

[Come a member](#)[Login](#)[Post](#)[Ask Question](#)

# Various Ways To Specify Connection String In ASP.NET Web And Windows Applications



Priti Kumari

Updated date Aug 16, 2021

237.7k

19

5

[Download Free .NET & JAVA Office Files API](#)  
[Try Free File Format APIs for Word/Excel/PDF](#)

This article describes various ways to declare a connection string. We will see the following points.

1. Specify Database connection string in Web Application.
2. Specify Database connection string in Windows application.

Let's start from the first point.

## Specify Database connection string in Web Application

In a web application we can specify a database connection string in one of the following two ways.

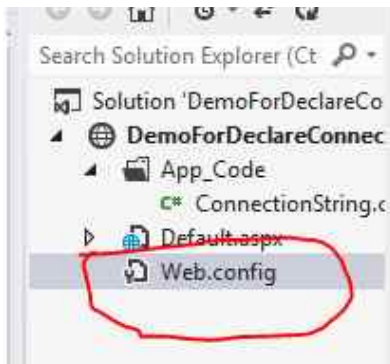
- Specify it in the web.config file.
- Create a common class file for the connection string.

## Write connection string in web.config file

The benefit of writing a connection string in a web config file is there is no need to write code every time to connect to the database. It reduces the redundancy of code over a single application. One more benefit of this is whenever we need to change the database name or credentials we need to change only the web config file.

Now I will describe how to specify a connection string in a *web.config* file.

Go to the Web.config file as in the following:



You need to use the *connectionStrings* tag to specify the connection string in the *web.config* file as in the following image.



Write this XML markup in the web config file for declaration of the connection. You can add many tags for each database connection.

```
01. <connectionStrings>
02.     <add name="mytest" connectionString="Data Source=.\SQLEXPRESS;Initial Cat
03.     <add name="Northwind" connectionString="Data Source=.\SQLEXPRESS;Initial
04.     <add name="Northwind" connectionString="Data Source=.\SQLEXPRESS;Initial
05. </connectionStrings>
```

## Read Connection string from Web.config file

Now I will describe how to read a web.config connection string in an ASP.NET web application.

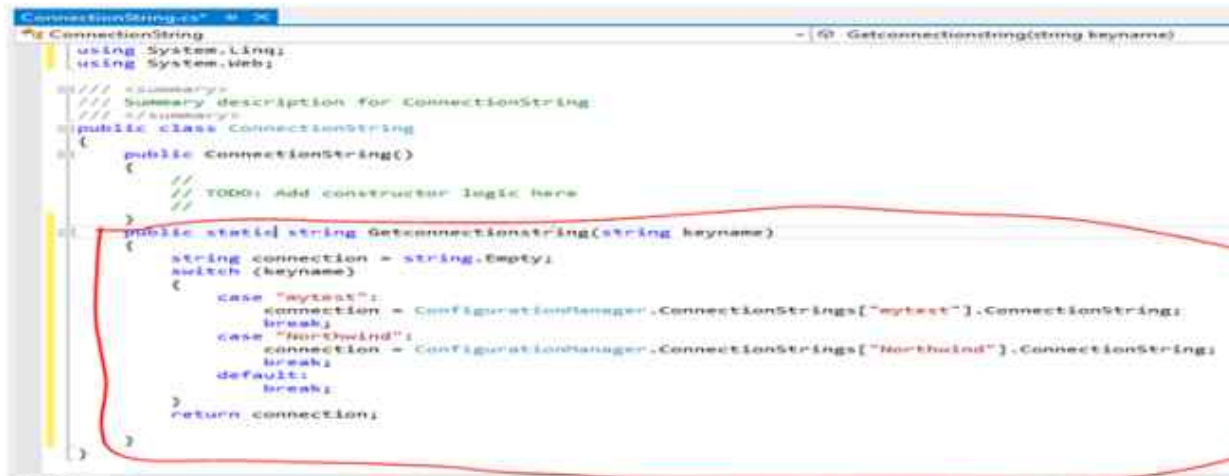
- You need to add class files to an ASP.Net web application.
- Then you need to include a namespace using System.Configuration in the class file.

```

using System.Configuration;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

```

Then write the following method in this class.



```

ConnectionString
using System.Linq;
using System.Web;

/// <summary>
/// Summary description for ConnectionString.
/// </summary>
public class ConnectionString
{
    public ConnectionString()
    {
        /// TODO: Add constructor logic here
    }

    public static string GetConnectionString(string keyname)
    {
        string connection = string.Empty;
        switch (keyname)
        {
            case "mytest":
                connection = ConfigurationManager.ConnectionStrings["mytest"].ConnectionString;
                break;
            case "Northwind":
                connection = ConfigurationManager.ConnectionStrings["Northwind"].ConnectionString;
                break;
            default:
                break;
        }
        return connection;
    }
}

```

```

01. public static string GetConnectionString(string keyname)
02. {
03.     string connection = string.Empty;
04.     switch (keyname)
05.     {
06.         case "mytest":
07.             connection = ConfigurationManager.ConnectionStrings["mytest"].Co
08.             break;
09.         case "Northwind":
10.             connection = ConfigurationManager.ConnectionStrings["Northwind"]
11.             break;
12.         default:
13.             break;
14.     }
15.     return connection;
16.
17. }

```

In the preceding I have written a method to get the connection on the basis of the keyname passed.

Then I will show you how to use the preceding method of this class in code to access connections. In the following code you can see I have used the GetConnectionString method of the ConnectionString class and passed the keyname to connect to the specific database using keyname.

```

SqlConnection con = new SqlConnection(ConnectionString.GetConnectionString("mytest"));
con.Open();
DataTable dt = new DataTable();
SqlDataAdapter ad = new SqlDataAdapter("SELECT * FROM TT", con);
ad.Fill(dt);
}

private void GetDataFromNorthwindDB()
{
    SqlConnection con = new SqlConnection(ConnectionString.GetConnectionString("Northwind"));
    con.Open();
    DataTable dt = new DataTable();
    SqlDataAdapter ad = new SqlDataAdapter("SELECT * FROM Categories", con);
    ad.Fill(dt);
}
}

```

If you got this error when reading a connection string from a web.config file then go to this link: [Click here to fix the following error](#).



Using the preceding described way you can connect to the database using this method, there is no need to write a connection code in every page.

Now we will see the next way to specify a connection.

Now I will create a common class file with connection code.

I have created a DBConnectionClass.cs file and then written the following code in this class. You can use this class file in multiple Web applications where you need to use the same database connection.

In this class I declared an enum for the list of the database. We can specify all the database names to be used across application and access them using an enum.

Then I wrote a method, GetConnectionString, with Database enum parameter.

```

{
    MyTest,
    Northwind
}

/// <summary>
/// This method return connction string
/// </summary>
/// <param name="DBName">pass Dbname using enum</param>
/// <returns></returns>
public static SqlConnection Getconnectionstring(Database DBName)
{
    SqlConnection connection=null;
    switch (DBName)
    {
        case Database.MyTest:
            connection = new SqlConnection("Data Source=.\SQLEXPRESS;Initial Catalog=mytest;Integrated Security=True");
            break;
        case Database.Northwind:
            connection = new SqlConnection("Data Source=.\SQLEXPRESS;Initial Catalog=Northwind;Integrated Security=True");
            break;
        default:
            break;
    }
    return connection;
}

```

Write the following code in your class:

```

01.  /// <summary>
02.  /// list of database name
03.  /// </summary>
04.  public enum Database
05.  {
06.      MyTest,
07.      Northwind
08.  }
09.  /// <summary>
10.  /// This method return connction string
11.  /// </summary>
12.  /// <param name="DBName">pass Dbname using enum</param>
13.  /// <returns></returns>
14.  public static SqlConnection Getconnectionstring(Database DBName)
15.  {
16.      SqlConnection connection=null;
17.      switch (DBName)
18.      {
19.          case Database.MyTest:
20.              connection = new SqlConnection("Data Source=.\SQLEXPRESS;Initi
21.              break;
22.          case Database.Northwind:
23.              connection = new SqlConnection("Data Source=.\SQLEXPRESS;Initi
24.              break;
25.          default:
26.              break;
27.      }
28.      return connection;
29.  }
30.  }

```

Then I will describe how to access this class method in a web page.

You can call this method as in

```

private void GetDataFromMytestDB()
{
    DataTable dt = new DataTable();
    SqlDataAdapter ad = new SqlDataAdapter("SELECT * FROM TT", DBConnectionClass.Getconnectionstring(DBConnectionClass.Database.MyTest));
    ad.Fill(dt);
}

private void GetDataFromNorthwindDB()
{
    DataTable dt = new DataTable();
    SqlDataAdapter ad = new SqlDataAdapter("SELECT * FROM Categories", DBConnectionClass.Getconnectionstring(DBConnectionClass.Database.Northwind));
    ad.Fill(dt);
}

```

## Specify Database connection string in Windows application

Now I will describe how to declare a connection string in an App.config file of a Windows application.

In the same way you saw in the web.config file you need to write this tag inside the app.config file.

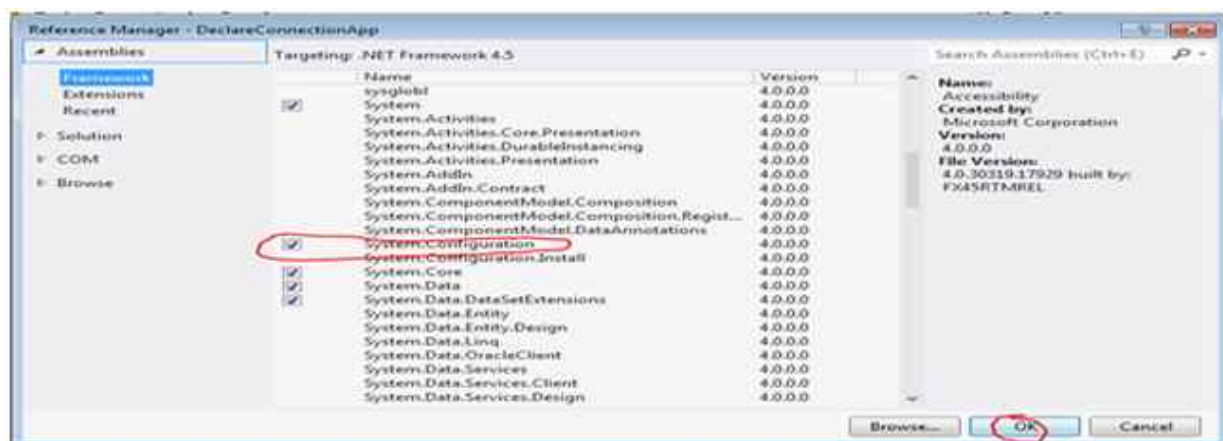
```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
  </startup>
  <connectionStrings>
    <add name="mytest" connectionString="Data Source=.\SQLEXPRESS;Initial Catalog=mytest;Integrated Security=True;" providerName="System.Data.SqlClient" />
    <add name="Northwind" connectionString="Data Source=.\SQLEXPRESS;Initial Catalog=Northwind;Integrated Security=True;" providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>

```

In the same way I already described, for the web application you can read a connection from the app.config.

But one extra thing you need to do in Windows applications is that you need to add a reference of System.Configuration to your application.



Then you need to follow the steps I already described for web applications.



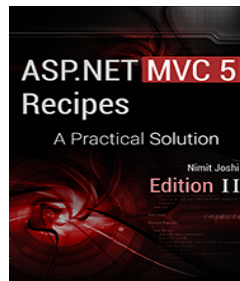
connection in the same way that I described in the Web section of this article.

[ASP.NET](#)
[Specify Connection String](#)
[Web Application](#)
[Window Application](#)

Next Recommended Reading

[Various Ways Of Form Submission In ASP.NET MVC - Part One](#)

## OUR BOOKS



Priti Kumari *TOP 500*

Software Developer

<https://dataoftech.com/>

370

4.9m

1

[View Previous Comments](#)

5

19



Type your comment here and press Enter Key (Minimum 10 characters)



Web.config should be encrypted! And using web.config across multiple project types with transforms is a major pain. Would love to see an article of how to move connect strs and app settings into a single, transformable set of files that will work across multiple project types within a single solution without the need to use crazy build scripts. Nice article. Thanks.

[Ted Krapf](#)

Mar 02, 2019

2061 3 0

0 0 Reply



But i need access more than 25 databases within one project without defining connection strings in webconfig. i pass only one connection string in webconfig and access more than 25 databases at runtime.



Just change dynamically database name from existing connection string.

[Priti Kumari](#)

370 6.1k 4.9m

Jun 22, 2018

0



Nice work Priti

[Ram Krishna Shukla](#)

1962 102 0

Jun 08, 2018

0

0 Reply



Thanks for sharing this information

[Manav Pandya](#)

115 19.9k 1.9m

Jan 18, 2017

0

0 Reply



Thanks everyone!!!

[Priti Kumari](#)

370 6.1k 4.9m

Aug 31, 2015

0

0 Reply



thanks for share..

[Ankit Bansal](#)

437 5k 1.2m

Aug 31, 2015

0

0 Reply



nice

[Hashim Shafiq](#)

1761 303 41.4k

Aug 30, 2015

0

0 Reply



Nice Share

[Sibeesh Venu](#)

12 49.4k 9.6m

Aug 30, 2015

0

0 Reply



Good article. Thanks for sharing

[Santhakumar Munuswamy](#)

43 34.4k 1.3m

Aug 29, 2015

0

0 Reply



Nice...

[Humayun Kabir Mamun](#)

396 5.6k 255

Aug 29, 2015

0

0 Reply



#### FEATURED ARTICLES

What Is The Metaverse

How To Upgrade to Windows 11

Exploring Subject <T> In Reactive Extensions For .Net

Micro Frontends With Webpack

What's New In iPhone 13

[View All](#) 

## TRENDING UP

- 01 JWT Token Authentication And Authorizations In .Net Core 6.0 Web API
- 02 New Features In Blazor With .NET 6
- 03 Implementing Smart Contract in ASP.NET Application
- 04 Build And Publish A .Net Core API To Perform CRUD Operations On Azure Cosmos DB (SQL API)
- 05 Understanding Thread Starvation in .NET Core Applications
- 06 Three Simple Ways To Get Started With Stratis Blockchain Development
- 07 Create A Bot In .NET Using Bot Framework SDK
- 08 Deploy A .NET API To Heroku Through GitHub Actions
- 09 Collection Performance - How Do You LINQ?
- 10 What Is The Difference Between Data Scientist And Data Analyst?

[View All](#) [About Us](#) [Contact Us](#) [Privacy Policy](#) [Terms](#) [Media Kit](#) [Sitemap](#) [Report a Bug](#) [FAQ](#) [Partners](#)[C# Tutorials](#) [Common Interview Questions](#) [Stories](#) [Consultants](#) [Ideas](#) [Certifications](#)

©2021 C# Corner. All contents are copyright of their authors.