

System.TimeSpan Structure

```
[ILASM]
.class public sequential sealed serializable TimeSpan
extends System.ValueType implements System.IComparable

[C#]
public struct TimeSpan: IComparable
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- *Version:* 1.0.x.x
- *Attributes:*
 - CLSCompliantAttribute(true)

Implements:

- **System.IComparable**

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 ticks would be

1 represented as "11.13:46:40", which is 11 days, 13 hours, 46
2 minutes, and 40 seconds.

3

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

6

1 TimeSpan(System.Int64) Constructor

```
2 [ILASM]  
3 public rtspecialname specialname instance void .ctor(int64  
4 ticks)  
5  
6 [C#]  
7 public TimeSpan(long ticks)
```

7 Summary

8 Constructs and initializes a new **System.TimeSpan** with the specified
9 number of ticks.

10 Parameters

11
12

Parameter	Description
<i>ticks</i>	A System.Int64 that specifies the number of ticks with which to initialize the new System.TimeSpan .

13
14
15

1 TimeSpan(System.Int32, System.Int32, 2 System.Int32) Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void .ctor(int32  
5 hours, int32 minutes, int32 seconds)  
  
6 [C#]  
7 public TimeSpan(int hours, int minutes, int seconds)
```

8 Summary

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

11 Parameters

12
13

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

14
15

15 Description

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

18 Exceptions

19
20

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

21
22
23

1 TimeSpan(System.Int32, System.Int32, 2 System.Int32, System.Int32) Constructor

```
3 [ILASM]  
4 public rtspecialname specialname instance void .ctor(int32  
5 days, int32 hours, int32 minutes, int32 seconds)  
  
6 [C#]  
7 public TimeSpan(int days, int hours, int minutes, int  
8 seconds)
```

9 Summary

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

12 Parameters

13
14

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

15

16 Description

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

19 Exceptions

20
21

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

22

23

24

1 TimeSpan(System.Int32, System.Int32, 2 System.Int32, System.Int32, 3 System.Int32) Constructor

```
4 [ILASM]  
5 public rtspecialname specialname instance void .ctor(int32  
6 days, int32 hours, int32 minutes, int32 seconds, int32  
7 milliseconds)
```

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

11 Summary

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

14 Parameters

15
16

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

17

18 Description

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

22 Exceptions

23
24

Exception	Condition
System.ArgumentOutOfRangeException	The parameters specify a System.TimeSpan value less than

1
2
3

	System.TimeSpan.MinValue or greater than System.TimeSpan.MaxValue.
--	--

1 TimeSpan.MaxValue Field

```
2 [ILASM]  
3 .field public static initOnly valuetype System.TimeSpan  
4 MaxValue  
5 [C#]  
6 public static readonly TimeSpan MaxValue
```

7 Summary

8 Returns a **System.TimeSpan** whose value is the maximum value for
9 the **System.TimeSpan** type.

10 Description

11 This field is read-only.

12
13 This field is a **System.TimeSpan** containing
14 **System.Int64.MaxValue** ticks, the maximum **System.TimeSpan**
15 value. The string representation of this value is positive
16 10675199.02:48:05.4775807.

17

1 TimeSpan.MinValue Field

```
2 [ILASM]  
3 .field public static initOnly valuetype System.TimeSpan  
4 MinValue  
5 [C#]  
6 public static readonly TimeSpan MinValue
```

7 Summary

8 Returns a **System.TimeSpan** whose value is the minimum value for
9 the **System.TimeSpan** type.

10 Description

11 This field is read-only.

12
13 This field is a **System.TimeSpan** containing
14 **System.Int64.MinValue** ticks, the minimum **System.TimeSpan**
15 value. The string representation of this value is negative
16 10675199.02:48:05.4775808.

17

1 TimeSpan.TicksPerDay Field

```
2 [ILASM]  
3 .field public static literal int64 TicksPerDay =  
4 864000000000  
5 [C#]  
6 public const long TicksPerDay = 864000000000
```

7 Summary

8 Represents the number of ticks in 1 day.

9 Description

10 The value of this constant is 864 billion (8.64×10^{11}).

11

1 TimeSpan.TicksPerHour Field

```
2 [ILASM]  
3 .field public static literal int64 TicksPerHour =  
4 36000000000  
5 [C#]  
6 public const long TicksPerHour = 36000000000
```

7 Summary

8 Represents the number of ticks in 1 hour.

9 Description

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

11

1 TimeSpan.TicksPerMillisecond Field

```
2 [ILASM]  
3 .field public static literal int64 TicksPerMillisecond =  
4 10000  
5 [C#]  
6 public const long TicksPerMillisecond = 10000
```

7 Summary

8 Represents the number of ticks in 1 millisecond.

9 Description

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

11

1 TimeSpan.TicksPerMinute Field

```
2 [ILASM]  
3 .field public static literal int64 TicksPerMinute =  
4 600000000  
5 [C#]  
6 public const long TicksPerMinute = 600000000
```

7 Summary

8 Represents the number of ticks in 1 minute.

9 Description

10 The value of this constant is 600 million (6×10^8).

11

1 TimeSpan.TicksPerSecond Field

```
2 [ILASM]  
3 .field public static literal int64 TicksPerSecond =  
4 10000000  
5 [C#]  
6 public const long TicksPerSecond = 10000000
```

7 Summary

8 Represents the number of ticks in 1 second.

9 Description

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

11

1 TimeSpan.Zero Field

```
2 [ILASM]  
3 .field public static initOnly valuetype System.TimeSpan  
4 Zero  
5 [C#]  
6 public static readonly TimeSpan Zero
```

7 Summary

8 Returns a **System.TimeSpan** whose value is 0.

9 Description

10 This field is read-only.

11
12 This field is a **System.TimeSpan** whose value is 0 ticks. [*Note:* This
13 provides a convenient source for 0 in **System.TimeSpan**
14 calculations.]

15

1 TimeSpan.Add(System.TimeSpan) Method

```
2 [ILASM]
3 .method public hidebysig instance valuetype System.TimeSpan
4 Add(valuetype System.TimeSpan ts)
5
6 [C#]
7 public TimeSpan Add(TimeSpan ts)
```

7 Summary

8 Adds the specified **System.TimeSpan** to the current instance.

9 Parameters

10
11

Parameter	Description
<code>ts</code>	A System.TimeSpan instance to add to the current instance.

12

13 Return Value

14

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

17 Exceptions

18
19

Exception	Condition
System.OverflowException	The sum of the value of the current instance and the value of <code>ts</code> is less than System.TimeSpan.MinValue or greater than System.TimeSpan.MaxValue .

20

21 Example

22

23 This example demonstrates the **System.TimeSpan.Add** method.

24

25

```
26 using System;
27 public class TimeSpanAddExample {
28     public static void Main() {
29         TimeSpan ts = new TimeSpan(Int32.MaxValue);
30         Console.WriteLine("The value of the timespan 'ts' is
31 {0}", ts);
32         Console.WriteLine("ts.Add(ts) = {0}", ts.Add(ts));
33     }
34 }
```

1
2
3
4
5
6
7

The output is

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

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

8

1 TimeSpan.Compare(System.TimeSpan, 2 System.TimeSpan) Method

```
3 [ILASM]  
4 .method public hidebysig static int32 Compare(valuetype  
5 System.TimeSpan t1, valuetype System.TimeSpan t2)  
6 [C#]  
7 public static int Compare(TimeSpan t1, TimeSpan t2)
```

8 Summary

9 Returns the sort order of two **System.TimeSpan** structures.

10 Parameters

11
12

Parameter	Description
<i>t1</i>	The first System.TimeSpan to compare.
<i>t2</i>	The second System.TimeSpan to compare.

13
14
15

14 Return Value

16 A **System.Int32** containing a value that reflects the sort order of *t1*
17 as compared to *t2*. The following table defines the conditions under
18 which the returned value is a negative number, zero, or a positive
19 number.

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

20
21

1 TimeSpan.CompareTo(System.Object) 2 Method

```
3 [ILASM]  
4 .method public final hidebysig virtual int32  
5 CompareTo(object value)  
  
6 [C#]  
7 public int CompareTo(object value)
```

8 Summary

9 Returns the sort order of the current instance compared to the
10 specified **System.Object**.

11 Parameters

12
13

Parameter	Description
<i>value</i>	The System.Object to compare to the current instance.

14
15
16

Return Value

17 A **System.Int32** containing a value that reflects the sort order of the
18 current instance as compared to *value*. The following table defines the
19 conditions under which the returned value is a negative number, zero,
20 or a positive number.

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

21

22 Description

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

25 Exceptions

26
27

Exception	Condition
-----------	-----------

1
2
3

System.ArgumentException	<i>value</i> is not a System.TimeSpan and is not a null reference.
---------------------------------	---

1 TimeSpan.Duration() Method

```
2 [ILASM]
3 .method public hidebysig instance valuetype System.TimeSpan
4 Duration()
5
6 [C#]
7 public TimeSpan Duration()
```

7 Summary

8 Returns a **System.TimeSpan** whose value is the absolute value of the
9 current instance.

10 Return Value

11

12 A **System.TimeSpan** whose value is the absolute value of the current
13 instance.

14 Exceptions

15

16

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

17

18 Example

19

20 The following example demonstrates the
21 **System.TimeSpan.Duration** method.

22

23

```
24 using System;
25 public class TimeSpanDurationExample {
26     public static void Main() {
27         TimeSpan ts = new TimeSpan(Int32.MinValue);
28         Console.WriteLine("The absolute value of TimeSpan {0} ",
29 ts);
30         Console.WriteLine("is {0}", ts.Duration());
31     }
32 }
```

33 The output is

34

```
1 The absolute value of TimeSpan -00:03:34.7483648 is
2 00:03:34.7483648
3
```

1 TimeSpan.Equals(System.Object) Method

```
2 [ILASM]  
3 .method public hidebysig virtual bool Equals(object value)  
4 [C#]  
5 public override bool Equals(object value)
```

6 Summary

7 Determines whether the current instance and the specified
8 **System.Object** represent the same type and value.

9 Parameters

10
11

Parameter	Description
<i>value</i>	The System.Object to compare to the current instance.

12
13
14

Return Value

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

18 Description

19 [Note: This method overrides **System.Object.Equals**.]
20

1 TimeSpan.Equals(System.TimeSpan, 2 System.TimeSpan) Method

```
3 [ILASM]  
4 .method public hidebysig static bool Equals(valuetype  
5 System.TimeSpan t1, valuetype System.TimeSpan t2)  
  
6 [C#]  
7 public static bool Equals(TimeSpan t1, TimeSpan t2)
```

8 Summary

9 Determines whether two **System.TimeSpan** structures represent the
10 same type and value.

11 Parameters

12
13

Parameter	Description
<i>t1</i>	The first instance of System.TimeSpan to compare for equality.
<i>t2</i>	The second instance of System.TimeSpan to compare for equality.

14
15
16

15 Return Value

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

18

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
<i>value</i>	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 <i>value</i> is greater than System.TimeSpan.MaxValue or less than System.TimeSpan.MinValue .
System.ArgumentException	<i>value</i> is equal to System.Double.NaN .

1
2
3

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

3 **TimeSpan.FromHours(System.Double)** 4 **Method**

```
5 [ILASM]  
6 .method public hidebysig static valuetype System.TimeSpan  
7 FromHours(float64 value)  
8 [C#]  
9 public static TimeSpan FromHours(double value)
```

10 **Summary**

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

13 **Parameters**

Parameter	Description
<i>value</i>	A System.Double that specifies the number of hours with which the new System.TimeSpan is initialized.

17 **Return Value**

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

20 **Description**

21 *value* will only be considered accurate to the nearest millisecond.

22
23 If *value* is **System.Double.PositiveInfinity**, a **System.TimeSpan**
24 with the value **System.TimeSpan.MaxValue** is returned. If *value* is
25 **System.Double.NegativeInfinity**, a **System.TimeSpan** with the
26 value **System.TimeSpan.MinValue** is returned.

27 **Exceptions**

Exception	Condition
System.OverflowException	The System.TimeSpan represented by <i>value</i> is greater than System.TimeSpan.MaxValue or less than System.TimeSpan.MinValue .

1
2
3

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
<i>value</i>	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 <i>value</i> is greater than System.TimeSpan.MaxValue or less

1
2
3

	than System.TimeSpan.MinValue .
System.ArgumentException	<i>value</i> is equal to System.Double.NaN .

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

3 TimeSpan.FromMinutes(System.Double) 4 Method

```
5 [ILASM]  
6 .method public hidebysig static valuetype System.TimeSpan  
7 FromMinutes(float64 value)  
8 [C#]  
9 public static TimeSpan FromMinutes(double value)
```

10 Summary

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

13 Parameters

Parameter	Description
<i>value</i>	A System.Double that specifies the number of minutes with which the new System.TimeSpan is initialized.

17 Return Value

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

20 Description

21 *value* will only be considered accurate to the nearest millisecond.

22
23 If *value* is **System.Double.PositiveInfinity**, a **System.TimeSpan**
24 with the value **System.TimeSpan.MaxValue** is returned. If *value* is
25 **System.Double.NegativeInfinity**, a **System.TimeSpan** with the
26 value **System.TimeSpan.MinValue** is returned.

27 Exceptions

Exception	Condition
System.OverflowException	The System.TimeSpan represented by <i>value</i> is greater than System.TimeSpan.MaxValue or less than System.TimeSpan.MinValue .

1
2
3

System.ArgumentException <i>value</i> is equal to System.Double.NaN .

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

3 **TimeSpan.FromSeconds(System.Double)** 4 **Method**

```
5 [ILASM]  
6 .method public hidebysig static valuetype System.TimeSpan  
7 FromSeconds(float64 value)  
8 [C#]  
9 public static TimeSpan FromSeconds(double value)
```

10 **Summary**

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

13 **Parameters**

Parameter	Description
<i>value</i>	A System.Double that specifies the number of seconds with which the new System.TimeSpan is initialized.

17 **Return Value**

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

20 **Description**

21 *value* will only be considered accurate to the nearest millisecond.

22
23 If *value* is **System.Double.PositiveInfinity**, a **System.TimeSpan**
24 with the value **System.TimeSpan.MaxValue** is returned. If *value* is
25 **System.Double.NegativeInfinity**, a **System.TimeSpan** with the
26 value **System.TimeSpan.MinValue** is returned.

27 **Exceptions**

Exception	Condition
System.OverflowException	The System.TimeSpan represented by <i>value</i> is greater than System.TimeSpan.MaxValue or less than System.TimeSpan.MinValue .

1
2
3

System.ArgumentException <i>value</i> is equal to System.Double.NaN .

1 TimeSpan.FromTicks(System.Int64)

2 Method

```
3 [ILASM]  
4 .method public hidebysig static valuetype System.TimeSpan  
5 FromTicks(int64 value)  
  
6 [C#]  
7 public static TimeSpan FromTicks(long value)
```

8 Summary

9 Returns a **System.TimeSpan** that represents the specified number of
10 ticks.

11 Parameters

12
13

Parameter	Description
<i>value</i>	A System.Int64 that specifies the number of ticks with which the new System.TimeSpan is initialized.

14
15
16

Return Value

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

18 Description

19 This method is equivalent to the **System.TimeSpan(System.Int64)**
20 constructor.

21

1 TimeSpan.GetHashCode() Method

```
2 [ILASM]  
3 .method public hidebysig virtual int32 GetHashCode()  
4 [C#]  
5 public override int GetHashCode()
```

6 Summary

7 Generates a hash code for the current instance.

8 Return Value

9

10 A **System.Int32** value containing a hash code for the current
11 instance.

12 Description

13 The algorithm used to generate the hash code is unspecified.

14

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

16

1 TimeSpan.Negate() Method

```
2 [ILASM]  
3 .method public hidebysig instance valuetype System.TimeSpan  
4 Negate()  
5  
6 [C#]  
7 public TimeSpan Negate()
```

7 Summary

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

10 Return Value

11

12 A **System.TimeSpan** with the same absolute value but with the
13 opposite sign as the current instance.

14 Exceptions

15

16

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

17

18

19

1 TimeSpan.op_Addition(System.TimeSpan, 2 System.TimeSpan) Method

```
3 [ILASM]  
4 .method public hidebysig static specialname valuetype  
5 System.TimeSpan op_Addition(valuetype System.TimeSpan t1,  
6 valuetype System.TimeSpan t2)  
7  
8 [C#]  
9 public static TimeSpan operator +(TimeSpan t1, TimeSpan t2)
```

9 Summary

10 Adds the values of two **System.TimeSpan** instances.

11 Parameters

12
13

Parameter	Description
<i>t1</i>	The first System.TimeSpan .
<i>t2</i>	The second System.TimeSpan .

14
15
16

15 Return Value

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

19 Exceptions

20
21

Exception	Condition
System.OverflowException	The sum of <i>t1</i> and <i>t2</i> is less than System.TimeSpan.MinValue or greater than System.TimeSpan.MaxValue .

22
23
24

1 TimeSpan.op_Equality(System.TimeSpan, 2 System.TimeSpan) Method

```
3 [ILASM]  
4 .method public hidebysig static specialname bool  
5 op_Equality(valuetype System.TimeSpan t1, valuetype  
6 System.TimeSpan t2)  
  
7 [C#]  
8 public static bool operator ==(TimeSpan t1, TimeSpan t2)
```

9 Summary

10 Determines whether the value of one **System.TimeSpan** is equal to
11 the value of another **System.TimeSpan**.

12 Parameters

13
14

Parameter	Description
<i>t1</i>	The first System.TimeSpan
<i>t2</i>	The second System.TimeSpan

15
16
17

Return Value

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

19

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

```
3 [ILASM]
4 .method public hidebysig static specialname bool
5 op_GreaterThan(valuetype System.TimeSpan t1, valuetype
6 System.TimeSpan t2)
7
8 [C#]
9 public static bool operator >(TimeSpan t1, TimeSpan t2)
```

9 Summary

10 Determines whether the value one **System.TimeSpan** is greater than
11 the value of another **System.TimeSpan**.

12 Parameters

13
14

Parameter	Description
<i>t1</i>	The first System.TimeSpan .
<i>t2</i>	The second System.TimeSpan .

15
16
17

16 Return Value

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

19

1 TimeSpan.op_GreaterThanOrEqual(System. 2 m.TimeSpan, System.TimeSpan) Method

```
3 [ILASM]  
4 .method public hidebysig static specialname bool  
5 op_GreaterThanOrEqual(valuetype System.TimeSpan t1,  
6 valuetype System.TimeSpan t2)  
7  
8 [C#]  
9 public static bool operator >=(TimeSpan t1, TimeSpan t2)
```

9 Summary

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

12 Parameters

13
14

Parameter	Description
<i>t1</i>	The first System.TimeSpan .
<i>t2</i>	The second System.TimeSpan .

15
16
17

16 Return Value

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

20

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

```
3 [ILASM]
4 .method public hidebysig static specialname bool
5 op_Inequality(valuetype System.TimeSpan t1, valuetype
6 System.TimeSpan t2)
7
8 [C#]
9 public static bool operator !=(TimeSpan t1, TimeSpan t2)
```

9 Summary

10 Determines whether the value of one **System.TimeSpan** is unequal to
11 the value of another **System.TimeSpan**.

12 Parameters

13
14

Parameter	Description
<i>t1</i>	The first System.TimeSpan .
<i>t2</i>	The second System.TimeSpan .

15
16
17

16 Return Value

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

19

1 TimeSpan.op_LessThan(System.TimeSpan 2 , System.TimeSpan) Method

```
3 [ILASM]  
4 .method public hidebysig static specialname bool  
5 op_LessThan(valuetype System.TimeSpan t1, valuetype  
6 System.TimeSpan t2)  
  
7 [C#]  
8 public static bool operator <(TimeSpan t1, TimeSpan t2)
```

9 Summary

10 Determines whether the value of one **System.TimeSpan** is less than
11 the value of another **System.TimeSpan**.

12 Parameters

13
14

Parameter	Description
<i>t1</i>	The first System.TimeSpan .
<i>t2</i>	The second System.TimeSpan .

15
16
17

16 Return Value

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

19

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

```
3 [ILASM]
4 .method public hidebysig static specialname bool
5 op_LessThanOrEqual(valuetype System.TimeSpan t1, valuetype
6 System.TimeSpan t2)
7
8 [C#]
9 public static bool operator <=(TimeSpan t1, TimeSpan t2)
```

9 Summary

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

12 Parameters

Parameter	Description
<i>t1</i>	The first System.TimeSpan .
<i>t2</i>	The second System.TimeSpan .

15 Return Value

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

20

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

```
3 [ILASM]
4 .method public hidebysig static specialname valuetype
5 System.TimeSpan op_Subtraction(valuetype System.TimeSpan
6 t1, valuetype System.TimeSpan t2)
7
8 [C#]
9 public static TimeSpan operator -(TimeSpan t1, TimeSpan t2)
```

9 Summary

10 Subtracts the value of one **System.TimeSpan** from the value of
11 another **System.TimeSpan**.

12 Parameters

13
14

Parameter	Description
<i>t1</i>	The first System.TimeSpan .
<i>t2</i>	The second System.TimeSpan .

15
16
17

16 Return Value

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

20 Exceptions

21
22

Exception	Condition
System.OverflowException	The value of <i>t2</i> subtracted from <i>t1</i> is less than System.TimeSpan.MinValue or greater than System.TimeSpan.MaxValue .

23
24
25

1 TimeSpan.op_UnaryNegation(System.TimeSpan) Method

```
3 [ILASM]  
4 .method public hidebysig static specialname valuetype  
5 System.TimeSpan op_UnaryNegation(valuetype System.TimeSpan  
6 t)  
7  
8 [C#]  
9 public static TimeSpan operator -(TimeSpan t)
```

9 Summary

10 Returns a **System.TimeSpan** whose value is the negated value of a
11 specified **System.TimeSpan**.

12 Parameters

13
14

Parameter	Description
<i>t</i>	A System.TimeSpan whose value will be negated.

15
16
17

16 Return Value

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

20 Exceptions

21
22

Exception	Condition
System.OverflowException	<i>t</i> equals System.TimeSpan.MinValue .

23
24
25

1 TimeSpan.op_UnaryPlus(System.TimeSpan) 2 n) Method

```
3 [ILASM]  
4 .method public hidebysig static specialname valuetype  
5 System.TimeSpan op_UnaryPlus(valuetype System.TimeSpan t)  
6  
7 [C#]  
8 public static TimeSpan operator +(TimeSpan t)
```

8 Summary

9 Returns the specified instance of **System.TimeSpan**.

10 Parameters

11
12

Parameter	Description
<i>t</i>	A System.TimeSpan .

13
14
15

14 Return Value

16 **System.TimeSpan** *t*.

17 Description

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

19

1 TimeSpan.Parse(System.String) Method

```
2 [ILASM]  
3 .method public hidebysig static valuetype System.TimeSpan  
4 Parse(string s)  
  
5 [C#]  
6 public static TimeSpan Parse(string s)
```

7 Summary

8 Returns the specified **System.String** converted to a
9 **System.TimeSpan** value.

10 Parameters

11
12

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.

13
14
15

Return Value

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

17 Description

18 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.

19

1 **Exceptions**

2

3

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.

4

5 **Example**

6

7 This example demonstrates parsing a string to obtain a
8 **System.TimeSpan**.

9

10

[C#]

11

```
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);
    }
}
```

12

13

14

15

16

17

18

19

20

21 The output is

22

23

24

25

26

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

1 TimeSpan.Subtract(System.TimeSpan)

2 Method

```
3 [ILASM]  
4 .method public hidebysig instance valuetype System.TimeSpan  
5 Subtract(valuetype System.TimeSpan ts)  
  
6 [C#]  
7 public TimeSpan Subtract(TimeSpan ts)
```

8 Summary

9 Subtracts the value of the specified **System.TimeSpan** from the value
10 of the current instance.

11 Parameters

12
13

Parameter	Description
<i>ts</i>	A System.TimeSpan whose value to subtract from the value of the current instance.

14
15
16

Return Value

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

19 Exceptions

20
21

Exception	Condition
System.OverflowException	The difference between the value of the current instance and <i>ts</i> is less than System.TimeSpan.MinValue or greater than System.TimeSpan.MaxValue .

22
23
24

1 TimeSpan.ToString() Method

```
2 [ILASM]  
3 .method public hidebysig virtual string ToString()  
4 [C#]  
5 public override string ToString()
```

6 Summary

7 Returns a **System.String** representation of the value of the current
8 instance.

9 Return Value

10

11 A **System.String** representation of the current instance formatted as
12 follows:

13

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

15

16 Items in square brackets ('[' and ']') are included provisionally: '-' is
17 included if and only if the current instance is negative; "d." and ".ff"
18 are included if and only if those components are non-zero. Colons and
19 periods (':' and '.') are literal characters. Other components are as
20 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.

21

22 Description

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

24 Example

25

26 This example demonstrates the **System.TimeSpan.ToString** method.

27

28 [C#]

```
1      using System;
2      public class TimeSpanToStringExample {
3          public static void Main() {
4              TimeSpan tsOne = new TimeSpan(1, 23, 45, 54, 321);
5              TimeSpan tsTwo = new TimeSpan(0, 23, 45, 54, 0);
6              Console.Write("TimeSpan one, with d. and.ff: ");
7              Console.WriteLine("{0}", tsOne.ToString());
8              Console.Write("TimeSpan two, without d. and.ff: ");
9              Console.WriteLine("{0}", tsTwo.ToString());
10         }
11     }
```

12 The output is

13
14 TimeSpan one, with d. and.ff: 1.23:45:54.3210000

15
16
17 TimeSpan two, without d. and.ff: 23:45:54

18

19

1 TimeSpan.Days Property

```
2 [ILASM]
3 .property int32 Days { public hidebysig specialname
4 instance int32 get_Days() }
5
6 [C#]
7 public int Days { get; }
```

7 Summary

8 Gets the number days represented by the current instance.

9 Property Value

10

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

14 Description

15 This property is read-only.

16 Example

17

18 This example demonstrates using the **System.TimeSpan.Days**
19 property.

20

21

```
22 using System;
23 public class TimeSpanPropertiesExampleOne {
24     public static void Main() {
25         TimeSpan ts = new TimeSpan((Int64)10e12+3456789);
26         Console.WriteLine(ts.ToString());
27         Console.WriteLine("Days: {0}", ts.Days);
28     }
29 }
```

30 The output is

31

32

11.13:46:40.3456789

33

34

1 Days: 11
2
3

1 TimeSpan.Hours Property

```
2 [ILASM]
3 .property int32 Hours { public hidebysig specialname
4 instance int32 get_Hours() }
5 [C#]
6 public int Hours { get; }
```

7 Summary

8 Gets the number of hours represented by the current instance.

9 Property Value

10

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

15 Description

16 This property is read-only.

17 Example

18

19 This example demonstrates using the **System.TimeSpan.Hours**
20 property.

21

22

```
23 using System;
24 public class TimeSpanPropertiesExampleOne {
25     public static void Main() {
26         TimeSpan ts = new TimeSpan((Int64)10e12+3456789);
27         Console.WriteLine(ts.ToString());
28         Console.WriteLine("Hours: {0}", ts.Hours);
29     }
30 }
```

31 The output is

32

33

34

35

11.13:46:40.3456789

1 Hours: 13
2
3

1 TimeSpan.Milliseconds Property

```
2 [ILASM]
3 .property int32 Milliseconds { public hidebysig specialname
4 instance int32 get_Milliseconds() }
5
6 [C#]
7 public int Milliseconds { get; }
```

7 Summary

8 Gets the number of milliseconds represented by the current instance.

9 Property Value

10

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

15 Description

16 This property is read-only.

17 Example

18

19 This example demonstrates using the
20 **System.TimeSpan.Milliseconds** property.

21

22

```
23 using System;
24 public class TimeSpanPropertiesExampleOne {
25     public static void Main() {
26         TimeSpan ts = new TimeSpan((Int64)10e12+3456789);
27         Console.WriteLine(ts.ToString());
28         Console.WriteLine("Milliseconds: {0}",
29 ts.Milliseconds);
30     }
31 }
```

32 The output is

33

34

35

11.13:46:40.3456789

1
2
3
4

Milliseconds: 345

1 TimeSpan.Minutes Property

```
2 [ILASM]
3 .property int32 Minutes { public hidebysig specialname
4 instance int32 get_Minutes() }
5 [C#]
6 public int Minutes { get; }
```

7 Summary

8 Gets the number of minutes represented by the current instance.

9 Property Value

10

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

15 Description

16 This property is read-only.

17 Example

18

19 This example demonstrates using the **System.TimeSpan.Minutes**
20 property.

21

22

```
23 using System;
24 public class TimeSpanPropertiesExampleOne {
25     public static void Main() {
26         TimeSpan ts = new TimeSpan((Int64)10e12+3456789);
27         Console.WriteLine(ts.ToString());
28         Console.WriteLine("Minutes: {0}", ts.Minutes);
29     }
30 }
```

31 The output is

32

33

34

35

11.13:46:40.3456789

1 Minutes: 46
2
3

1 TimeSpan.Seconds Property

```
2 [ILASM]
3 .property int32 Seconds { public hidebysig specialname
4 instance int32 get_Seconds() }
5 [C#]
6 public int Seconds { get; }
```

7 Summary

8 Gets the number of seconds represented by the current instance.

9 Property Value

10

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

15 Description

16 This property is read-only.

17 Example

18

19 This example demonstrates using the **System.TimeSpan.Seconds**
20 property.

21

22

```
[C#]
```

23

```
using System;
```

24

```
public class TimeSpanPropertiesExampleOne {
```

25

```
    public static void Main() {
```

26

```
        TimeSpan ts = new TimeSpan((Int64)10e12+3456789);
```

27

```
        Console.WriteLine(ts.ToString());
```

28

```
        Console.WriteLine("Seconds: {0}", ts.Seconds);
```

29

```
    }
```

30

```
}
```

31

The output is

32

33

```
11.13:46:40.3456789
```

34

35

1 Seconds: 40
2
3

1 TimeSpan.Ticks Property

```
2 [ILASM]  
3 .property int64 Ticks { public hidebysig specialname  
4 instance int64 get_Ticks() }  
5 [C#]  
6 public long Ticks { get; }
```

7 Summary

8 Gets the number of ticks represented by the current instance.

9 Property Value

10

11 A **System.Int64** specifying the number of ticks represented by the
12 current instance.

13 Description

14 This property is read-only.

15

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

3 TimeSpan.TotalDays Property

```
4 [ILASM]  
5 .property float64 TotalDays { public hidebysig specialname  
6 instance float64 get_TotalDays() }  
7 [C#]  
8 public double TotalDays { get; }
```

9 Summary

10 Gets the value of the current instance expressed in days.

11 Property Value

12

13 A **System.Double** that specifies the total number of days represented
14 by the current instance.

15 Description

16 This property is read-only.

17

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

20 Example

21

22 This example demonstrates using the **System.TimeSpan.TotalDays**
23 property.

24

25

```
26 using System;  
27 public class TimeSpanTotalUnitsProperties{  
28     public static void Main() {  
29         TimeSpan ts = new TimeSpan((Int64)10e12);  
30         Console.WriteLine(ts.ToString());  
31         Console.WriteLine("TotalDays: {0}", ts.TotalDays);  
32     }  
33 }
```

34 The output is

35

36

11.13:46:40

1
2
3
4
5

TotalDays: 11.5740740740741

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

3 TimeSpan.TotalHours Property

```
4 [ILASM]  
5 .property float64 TotalHours { public hidebysig specialname  
6 instance float64 get_TotalHours() }  
  
7 [C#]  
8 public double TotalHours { get; }
```

9 Summary

10 Gets the value of the current instance expressed in hours.

11 Property Value

13 A **System.Double** that specifies the total number of hours
14 represented by the current instance.

15 Description

16 This property is read-only.

17
18 [*Note:* This property converts the value of the current instance from
19 ticks to hours. This number may include whole and fractional hours.]

20 Example

22 This example demonstrates using the **System.TimeSpan.TotalHours**
23 property.

```
24 [C#]  
25  
26 using System;  
27 public class TimeSpanTotalUnitsProperties{  
28     public static void Main() {  
29         TimeSpan ts = new TimeSpan((Int64)10e12);  
30         Console.WriteLine(ts.ToString());  
31         Console.WriteLine("TotalHours: {0}", ts.TotalHours);  
32     }  
33 }
```

34 The output is

35 11.13:46:40
36

1
2
3
4
5

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 may 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

1 11.13:46:40
2
3
4 TotalMilliseconds: 1000000000
5
6

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

3 TimeSpan.TotalMinutes Property

```
4 [ILASM]  
5 .property float64 TotalMinutes { public hidebysig  
6 specialname instance float64 get_TotalMinutes() }  
7 [C#]  
8 public double TotalMinutes { get; }
```

9 Summary

10 Gets the value of the current instance expressed in minutes.

11 Property Value

13 A **System.Double** that specifies the total number of minutes
14 represented by the current instance.

15 Description

16 This property is read-only.

17
18 [*Note:* This property converts the value of the current instance from
19 ticks to minutes. This number may include whole and fractional
20 minutes.]

21 Example

23 This example demonstrates using the
24 **System.TimeSpan.TotalMinutes** property.

```
25 [C#]  
26  
27 using System;  
28 public class TimeSpanTotalUnitsProperties{  
29     public static void Main() {  
30         TimeSpan ts = new TimeSpan((Int64)10e12);  
31         Console.WriteLine(ts.ToString());  
32         Console.WriteLine("TotalMinutes: {0}",  
33         ts.TotalMinutes);  
34     }  
35 }
```

36 The output is
37

1
2
3
4
5
6

11.13:46:40

TotalMinutes: 16666.6666666667

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

3 TimeSpan.TotalSeconds Property

```
4 [ILASM]  
5 .property float64 TotalSeconds { public hidebysig  
6 specialname instance float64 get_TotalSeconds() }  
7 [C#]  
8 public double TotalSeconds { get; }
```

9 Summary

10 Gets the value of the current instance expressed in seconds.

11 Property Value

13 A **System.Double** that specifies the total number of seconds
14 represented by the current instance.

15 Description

16 This property is read-only.

17
18 [*Note:* This property converts the value of the current instance from
19 ticks to seconds. This number may include whole and fractional
20 seconds.]

21 Example

23 This example demonstrates using the
24 **System.TimeSpan.TotalSeconds** property.

```
25 [C#]  
26  
27 using System;  
28 public class TimeSpanTotalUnitsProperties{  
29     public static void Main() {  
30         TimeSpan ts = new TimeSpan((Int64)10e12);  
31         Console.WriteLine(ts.ToString());  
32         Console.WriteLine("TotalSeconds: {0}",  
33         ts.TotalSeconds);  
34     }  
35 }
```

36 The output is
37

1 11.13:46:40
2
3
4 TotalSeconds:1000000
5
6