1. DateTime

```
amespace CSharpFundamentals
     class Program
           static void Main(string[] args)
                  DateTime

■ DateTime (in System)

                  DataSetDateTime (in System.Data)
                  DateTimeConstantAttribute (in System.Runtime.CompilerServices)
                  DateTimeConverter (in System.ComponentModel)
                  tion) 🙀 DateTimeFormatInfo (in System.Globalization)
                  DateTimeKind (in System)

    □ DateTimeOffset (in System)

                  DateTimeOffsetConverter (in System.ComponentModel)
                  DateTimeSerializationMode (in System.Xml.Serialization.Configuration.DateTimeSerializationSection)
                  DateTimeSerializationSection (in System.Xml.Serialization.Configuration)
                  DateTimeStyles (in System.Globalization)
```

```
sing System;
amespace CSharpFundamentals
                                                             (<no parameters>)
    class Program
                                                             (int year, int month, int day)
                                                             (int year, int month, int day, [NotNull] Calendar calendar)
                                                            (int year, int month, int day, int hour, int minute, int second)
           static void Main(string[] args)
                                                             (int year, int month, int day, int hour, int minute, int second, DateTimeKind kind)
                 var dateTime = new DateTime(
                                                              ⁰ ValueType
                                                              Version

■ Void

                                                              WeakReference
                                                              WeakReference<>
                                                              _AppDomain
                                                             (♠) args
                                                                                                       Parameter string[] args
                                                              async
                                                              await
                                                              Dool
                                                              byte
                                                              () ♦ → □ □ □ □ □ □ (•) ₩ ☑ 🔒 🚭
```

```
using System;
¤namespace CSharpFundamentals
      class Program
           static void Main(string[] args)
                 var dateTime = new DateTime(2015, 1, 1);
                 DateTime.Now
                             ■ Now
                                            Property DateTime System.DateTime.Now
                             ■ UtcNow
                                            Gets a DateTime object that is set to the current date and time on this computer, expressed as the local time.
```

```
→ 🗣 Main(string[]
CSharp Fundamentals. Program
   using System;
  pnamespace CSharpFundamentals
       class Program
            static void Main(string[] args)
                var dateTime = new DateTime(2015, 1, 1);
                var now = DateTime.Now;
                var today = DateTime.Today;
```

```
using System;
pnamespace CSharpFundamentals
     class Program
        static void Main(string[] args)
             var dateTime = new DateTime(2015, 1, 1);
             var now = DateTime.Now;
             var today = DateTime.Today;
             Console.WriteLine("Hour: " + now.Hour);
             Console.WriteLine("Minute: " + now.Minute);
```

```
Hour: 12
Minute: 40
Press any key to continue . . .
```

```
using System;
pnamespace CSharpFundamentals
     class Program
         static void Main(string[] args)
             var dateTime = new DateTime(2015, 1, 1);
             var now = DateTime.Now;
             var today = DateTime.Today;
             Console.WriteLine("Hour: " + now.Hour);
             Console.WriteLine("Minute: " + now.Minute);
             now.Add
                  Add 
                  AddDays
                  AddHours
                  AddMilliseconds
                  AddMinutes
                  AddMonths
                  AddSeconds
                  AddTicks
                  AddYears
                  =
```

```
using System;
¤namespace CSharpFundamentals
    class Program
         static void Main(string[] args)
             var dateTime = new DateTime(2015, 1, 1);
            var now = DateTime.Now;
             var today = DateTime.Today;
             Console.WriteLine("Hour: " + now.Hour);
             Console.WriteLine("Minute: " + now.Minute);
             var tomrrow = now.AddDays(1);
```

```
using System;
¤namespace CSharpFundamentals
     class Program
         static void Main(string[] args)
             var dateTime = new DateTime(2015, 1, 1);
             var now = DateTime.Now;
             var today = DateTime.Today;
             Console.WriteLine("Hour: " + now.Hour);
             Console.WriteLine("Minute: " + now.Minute);
             var tomrrow = now.AddDays(1);
             var yesterday = now.AddDays(-1);
```

```
using System;
□namespace CSharpFundamentals
     class Program
         static void Main(string[] args)
             var dateTime = new DateTime(2015, 1, 1);
             var now = DateTime.Now;
             var today = DateTime.Today;
             //Console.WriteLine("Hour: " + now.Hour);
             //Console.WriteLine("Minute: " + now.Minute);
             var tomrrow = now.AddDays(1);
             var yesterday = now.AddDays(-1);
             Console.WriteLine(now.ToLongDateString());
             Console.WriteLine(now.ToShortDateString());
             Console.WriteLine(now.ToLongTimeString());
             Console.WriteLine(now.ToShortTimeString());
```

```
Saturday, 23 May 2015
23/05/2015
12:43:31 PM
12:43 PM
Press any key to continue . . . _
```

```
using System;
□namespace CSharpFundamentals
     class Program
         static void Main(string[] args)
             var dateTime = new DateTime(2015, 1, 1);
            var now = DateTime.Now;
             var today = DateTime.Today;
             //Console.WriteLine("Hour: " + now.Hour);
             //Console.WriteLine("Minute: " + now.Minute);
            var tomrrow = now.AddDays(1);
             var yesterday = now.AddDays(-1);
             Console.WriteLine(now.ToLongDateString());
             Console.WriteLine(now.ToShortDateString());
             Console.WriteLine(now.ToLongTimeString());
             Console.WriteLine(now.ToShortTimeString());
            Console.WriteLine(now.ToString());
```

```
Saturday, 23 May 2015
23/05/2015
12:44:36 PM
12:44 PM
23/05/2015 12:44:36 PM
Press and key to continue . . . .
```

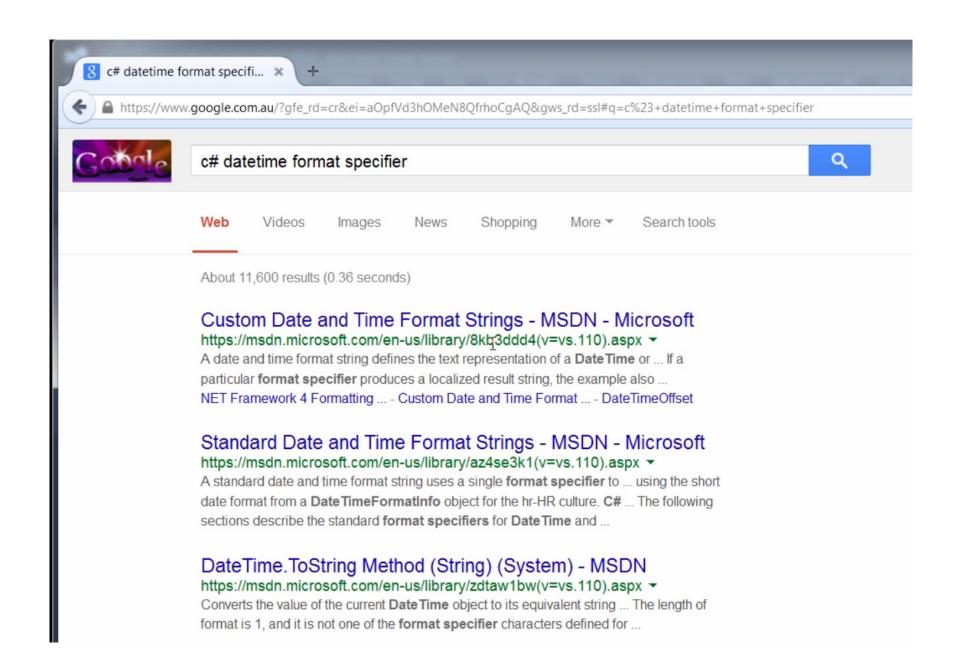
```
static void Main(string | args)
        var dateTime = new DateTime(2015, 1, 1);
        var now = DateTime.Now;
        var today = DateTime.Today;
        //Console.WriteLine("Hour: " + now.Hour);
        //Console.WriteLine("Minute: " + now.Minute);
        var tomrrow = now.AddDays(1);
        var yesterday = now.AddDays(-11)
                                                     (IFormatProvider provider):string
                                                      (string format, IFormatProvider provider):string
        Console.WriteLine(now.ToLongDa [NotNull] (<no parameters>): string
        Console.WriteLine(now.ToShortD [NotNull] (string format):string
                                                       Converts the value of the current DateTime object to its equivalent
        Console.WriteLine(now.ToLongTi
                                                       string representation using the specified format.
        Console.WriteLine(now.ToShortT
                                                       format: A standard or custom date and time format string (see Remarks).
        Console.WriteLine(now.ToString(""));
                                                        ■ D - long date
                                                                              Thursday, 10 April 2008
                                                        d - short date
                                                                                      04/10/2008
🗞 Program
                                                                        Thursday, 10 April 2008 06:30:00
                                                        ■ F - full date long
                                                        f - full date short
                                                                          Thursday, 10 April 2008 06:30
                                                        ■ G - general long
                                                                                04/10/2008 06:30:00
                                                        g - general short
                                                                                  04/10/2008 06:30
                                                        ■ M - month
                                                                                         April 10
                                                        O - ISO 8601
                                                                          2008-04-10T06:30:00.0000000
                                                        ■ R - RFC 1123
                                                                         Thu, 10 Apr 2008 06:30:00 GMT
                                                        s - sortable
                                                                                2008-04-10T06:30:00
                                                        ■ T - long time
                                                                                         06:30:00 🔻
```

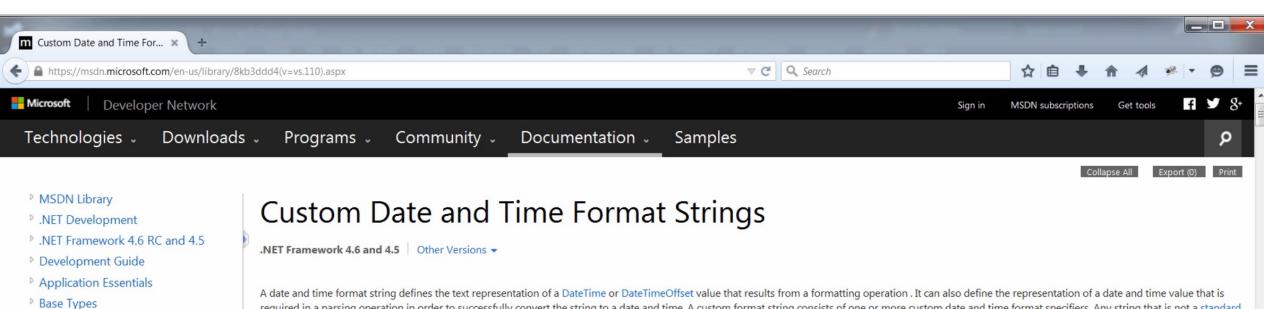
```
static void Main(string[] args)
   var dateTime = new DateTime(2015, 1, 1);
    var now = DateTime.Now;
   var today = DateTime.Today;
   //Console.WriteLine("Hour: " + now.Hour);
    //Console.WriteLine("Minute: " + now.Minute);
   var tomrrow = now.AddDays(1);
    var yesterday = now.AddDays(-1);
   Console.WriteLine(now.ToLongDateString());
    Console.WriteLine(now.ToShortDateString());
    Console.WriteLine(now.ToLongTimeString());
    Console.WriteLine(now.ToShortTimeString());
   Console.WriteLine(now.ToString("yyyy-MM-dd"));
```

```
Saturday, 23 May 2015
23/05/2015
12:46:19 PM
12:46 PM
2015-05-23
Press any key to continue . . .
```

```
static void Main(string[] args)
   var dateTime = new DateTime(2015, 1, 1);
   var now = DateTime.Now;
   var today = DateTime.Today;
   //Console.WriteLine("Hour: " + now.Hour);
    //Console.WriteLine("Minute: " + now.Minute);
   var tomrrow = now.AddDays(1);
   var yesterday = now.AddDays(-1);
   Console.WriteLine(now.ToLongDateString());
    Console.WriteLine(now.ToShortDateString());
    Console.WriteLine(now.ToLongTimeString());
    Console.WriteLine(now.ToShortTimeString());
   Console.WriteLine(now.ToString("yyyy-MM-dd HH:mm"));
```

Saturday, 23 May 2015 23/05/2015 12:46:43 PM 12:46 PM 2015-05-23 12:46 Press any key to continue . . .





required in a parsing operation in order to successfully convert the string to a date and time. A custom format string consists of one or more custom date and time format specifiers. Any string that is not a standard date and time format string is interpreted as a custom date and time format string.

Custom date and time format strings can be used with both DateTime and DateTimeOffset values.

Formatting Types

Strings

Strings

Strings

Strings

Operations

Format Strings

Format Strings

Standard Numeric Format

Custom Numeric Format

Standard Date and Time

Custom Date and Time

Standard TimeSpan Format

Custom TimeSpan Format

Enumeration Format Strings Composite Formatting

Performing Formatting

```
Tip

You can download the Formatting Utility, an application that enables you to apply format strings to either date and time or numeric values and displays the result string.
```

In formatting operations, custom date and time format strings can be used either with the **ToString** method of a date and time instance or with a method that supports composite formatting. The following example illustrates both uses.









- MSDN Library
- ▶ .NET Development
- ▶ .NET Framework 4.6 RC and 4.5
- Development Guide
- Application Essentials
- Base Types
- Formatting Types

Standard Numeric Format Strings

Custom Numeric Format Strings Standard Date and Time Format

Strings

Custom Date and Time Format Strings

Standard TimeSpan Format Strings

Custom TimeSpan Format Strings

Enumeration Format Strings

Composite Formatting

Performing Formatting Operations

The following table describes the custom date and time format specifiers and displays a result string produced by each format specifier. By default, result strings reflect the formatting conventions of the en-US culture. If a particular format specifier produces a localized result string, the example also notes the culture to which the result string applies. See the Notes section for additional information about using custom date and time format strings.

Format specifier	Description	Examples
"d"	The day of the month, from 1 through 31.	2009-06-01T13:45:30 -> 1
	More information: The "d" Custom Format Specifier.	2009-06-15T13:45:30 -> 15
"dd"	The day of the month, from 01 through 31.	2009-06-01T13:45:30 -> 01
	More information: The "dd" Custom Format Specifier.	2009-06-15T13:45:30 -> 15
"ddd"	The abbreviated name of the day of the week.	2009-06-15T13:45:30 -> Mon (en-US)
	More information: The "ddd" Custom Format Specifier.	2009-06-15T13:45:30 -> Пн (ru-RU)
		2009-06-15T13:45:30 -> lun. (fr-FR)
"dddd"	The full name of the day of the week.	2009-06-15T13:45:30 -> Monday (en-US)
	More information: The "dddd" Custom Format Specifier.	2009-06-15T13:45:30 -> понедельник (ru-RU)
		2009-06-15T13:45:30 -> lundi (fr-FR)
"f"	The tenths of a second in a date and time value.	2009-06-15T13:45:30.6170000 -> 6
	More information: The "f" Custom Format Specifier.	2009-06-15T13:45:30.05 -> 0
"ff"	The hundredths of a second in a date and time value.	2009-06-15T13:45:30.6170000 -> 61
	More information: The "ff" Custom Format Specifier.	2009-06-15T13:45:30.0500000 -> 00
"fff"	The milliseconds in a date and time value.	6/15/2009 13:45:30.617 -> 617
	More information: The "fff" Custom Format Specifier.	6/15/2009 13:45:30.0005 -> 000
"ffff"	The ten thousandths of a second in a date and time value.	2009-06-15T13:45:30.6175000 -> 6175