CHAPTER 10

The String Data Type

This chapter will discuss the string data type in Microsoft .NET Framework using C# language. First I will show how the CLR manages to instantiate a string in .NET. I will then discuss string immutability through which CLR ensures that when a string is created, it can't be changed, and examine its contents, chaining operations in string, and various concatenation techniques used in .NET Framework for the string.

Throughout the chapter, I reference StringBuilder, which is a class that can be used to generate string efficiently. It can also be used to manipulate string, such as append, insert, or remove string. You will see this class used in several examples, but it’s not until later in the chapter that I detail the internal workings of the StringBuilder class. There we will examine the constructor of the StringBuilder and the addition, insertion, and remove operations to see how CLR deals with string when using StringBuilder to generate it.

String in .NET

In C#, you can represent numbers such as 1, 2, 3, and so forth using Int32 data type as characters, such as 'A', 'B', or 'C' using char data type. If you want to represent a word, a sentence, and so on, you can use String data type. In .NET, C# string is a sealed class defined in the System namespace of themscorlib.dll assembly (located in C:\Windows\Microsoft.NET\Framework\v4.0.30319\mscorlib.dll), as shown in Figure 10-1.
The class definition of the String is extracted using the ildasm.exe, as shown in Listing 10-1.

Listing 10-1. Definition of the String Class in .NET

```csharp
.class public auto ansi serializable sealed beforefieldinit String
    extends
        System.Object
    implements
        System.IComparable, System.ICloneable, System.IConvertible,
        System.IComparable'1<string>, System.Collections.Generic.IEnumerable'1<char>,
        System.Collections.IEnumerable, System.IEquatable'1<string>
```

So based on the class definition, you can see that string class is derived from the System.Object. It is not possible to inherit a type from the String class as it is sealed. As the String class implements the IEnumerable'1<char> interface, you will be able to use the Linq (discussed in the Chapter 12) functionality over the String. Listing 10-2 gives an example of the String in .NET using C#.

Listing 10-2. An Example of String

```csharp
using System;
using System.Text;
namespace Ch10
{
    class Program
    {
        static void Main(string[] args)
        {
            string bookName = "Expert C# 5.0: with the .NET 4.5 Framework";
            /* CLR will create a String with - by repeating the number
             * of the Length of the bookName string .*/
            string dashedLine = new string('-', bookName.Length);
            StringBuilder sb = new StringBuilder("by Mohammad Rahman");
        }
    }
}
```