have default values of T.

C#

```
public bool IsEmpty { get; }
```

Property Value

Boolean

The value true if all properties of this Range´1 have default values; otherwise the value false.

Start

Gets the lower bound of the range.

C#

```
public T Start { get; }
```

Property Value

T

The lower bound of the range.

Methods

Contains(Range<T>)

Determines if the specified `range` is contained within this Range´1 structure.

C#

```
public bool Contains(Range<T> range)
```

Parameters

`range` Range`1

The Range´1 to test.

Returns

Boolean

This methods returns true if the `range` is contained within this Range´1 structure; otherwise false.

Contains(T)

Determines if the specified `value` is contained within this Range´1 structure.

C#

```
public bool Contains(T value)
```

Parameters

`value` T

The `T` to test.

Returns

Boolean

This method returns true if the `value` is contained within this Range´1 structure; otherwise false.

Equals(Object)

Tests whether `obj` is a Range´1 structure with the same Start and End of this Range´1.

C#

```
public override bool Equals(object obj)
```

Parameters

`obj` Object

The Object to test.

Returns

Boolean

This method returns true if `obj` is a Range´1 and Start and End properties are equal to the corresponding properties of this Range´1 structure; otherwise false.

Equals(Range<T>)

Tests whether `other`Range´1 structure does have the same Start and End of this Range´1.

C#

```
public bool Equals(Range<T> other)
```

Parameters

`other` Range`1

The Range´1 to test.

Returns

Boolean

This method returns true if the Start and End properties are equal to the corresponding properties of this Range´1 structure; otherwise false.

GetHashCode()

Returns the hash code for this Range´1 structure. For information about the use of hash codes, see GetHashCode.

C#

```
public override int GetHashCode()
```

Returns

Int32

An integer that represents the hash code for this range.

Intersect(Range<T>, Range<T>)

Returns a third Range´1 structure that represents the intersection of two other Range´1 structures. If there is no intersection an empty Range´1 is returned.

C#

```
public static Range<T> Intersect(Range<T> a, Range<T> b)
```

Parameters

a Range`1

A range to intersect.

b Range`1

A range to intersect.

Returns

Range`1

A Range´1 that represents the intersection of a and b.

IntersectsWith(Range<T>)

Determines if this range intersects with `range`.

C#

```
public bool IntersectsWith(Range<T> range)
```

Parameters

`range` Range`1

The range to rest.

Returns

Boolean

This method returns true if there is any intersection; otherwise false.

ToString()

Converts the attributes of this Range´1 to a human-readable string.

C#

```
public override string ToString()
```

Returns

String

A string that contains the start and end of this Range´1 structure for example, {Start=0, End=10}.

Union(Range<T>, Range<T>)

Returns a Range´1 structure that contains the union of two Range´1 structures.

C#

```
public static Range<T> Union(Range<T> a, Range<T> b)
```

Parameters

a Range`1

A rectangle to union.

b Range`1

A rectangle to union.

Returns

Range`1

A Range´1 structure that bounds the union of the two Range´1 structures.

Operators

Equality(Range<T>, Range<T>)

Tests whether two Range´1 structures have equal start and end.

C#

```
public static bool operator ==(Range<T> left, Range<T> right)
```

Inequality(Range<T>, Range<T>)

Tests whether two Range´1 structures differ.

C#

```
public static bool operator !=(Range<T> left, Range<T> right)
```

Table of Contents

Constructors 1
Range(T, T) 1
Fields 1
Empty 1
Properties 2
End 2
IsEmpty 2
Start 2
Methods 2
Contains(Range<T>) 3
Contains(T) 3
Equals(Object) 3
Equals(Range<T>) 4
GetHashCode() 4
Intersect(Range<T>, Range<T>) 4
IntersectsWith(Range<T>) 5
ToString() 5
Union(Range<T>, Range<T>) 6
Operators 6
Equality(Range<T>, Range<T>) 6
Inequality(Range<T>, Range<T>) 6