System.TimeSpan Structure

```
[ILAsm]
.class public sequential sealed serializable TimeSpan extends
System.ValueType implements System.IComparable,
System.IComparable`1<valuetype System.TimeSpan>,
System.IEquatable`1<valuetype System.TimeSpan>

[C#]
public struct TimeSpan: IComparable, IComparable<TimeSpan>,
IEquatable<TimeSpan>
```

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.IComparable**
- **System.IComparable<System.TimeSpan>**
- **System.IEquatable<System.TimeSpan>**

Summary

Represents an interval of time.

Inherits From: System.ValueType

Library: BCL

Thread Safety: All public static members of this type are safe for multithreaded operations. No instance members are guaranteed to be thread safe.

Description

The `System.TimeSpan` structure represents an interval of time with values ranging from `System.Int64.MinValue` to `System.Int64.MaxValue` 100-nanosecond *ticks*.

[*Note:* The value of a `System.TimeSpan` is represented internally as a number of 100-nanosecond ticks. Both the specification of a number of ticks and the value of a `System.TimeSpan` can be positive or negative.

A `System.TimeSpan` can be represented as a string in the format "[-]d.hh:mm:ss.ff" where "-" is an optional sign for negative `System.TimeSpan` values, the "d" component is days, "hh" is hours, "mm" is minutes, "ss" is seconds, and "ff" is fractions of a second. For example, a `System.TimeSpan` initialized with 10^{13} ticks would be represented as "11.13:46:40", which is 11 days, 13 hours, 46 minutes, and 40 seconds.

Due to a varying number of days in months and years, the longest unit of time that is used by this structure is the day.

]

TimeSpan(System.Int64) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(int64 ticks)

[C#]
public TimeSpan(long ticks)
```

Summary

Constructs and initializes a new `System.TimeSpan` with the specified number of ticks.

Parameters

Parameter	Description
ticks	A `System.Int64` that specifies the number of ticks with which to initialize the new `System.TimeSpan`.

TimeSpan(System.Int32, System.Int32, System.Int32) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(int32 hours,
int32 minutes, int32 seconds)

[C#]
public TimeSpan(int hours, int minutes, int seconds)
```

Summary

Constructs and initializes a new `System.TimeSpan` with the specified numbers of hours, minutes, and seconds.

Parameters

Parameter	Description
hours	A `System.Int32` that specifies the number of hours with which to initialize the new `System.TimeSpan`.
minutes	A `System.Int32` that specifies the number of minutes with which to initialize the new `System.TimeSpan`.
seconds	A `System.Int32` that specifies the number of seconds with which to initialize the new `System.TimeSpan`.

Description

The specified *hours*, *minutes*, and *seconds* are converted to ticks, and that value is used to initialize the new `System.TimeSpan`.

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	The parameters specify a `System.TimeSpan` value less than `System.TimeSpan.MinValue` or greater than `System.TimeSpan.MaxValue`.

TimeSpan(System.Int32, System.Int32, System.Int32, System.Int32, System.Int32) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(int32 days,
int32 hours, int32 minutes, int32 seconds, int32 milliseconds)

[C#]
public TimeSpan(int days, int hours, int minutes, int seconds, int
milliseconds)
```

Summary

Constructs and initializes a new `System.TimeSpan` with the specified numbers of days, hours, minutes, seconds, and milliseconds.

Parameters

Parameter	Description
days	A `System.Int32` that specifies the number of days with which to initialize the new `System.TimeSpan`.
hours	A `System.Int32` that specifies the number of hours with which to initialize the new `System.TimeSpan`.
minutes	A `System.Int32` that specifies the number of minutes with which to initialize the new `System.TimeSpan`.
seconds	A `System.Int32` that specifies the number of seconds with which to initialize the new `System.TimeSpan`.
milliseconds	A `System.Int32` that specifies the number of milliseconds with which to initialize the new `System.TimeSpan`.

Description

The specified *days*, *hours*, *minutes*, *seconds*, and *milliseconds* are converted to ticks, and that value is used to initialize the new `System.TimeSpan`.

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	The parameters specify a `System.TimeSpan` value less than `System.TimeSpan.MinValue` or greater than `System.TimeSpan.MaxValue`.

TimeSpan(System.Int32, System.Int32, System.Int32, System.Int32) Constructor

```
[ILAsm]
public rtspecialname specialname instance void .ctor(int32 days,
int32 hours, int32 minutes, int32 seconds)

[C#]
public TimeSpan(int days, int hours, int minutes, int seconds)
```

Summary

Constructs and initializes a new `System.TimeSpan` with the specified numbers of days, hours, minutes, and seconds.

Parameters

Parameter	Description
days	A `System.Int32` that specifies the number of days with which to initialize the new `System.TimeSpan`.
hours	A `System.Int32` that specifies the number of hours with which to initialize the new `System.TimeSpan`.
minutes	A `System.Int32` that specifies the number of minutes with which to initialize the new `System.TimeSpan`.
seconds	A `System.Int32` that specifies the number of seconds with which to initialize the new `System.TimeSpan`.

Description

The specified *days*, *hours*, *minutes*, and *seconds* are converted to ticks, and that value is used to initialize the new `System.TimeSpan`.

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	The parameters specify a `System.TimeSpan` value less than `System.TimeSpan.MinValue` or greater than `System.TimeSpan.MaxValue`.

TimeSpan.MaxValue Field

[ILAsm]
```
.field public static initOnly valuetype System.TimeSpan MaxValue
```

[C#]
```
public static readonly TimeSpan MaxValue
```

Summary

Returns a `System.TimeSpan` whose value is the maximum value for the `System.TimeSpan` type.

Description

This field is read-only.

This field is a `System.TimeSpan` containing `System.Int64.MaxValue` ticks, the maximum `System.TimeSpan` value. The string representation of this value is positive 10675199.02:48:05.4775807.

TimeSpan.MinValue Field

[ILAsm]
```
.field public static initOnly valuetype System.TimeSpan MinValue
```

[C#]
```
public static readonly TimeSpan MinValue
```

Summary

Returns a `System.TimeSpan` whose value is the minimum value for the `System.TimeSpan` type.

Description

This field is read-only.

This field is a `System.TimeSpan` containing `System.Int64.MinValue` ticks, the minimum `System.TimeSpan` value. The string representation of this value is negative 10675199.02:48:05.4775808.

TimeSpan.TicksPerDay Field

[ILAsm]

```
.field public static literal int64 TicksPerDay = 864000000000
```

[C#]

```
public const long TicksPerDay = 864000000000
```

Summary

Represents the number of ticks in 1 day.

Description

The value of this constant is 864 billion ($8.64x10^{11}$).

TimeSpan.TicksPerHour Field

[ILAsm]
```
.field public static literal int64 TicksPerHour = 36000000000
```

[C#]
```
public const long TicksPerHour = 36000000000
```

Summary

Represents the number of ticks in 1 hour.

Description

The value of this constant is 36 billion ($3.6x10^{10}$).

TimeSpan.TicksPerMillisecond Field

[ILAsm]
```
.field public static literal int64 TicksPerMillisecond = 10000
```

[C#]
```
public const long TicksPerMillisecond = 10000
```

Summary

Represents the number of ticks in 1 millisecond.

Description

The value of this constant is 10 thousand (10^4).

TimeSpan.TicksPerMinute Field

[ILAsm]
```
.field public static literal int64 TicksPerMinute = 600000000
```

[C#]
```
public const long TicksPerMinute = 600000000
```

Summary

Represents the number of ticks in 1 minute.

Description

The value of this constant is 600 million ($6x10^8$).

TimeSpan.TicksPerSecond Field

[ILAsm]
```
.field public static literal int64 TicksPerSecond = 10000000
```

[C#]
```
public const long TicksPerSecond = 10000000
```

Summary

Represents the number of ticks in 1 second.

Description

The value of this constant is 10 million (10^7).

TimeSpan.Zero Field

[ILAsm]
```
.field public static initOnly valuetype System.TimeSpan Zero
```

[C#]
```
public static readonly TimeSpan Zero
```

Summary

Returns a `System.TimeSpan` whose value is 0.

Description

This field is read-only.

This field is a `System.TimeSpan` whose value is 0 ticks. [*Note:* This provides a convenient source for 0 in `System.TimeSpan` calculations.]

TimeSpan.Add(System.TimeSpan) Method

[ILAsm]
```
.method public hidebysig instance valuetype System.TimeSpan
Add(valuetype System.TimeSpan ts)
```

[C#]
```
public TimeSpan Add(TimeSpan ts)
```

Summary

Adds the specified `System.TimeSpan` to the current instance.

Parameters

Parameter	**Description**
ts	A `System.TimeSpan` instance to add to the current instance.

Return Value

A `System.TimeSpan` that represents the value of the current instance plus the value of *ts*.

Exceptions

Exception	**Condition**
System.OverflowException	The sum of the value of the current instance and the value of *ts* is less than `System.TimeSpan.MinValue` or greater than `System.TimeSpan.MaxValue`.

Example

This example demonstrates the `System.TimeSpan.Add` method.

[C#]

```
using System;
public class TimeSpanAddExample {
   public static void Main() {
      TimeSpan ts = new TimeSpan(Int32.MaxValue);
      Console.WriteLine("The value of the timespan 'ts' is {0}", ts);
      Console.WriteLine("ts.Add(ts) = {0}", ts.Add(ts));
   }
}
```

The output is

```
The value of the timespan 'ts' is 00:03:34.7483647


ts.Add(ts) = 00:07:09.4967294
```

TimeSpan.Compare(System.TimeSpan, System.TimeSpan) Method

```
[ILAsm]
.method public hidebysig static int32 Compare(valuetype
System.TimeSpan t1, valuetype System.TimeSpan t2)

[C#]
public static int Compare(TimeSpan t1, TimeSpan t2)
```

Summary

Returns the sort order of two `System.TimeSpan` structures.

Parameters

Parameter	**Description**
t1	The first `System.TimeSpan` to compare.
t2	The second `System.TimeSpan` to compare.

Return Value

The return value is a negative number, zero, or a positive number reflecting the sort order of *t1* as compared to *t2*. For non-zero return values, the exact value returned by this method is unspecified. The following table defines the return value:

Value	**Condition**
Any negative number	*t1* < *t2*.
Zero	*t1* == *t2*.
Any positive number	*t1* > *t2*.

TimeSpan.CompareTo(System.Object) Method

[ILAsm]

```
.method public final hidebysig virtual int32 CompareTo(object value)
```

[C#]

```
public int CompareTo(object value)
```

Summary

Returns the sort order of the current instance compared to the specified `System.Object`.

Parameters

Parameter	Description
value	The `System.Object` to compare to the current instance.

Return Value

The return value is a negative number, zero, or a positive number reflecting the sort order of the current instance as compared to *value*. For non-zero return values, the exact value returned by this method is unspecified. The following table defines the return value:

Value	Condition
Any negative number	Current instance < *value*.
Zero	Current instance == *value*.
Any positive number	Current instance > *value*, or *value* is a null reference.

Description

[*Note:* This method is implemented to support the `System.IComparable` interface.]

Exceptions

Exception	**Condition**
System.ArgumentException	*value* is not a `System.TimeSpan` and is not a null reference.

TimeSpan.CompareTo(System.TimeSpan) Method

```
[ILAsm]
.method public final hidebysig virtual int32 CompareTo(valuetype
System.TimeSpan value)

[C#]
public int CompareTo(TimeSpan value)
```

Summary

Returns the sort order of the current instance compared to the specified `System.TimeSpan`.

Parameters

Parameter	Description
value	The `System.TimeSpan` to compare to the current instance.

Return Value

The return value is a negative number, zero, or a positive number reflecting the sort order of the current instance as compared to *value*. For non-zero return values, the exact value returned by this method is unspecified. The following table defines the return value:

Value	Condition
Any negative number	Current instance < *value*.
Zero	Current instance == *value*.
Any positive number	Current instance > *value*.

Description

[*Note:* This method is implemented to support the `System.IComparable<System.TimeSpan>` interface.]

TimeSpan.Duration() Method

[ILAsm]
`.method public hidebysig instance valuetype System.TimeSpan Duration()`

[C#]
`public TimeSpan Duration()`

Summary

Returns a `System.TimeSpan` whose value is the absolute value of the current instance.

Return Value

A `System.TimeSpan` whose value is the absolute value of the current instance.

Exceptions

Exception	Condition
System.OverflowException	The value of the current instance is `System.TimeSpan.MinValue`.

Example

The following example demonstrates the `System.TimeSpan.Duration` method.

[C#]

```
using System;
public class TimeSpanDurationExample {
   public static void Main() {
      TimeSpan ts = new TimeSpan(Int32.MinValue);
      Console.Write("The absolute value of TimeSpan {0} ", ts);
      Console.WriteLine("is {0}", ts.Duration());
   }
}
```

The output is

```
The absolute value of TimeSpan -00:03:34.7483648 is 00:03:34.7483648
```

TimeSpan.Equals(System.TimeSpan, System.TimeSpan) Method

[ILAsm]

```
.method public hidebysig static bool Equals(valuetype
System.TimeSpan t1, valuetype System.TimeSpan t2)
```

[C#]

```
public static bool Equals(TimeSpan t1, TimeSpan t2)
```

Summary

Determines whether two `System.TimeSpan` structures represent the same type and value.

Parameters

Parameter	Description
t1	The first instance of `System.TimeSpan` to compare for equality.
t2	The second instance of `System.TimeSpan` to compare for equality.

Return Value

`true` if *t1* and *t2* represent the same value; otherwise, `false`.

TimeSpan.Equals(System.Object) Method

[ILAsm]
```
.method public hidebysig virtual bool Equals(object value)
```

[C#]
```
public override bool Equals(object value)
```

Summary

Determines whether the current instance and the specified `System.Object` represent the same type and value.

Parameters

Parameter	Description
value	The `System.Object` to compare to the current instance.

Return Value

`true` if *value* represents the same type and value as the current instance. If *value* is a null reference or is not a `System.TimeSpan`, returns `false`.

Description

[*Note:* This method overrides `System.Object.Equals`.]

TimeSpan.Equals(System.TimeSpan) Method

```
[ILAsm]
.method public hidebysig virtual bool Equals(valuetype
System.TimeSpan obj)

[C#]
public override bool Equals(TimeSpan obj)
```

Summary

Determines whether the current instance and the specified `System.TimeSpan` represent the same value.

Parameters

Parameter	Description
value	The `System.TimeSpan` to compare to the current instance.

Return Value

`true` if *value* represents the same value as the current instance; otherwise, `false`.

Description

[*Note:* This method is implemented to support the `System.IEquatable<System.TimeSpan>` interface.]

The following member must be implemented if the ExtendedNumerics library is present in the implementation.

TimeSpan.FromDays(System.Double) Method

```
[ILAsm]
.method public hidebysig static valuetype System.TimeSpan
FromDays(float64 value)

[C#]
public static TimeSpan FromDays(double value)
```

Summary

Returns a `System.TimeSpan` that represents the specified number of days where the specification is accurate to the nearest millisecond.

Parameters

Parameter	Description
value	A `System.Double` that specifies the number of days with which the new `System.TimeSpan` is initialized.

Return Value

A `System.TimeSpan` that represents *value*.

Description

value will only be considered accurate to the nearest millisecond.

If *value* is `System.Double.PositiveInfinity`, a `System.TimeSpan` with the value `System.TimeSpan.MaxValue` is returned. If *value* is `System.Double.NegativeInfinity`, a `System.TimeSpan` with the value `System.TimeSpan.MinValue` is returned.

Exceptions

Exception	Condition
System.OverflowException	The `System.TimeSpan` represented by *value* is greater than `System.TimeSpan.MaxValue` or less than `System.TimeSpan.MinValue`.
System.ArgumentException	*value* is equal to `System.Double.NaN`.

The following member must be implemented if the ExtendedNumerics library is present in the implementation.

TimeSpan.FromHours(System.Double) Method

```
[ILAsm]
.method public hidebysig static valuetype System.TimeSpan
FromHours(float64 value)

[C#]
public static TimeSpan FromHours(double value)
```

Summary

Returns a `System.TimeSpan` that represents the specified number of hours where the specification is accurate to the nearest millisecond.

Parameters

Parameter	Description
value	A `System.Double` that specifies the number of hours with which the new `System.TimeSpan` is initialized.

Return Value

A `System.TimeSpan` that represents *value*.

Description

value will only be considered accurate to the nearest millisecond.

If *value* is `System.Double.PositiveInfinity`, a `System.TimeSpan` with the value `System.TimeSpan.MaxValue` is returned. If *value* is `System.Double.NegativeInfinity`, a `System.TimeSpan` with the value `System.TimeSpan.MinValue` is returned.

Exceptions

Exception	Condition
System.OverflowException	The `System.TimeSpan` represented by *value* is greater than `System.TimeSpan.MaxValue` or less than `System.TimeSpan.MinValue`.

System.ArgumentException	*value* is equal to `System.Double.NaN`.

The following member must be implemented if the ExtendedNumerics library is present in the implementation.

TimeSpan.FromMilliseconds(System.Double) Method

```
[ILAsm]
.method public hidebysig static valuetype System.TimeSpan
FromMilliseconds(float64 value)

[C#]
public static TimeSpan FromMilliseconds(double value)
```

Summary

Returns a `System.TimeSpan` that represents the specified number of milliseconds where the specification is accurate to the nearest millisecond.

Parameters

Parameter	Description
value	A `System.Double` that specifies the number of milliseconds with which the new `System.TimeSpan` is initialized.

Return Value

A `System.TimeSpan` that represents *value*.

Description

value will only be considered accurate to the nearest millisecond.

If *value* is `System.Double.PositiveInfinity`, a `System.TimeSpan` with the value `System.TimeSpan.MaxValue` is returned. If *value* is `System.Double.NegativeInfinity`, a `System.TimeSpan` with the value `System.TimeSpan.MinValue` is returned.

Exceptions

Exception	Condition
System.OverflowException	The `System.TimeSpan` represented by *value* is greater than `System.TimeSpan.MaxValue` or less than `System.TimeSpan.MinValue`.

System.ArgumentException	*value* is equal to `System.Double.NaN`.

The following member must be implemented if the ExtendedNumerics library is present in the implementation.

TimeSpan.FromMinutes(System.Double) Method

[ILAsm]
```
.method public hidebysig static valuetype System.TimeSpan
FromMinutes(float64 value)
```

[C#]
```
public static TimeSpan FromMinutes(double value)
```

Summary

Returns a `System.TimeSpan` that represents the specified number of minutes where the specification is accurate to the nearest millisecond.

Parameters

Parameter	Description
value	A `System.Double` that specifies the number of minutes with which the new `System.TimeSpan` is initialized.

Return Value

A `System.TimeSpan` that represents *value*.

Description

value will only be considered accurate to the nearest millisecond.

If *value* is `System.Double.PositiveInfinity`, a `System.TimeSpan` with the value `System.TimeSpan.MaxValue` is returned. If *value* is `System.Double.NegativeInfinity`, a `System.TimeSpan` with the value `System.TimeSpan.MinValue` is returned.

Exceptions

Exception	Condition
System.OverflowException	The `System.TimeSpan` represented by *value* is greater than `System.TimeSpan.MaxValue` or less than `System.TimeSpan.MinValue`.

System.ArgumentException	*value* is equal to `System.Double.NaN`.

The following member must be implemented if the ExtendedNumerics library is present in the implementation.

TimeSpan.FromSeconds(System.Double) Method

```
[ILAsm]
.method public hidebysig static valuetype System.TimeSpan
FromSeconds(float64 value)

[C#]
public static TimeSpan FromSeconds(double value)
```

Summary

Returns a `System.TimeSpan` that represents the specified number of seconds where the specification is accurate to the nearest millisecond.

Parameters

Parameter	Description
value	A `System.Double` that specifies the number of seconds with which the new `System.TimeSpan` is initialized.

Return Value

A `System.TimeSpan` that represents *value*.

Description

value will only be considered accurate to the nearest millisecond.

If *value* is `System.Double.PositiveInfinity`, a `System.TimeSpan` with the value `System.TimeSpan.MaxValue` is returned. If *value* is `System.Double.NegativeInfinity`, a `System.TimeSpan` with the value `System.TimeSpan.MinValue` is returned.

Exceptions

Exception	Condition
System.OverflowException	The `System.TimeSpan` represented by *value* is greater than `System.TimeSpan.MaxValue` or less than `System.TimeSpan.MinValue`.

System.ArgumentException	*value* is equal to `System.Double.NaN`.

TimeSpan.FromTicks(System.Int64) Method

```
[ILAsm]
.method public hidebysig static valuetype System.TimeSpan
FromTicks(int64 value)

[C#]
public static TimeSpan FromTicks(long value)
```

Summary

Returns a `System.TimeSpan` that represents the specified number of ticks.

Parameters

Parameter	Description
value	A `System.Int64` that specifies the number of ticks with which the new `System.TimeSpan` is initialized.

Return Value

A `System.TimeSpan` with a value of *value*.

Description

This method is equivalent to the `System.TimeSpan`(`System.Int64`) constructor.

TimeSpan.GetHashCode() Method

[ILAsm]
```
.method public hidebysig virtual int32 GetHashCode()
```

[C#]
```
public override int GetHashCode()
```

Summary

Generates a hash code for the current instance.

Return Value

A `System.Int32` value containing a hash code for the current instance.

Description

The algorithm used to generate the hash code is unspecified.

[*Note:* This method overrides `System.Object.GetHashCode`.]

TimeSpan.Negate() Method

[ILAsm]
```
.method public hidebysig instance valuetype System.TimeSpan Negate()
```

[C#]
```
public TimeSpan Negate()
```

Summary

Returns a `System.TimeSpan` with the same absolute value but opposite sign as the current instance.

Return Value

A `System.TimeSpan` with the same absolute value but with the opposite sign as the current instance.

Exceptions

Exception	Condition
System.OverflowException	The value of the current instance is `System.TimeSpan.MinValue`.

TimeSpan.op_Addition(System.TimeSpan, System.TimeSpan) Method

```
[ILAsm]
.method public hidebysig static specialname valuetype
System.TimeSpan op_Addition(valuetype System.TimeSpan t1, valuetype
System.TimeSpan t2)

[C#]
public static TimeSpan operator +(TimeSpan t1, TimeSpan t2)
```

Summary

Adds the values of two `System.TimeSpan` instances.

Parameters

Parameter	**Description**
t1	The first `System.TimeSpan`.
t2	The second `System.TimeSpan`.

Return Value

A `System.TimeSpan` whose value is the sum of the values of *t1* and *t2*.

Exceptions

Exception	**Condition**
System.OverflowException	The sum of *t1* and *t2* is less than `System.TimeSpan.MinValue` or greater than `System.TimeSpan.MaxValue`.

TimeSpan.op_Equality(System.TimeSpan, System.TimeSpan) Method

```
[ILAsm]
.method public hidebysig static specialname bool
op_Equality(valuetype System.TimeSpan t1, valuetype System.TimeSpan
t2)

[C#]
public static bool operator ==(TimeSpan t1, TimeSpan t2)
```

Summary

Determines whether the value of one `System.TimeSpan` is equal to the value of another `System.TimeSpan`.

Parameters

Parameter	Description
t1	The first `System.TimeSpan`
t2	The second `System.TimeSpan`

Return Value

`true` if the values of *t1* and *t2* are equal; otherwise, `false`.

TimeSpan.op_GreaterThan(System.TimeSpan, System.TimeSpan) Method

[ILAsm]
```
.method public hidebysig static specialname bool
op_GreaterThan(valuetype System.TimeSpan t1, valuetype
System.TimeSpan t2)
```

[C#]
```
public static bool operator >(TimeSpan t1, TimeSpan t2)
```

Summary

Determines whether the value one `System.TimeSpan` is greater than the value of another `System.TimeSpan`.

Parameters

Parameter	Description
t1	The first `System.TimeSpan`.
t2	The second `System.TimeSpan`.

Return Value

`true` if the value of *t1* is greater than the value of *t2*; otherwise, `false`.

TimeSpan.op_GreaterThanOrEqual(System.TimeSpan, System.TimeSpan) Method

[ILAsm]
```
.method public hidebysig static specialname bool
op_GreaterThanOrEqual(valuetype System.TimeSpan t1, valuetype
System.TimeSpan t2)
```

[C#]
```
public static bool operator >=(TimeSpan t1, TimeSpan t2)
```

Summary

Determines whether the value of one `System.TimeSpan` is greater than or equal to the value of another `System.TimeSpan`.

Parameters

Parameter	Description
t1	The first `System.TimeSpan`.
t2	The second `System.TimeSpan`.

Return Value

`true` if the value of *t1* is greater than or equal to the value of *t2*; otherwise, `false`.

TimeSpan.op_Inequality(System.TimeSpan, System.TimeSpan) Method

[ILAsm]
```
.method public hidebysig static specialname bool
op_Inequality(valuetype System.TimeSpan t1, valuetype
System.TimeSpan t2)
```

[C#]
```
public static bool operator !=(TimeSpan t1, TimeSpan t2)
```

Summary

Determines whether the value of one `System.TimeSpan` is unequal to the value of another `System.TimeSpan`.

Parameters

Parameter	Description
t1	The first `System.TimeSpan`.
t2	The second `System.TimeSpan`.

Return Value

`true` if the values of *t1* and *t2* are unequal; otherwise, `false`.

TimeSpan.op_LessThan(System.TimeSpan, System.TimeSpan) Method

[ILAsm]
```
.method public hidebysig static specialname bool
op_LessThan(valuetype System.TimeSpan t1, valuetype System.TimeSpan
t2)
```

[C#]
```
public static bool operator <(TimeSpan t1, TimeSpan t2)
```

Summary

Determines whether the value of one `System.TimeSpan` is less than the value of another `System.TimeSpan`.

Parameters

Parameter	**Description**
t1	The first `System.TimeSpan`.
t2	The second `System.TimeSpan`.

Return Value

`true` if the value of *t1* is less than the value of *t2*; otherwise, `false`.

TimeSpan.op_LessThanOrEqual(System.TimeSpan, System.TimeSpan) Method

[ILAsm]
```
.method public hidebysig static specialname bool
op_LessThanOrEqual(valuetype System.TimeSpan t1, valuetype
System.TimeSpan t2)
```

[C#]
```
public static bool operator <=(TimeSpan t1, TimeSpan t2)
```

Summary

Determines whether the value of one `System.TimeSpan` is less than or equal to the value of another `System.TimeSpan`.

Parameters

Parameter	Description
t1	The first `System.TimeSpan`.
t2	The second `System.TimeSpan`.

Return Value

`true` if the value of *t1* is less than or equal to the value of *t2*; otherwise, `false`.

TimeSpan.op_Subtraction(System.TimeSpan, System.TimeSpan) Method

[ILAsm]
```
.method public hidebysig static specialname valuetype
System.TimeSpan op_Subtraction(valuetype System.TimeSpan t1,
valuetype System.TimeSpan t2)
```

[C#]
```
public static TimeSpan operator -(TimeSpan t1, TimeSpan t2)
```

Summary

Subtracts the value of one `System.TimeSpan` from the value of another `System.TimeSpan`.

Parameters

Parameter	Description
t1	The first `System.TimeSpan`.
t2	The second `System.TimeSpan`.

Return Value

A `System.TimeSpan` whose value is the result of the value of *t1* minus the value of *t2*.

Exceptions

Exception	Condition
System.OverflowException	The value of *t2* subtracted from *t1* is less than `System.TimeSpan.MinValue` or greater than `System.TimeSpan.MaxValue`.

TimeSpan.op_UnaryNegation(System.TimeSpan) Method

[ILAsm]
```
.method public hidebysig static specialname valuetype
System.TimeSpan op_UnaryNegation(valuetype System.TimeSpan t)
```

[C#]
```
public static TimeSpan operator -(TimeSpan t)
```

Summary

Returns a `System.TimeSpan` whose value is the negated value of a specified `System.TimeSpan`.

Parameters

Parameter	Description
t	A `System.TimeSpan` whose value will be negated.

Return Value

A `System.TimeSpan` with the same absolute value but the opposite sign as *t*.

Exceptions

Exception	Condition
System.OverflowException	*t* equals `System.TimeSpan.MinValue`.

TimeSpan.op_UnaryPlus(System.TimeSpan) Method

[ILAsm]
```
.method public hidebysig static specialname valuetype
System.TimeSpan op_UnaryPlus(valuetype System.TimeSpan t)
```

[C#]
```
public static TimeSpan operator +(TimeSpan t)
```

Summary

Returns the specified instance of `System.TimeSpan`.

Parameters

Parameter	**Description**
t	A `System.TimeSpan`.

Return Value

`System.TimeSpan`*t*.

Description

This method returns `System.TimeSpan`*t*.

TimeSpan.Parse(System.String) Method

```
[ILAsm]
.method public hidebysig static valuetype System.TimeSpan
Parse(string s)

[C#]
public static TimeSpan Parse(string s)
```

Summary

Returns the specified `System.String` converted to a `System.TimeSpan` value.

Parameters

Parameter	Description
s	A `System.String` containing the value to convert. *s* contains a time interval in the following form: [ws][-][d.]hh:mm:ss[.ff][ws] Items in square brackets ('[' and']') are optional. Colons and periods (':' and'.') are literal characters. For details on the remaining symbols, see the description section.

Return Value

The `System.TimeSpan` value obtained from *s*.

Description

The symbols used in the parameter description for *s* are as follows:

Item	Description
ws	White space (zero or more space and/or tab characters).
"-"	Minus sign, indicating a negative time interval.
"d"	Days.
"hh"	Hours, ranging from 0 to 23 inclusive.
"mm"	Minutes, ranging from 0 to 59 inclusive.
"ss"	Seconds, ranging from 0 to 59 inclusive.
"ff"	Fractional seconds, from 1 to 7 decimal digits inclusive.

Exceptions

Exception	Condition
System.ArgumentNullException	*s* is a null reference.
System.FormatException	*s* is in an invalid format.
System.OverflowException	*s* represents a number greater than `System.TimeSpan.MaxValue` or less than `System.TimeSpan.MinValue`. -or- At least one of the hours, minutes, or seconds components is outside its valid range.

Example

This example demonstrates parsing a string to obtain a `System.TimeSpan`.

[C#]

```
using System;
public class TimeSpanParseExample {
   public static void Main() {
      String str = "    -5.12:34:56.789      ";
      TimeSpan ts = TimeSpan.Parse(str);
      Console.WriteLine(@"The string ""{0}""", str);
      Console.WriteLine("parses to TimeSpan {0}", ts);
   }
}
```

The output is

```
The string "    -5.12:34:56.789      "
parses to TimeSpan -5.12:34:56.7890000
```

TimeSpan.Subtract(System.TimeSpan) Method

```
[ILAsm]
.method public hidebysig instance valuetype System.TimeSpan
Subtract(valuetype System.TimeSpan ts)

[C#]
public TimeSpan Subtract(TimeSpan ts)
```

Summary

Subtracts the value of the specified `System.TimeSpan` from the value of the current instance.

Parameters

Parameter	Description
ts	A `System.TimeSpan` whose value to subtract from the value of the current instance.

Return Value

A `System.TimeSpan` whose value is equal to the value of the current instance minus the value of *ts*.

Exceptions

Exception	Condition
System.OverflowException	The difference between the value of the current instance and *ts* is less than `System.TimeSpan.MinValue` or greater than `System.TimeSpan.MaxValue`.

TimeSpan.ToString() Method

[ILAsm]
```
.method public hidebysig virtual string ToString()
```

[C#]
```
public override string ToString()
```

Summary

Returns a `System.String` representation of the value of the current instance.

Return Value

A `System.String` representation of the current instance formatted as follows:

[-][d.]hh:mm:ss[.ff]

Items in square brackets ('[' and ']') are included provisionally: '-' is included if and only if the current instance is negative; "d." and ".ff" are included if and only if those components are non-zero. Colons and periods (':' and '.') are literal characters. Other components are as follows.

Component	Description
"-"	Minus sign, indicating a negative time interval.
"d"	Days.
"hh"	Hours, ranging from 0 to 23 inclusive.
"mm"	Minutes, ranging from 0 to 59 inclusive.
"ss"	Seconds, ranging from 0 to 59 inclusive.
"ff"	Fractional seconds.

Description

[*Note:* This method overrides `System.Object.ToString`.]

Example

This example demonstrates the `System.TimeSpan.ToString` method.

[C#]

```
using System;
public class TimeSpanToStringExample {
   public static void Main() {
      TimeSpan tsOne = new TimeSpan(1, 23, 45, 54, 321);
      TimeSpan tsTwo = new TimeSpan(0, 23, 45, 54, 0);
      Console.Write("TimeSpan one, with d. and.ff: ");
      Console.WriteLine("{0}", tsOne.ToString());
      Console.Write("TimeSpan two, without d. and.ff: ");
      Console.WriteLine("{0}", tsTwo.ToString());
   }
}
```

The output is

```
TimeSpan one, with d. and.ff: 1.23:45:54.3210000


TimeSpan two, without d. and.ff: 23:45:54
```

TimeSpan.Days Property

[ILAsm]
```
.property int32 Days { public hidebysig specialname instance int32
get_Days() }
```

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

Summary

Gets the number days represented by the current instance.

Property Value

A `System.Int32` represents the days component of the current instance. [*Note:* See `System.TimeSpan.ToString` for a more detailed description of the days component.]

Description

This property is read-only.

Example

This example demonstrates using the `System.TimeSpan.Days` property.

[C#]

```
using System;
public class TimeSpanPropertiesExampleOne {
   public static void Main() {
      TimeSpan ts = new TimeSpan((Int64)10e12+3456789);
      Console.WriteLine(ts.ToString());
      Console.WriteLine("Days: {0}", ts.Days );
   }
}
```

The output is

```
11.13:46:40.3456789


Days: 11
```

TimeSpan.Hours Property

[ILAsm]

```
.property int32 Hours { public hidebysig specialname instance int32
get_Hours() }
```

[C#]

```
public int Hours { get; }
```

Summary

Gets the number of hours represented by the current instance.

Property Value

A `System.Int32` between 0 and 23 inclusive, that represents the hours component of the current instance. [*Note:* See `System.TimeSpan.ToString` for a more detailed description of the hours component.]

Description

This property is read-only.

Example

This example demonstrates using the `System.TimeSpan.Hours` property.

[C#]

```
using System;
public class TimeSpanPropertiesExampleOne {
   public static void Main() {
      TimeSpan ts = new TimeSpan((Int64)10e12+3456789);
      Console.WriteLine(ts.ToString());
      Console.WriteLine("Hours: {0}", ts.Hours );
   }
}
```

The output is

```
11.13:46:40.3456789


Hours: 13
```

TimeSpan.Milliseconds Property

[ILAsm]

```
.property int32 Milliseconds { public hidebysig specialname instance
int32 get_Milliseconds() }
```

[C#]

```
public int Milliseconds { get; }
```

Summary

Gets the number of milliseconds represented by the current instance.

Property Value

A `System.Int32` between 0 and 999 inclusive, that represents the fractional seconds component of the current instance converted to milliseconds. [*Note:* See `System.TimeSpan.ToString` for a more detailed description of the fractional seconds component.]

Description

This property is read-only.

Example

This example demonstrates using the `System.TimeSpan.Milliseconds` property.

[C#]

```
using System;
public class TimeSpanPropertiesExampleOne {
   public static void Main() {
      TimeSpan ts = new TimeSpan((Int64)10e12+3456789);
      Console.WriteLine(ts.ToString());
      Console.WriteLine("Milliseconds: {0}", ts.Milliseconds );
   }
}
```

The output is

```
11.13:46:40.3456789


Milliseconds: 345
```

TimeSpan.Minutes Property

[ILAsm]

```
.property int32 Minutes { public hidebysig specialname instance
int32 get_Minutes() }
```

[C#]

```
public int Minutes { get; }
```

Summary

Gets the number of minutes represented by the current instance.

Property Value

A `System.Int32` between 0 and 59 inclusive, that represents the minutes component of the current instance. [*Note:* See `System.TimeSpan.ToString` for a more detailed description of the minutes component.]

Description

This property is read-only.

Example

This example demonstrates using the `System.TimeSpan.Minutes` property.

[C#]

```
using System;
public class TimeSpanPropertiesExampleOne {
   public static void Main() {
      TimeSpan ts = new TimeSpan((Int64)10e12+3456789);
      Console.WriteLine(ts.ToString());
      Console.WriteLine("Minutes: {0}", ts.Minutes );
   }
}
```

The output is

```
11.13:46:40.3456789


Minutes: 46
```

TimeSpan.Seconds Property

[ILAsm]
```
.property int32 Seconds { public hidebysig specialname instance
int32 get_Seconds() }
```

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

Summary

Gets the number of seconds represented by the current instance.

Property Value

A `System.Int32` between 0 and 59 inclusive, that represents the seconds component of the current instance. [*Note:* See `System.TimeSpan.ToString` for a more detailed description of the seconds component.]

Description

This property is read-only.

Example

This example demonstrates using the `System.TimeSpan.Seconds` property.

[C#]

```
using System;
public class TimeSpanPropertiesExampleOne {
   public static void Main() {
      TimeSpan ts = new TimeSpan((Int64)10e12+3456789);
      Console.WriteLine(ts.ToString());
      Console.WriteLine("Seconds: {0}", ts.Seconds );
   }
}
```

The output is

```
11.13:46:40.3456789


Seconds: 40
```

TimeSpan.Ticks Property

[ILAsm]

```
.property int64 Ticks { public hidebysig specialname instance int64
get_Ticks() }
```

[C#]

```
public long Ticks { get; }
```

Summary

Gets the number of ticks represented by the current instance.

Property Value

A `System.Int64` specifying the number of ticks represented by the current instance.

Description

This property is read-only.

The following member must be implemented if the ExtendedNumerics library is present in the implementation.

TimeSpan.TotalDays Property

[ILAsm]

```
.property float64 TotalDays { public hidebysig specialname instance
float64 get_TotalDays() }
```

[C#]

```
public double TotalDays { get; }
```

Summary

Gets the value of the current instance expressed in days.

Property Value

A `System.Double` that specifies the total number of days represented by the current instance.

Description

This property is read-only.

[*Note:* This property converts the value of the current instance from ticks to days. This number can include whole and fractional days.]

Example

This example demonstrates using the `System.TimeSpan.TotalDays` property.

[C#]

```
using System;
public class TimeSpanTotalUnitsProperties{
   public static void Main() {
      TimeSpan ts = new TimeSpan((Int64)10e12);
      Console.WriteLine(ts.ToString());
      Console.WriteLine("TotalDays: {0}", ts.TotalDays);
   }
}
```

The output is

```
11.13:46:40


TotalDays: 11.5740740740741
```

The following member must be implemented if the ExtendedNumerics library is present in the implementation.

TimeSpan.TotalHours Property

[ILAsm]
```
.property float64 TotalHours { public hidebysig specialname instance float64 get_TotalHours() }
```

[C#]
```
public double TotalHours { get; }
```

Summary

Gets the value of the current instance expressed in hours.

Property Value

A `System.Double` that specifies the total number of hours represented by the current instance.

Description

This property is read-only.

[*Note:* This property converts the value of the current instance from ticks to hours. This number can include whole and fractional hours.]

Example

This example demonstrates using the `System.TimeSpan.TotalHours` property.

[C#]

```
using System;
public class TimeSpanTotalUnitsProperties{
   public static void Main() {
      TimeSpan ts = new TimeSpan((Int64)10e12);
      Console.WriteLine(ts.ToString());
      Console.WriteLine("TotalHours: {0}", ts.TotalHours);
   }
}
```

The output is

```
11.13:46:40


TotalHours: 277.777777777778
```

The following member must be implemented if the ExtendedNumerics library is present in the implementation.

TimeSpan.TotalMilliseconds Property

[ILAsm]
```
.property float64 TotalMilliseconds { public hidebysig specialname
instance float64 get_TotalMilliseconds() }
```

[C#]
```
public double TotalMilliseconds { get; }
```

Summary

Gets the value of the current instance expressed in milliseconds.

Property Value

A `System.Double` that specifies the total number of milliseconds represented by the current instance.

Description

This property is read-only.

[*Note:* This property converts the value of the current instance from ticks to milliseconds. This number can include whole and fractional milliseconds.]

Example

This example demonstrates using the `System.TimeSpan.TotalMilliseconds` property.

[C#]

```
using System;
public class TimeSpanTotalUnitsProperties{
   public static void Main() {
      TimeSpan ts = new TimeSpan((Int64)10e12);
      Console.WriteLine(ts.ToString());
      Console.WriteLine("TotalMilliseconds: {0}",
ts.TotalMilliseconds);
   }
}
```

The output is

```
11.13:46:40
```

```
TotalMilliseconds: 1000000000
```

The following member must be implemented if the ExtendedNumerics library is present in the implementation.

TimeSpan.TotalMinutes Property

```
[ILAsm]
.property float64 TotalMinutes { public hidebysig specialname
instance float64 get_TotalMinutes() }

[C#]
public double TotalMinutes { get; }
```

Summary

Gets the value of the current instance expressed in minutes.

Property Value

A `System.Double` that specifies the total number of minutes represented by the current instance.

Description

This property is read-only.

[*Note:* This property converts the value of the current instance from ticks to minutes. This number can include whole and fractional minutes.]

Example

This example demonstrates using the `System.TimeSpan.TotalMinutes` property.

[C#]

```
using System;
public class TimeSpanTotalUnitsProperties{
   public static void Main() {
      TimeSpan ts = new TimeSpan((Int64)10e12);
      Console.WriteLine(ts.ToString());
      Console.WriteLine("TotalMinutes: {0}", ts.TotalMinutes);
   }
}
```

The output is

```
11.13:46:40


TotalMinutes: 16666.6666666667
```

The following member must be implemented if the ExtendedNumerics library is present in the implementation.

TimeSpan.TotalSeconds Property

[ILAsm]

```
.property float64 TotalSeconds { public hidebysig specialname
instance float64 get_TotalSeconds() }
```

[C#]

```
public double TotalSeconds { get; }
```

Summary

Gets the value of the current instance expressed in seconds.

Property Value

A `System.Double` that specifies the total number of seconds represented by the current instance.

Description

This property is read-only.

[*Note:* This property converts the value of the current instance from ticks to seconds. This number can include whole and fractional seconds.]

Example

This example demonstrates using the `System.TimeSpan.TotalSeconds` property.

[C#]

```
using System;
public class TimeSpanTotalUnitsProperties{
   public static void Main() {
      TimeSpan ts = new TimeSpan((Int64)10e12);
      Console.WriteLine(ts.ToString());
      Console.WriteLine("TotalSeconds: {0}", ts.TotalSeconds);
   }
}
```

The output is

```
11.13:46:40


TotalSeconds:1000000
```