ADO.NET SqlConnection Class

It is used to establish an open connection to the SQL Server database. It is a sealed class so that cannot be inherited. SqlConnection class uses SqlDataAdapter and SqlCommand classes together to increase performance when connecting to a Microsoft SQL Server database.

Connection does not close explicitly even it goes out of scope. Therefore, you must explicitly close the connection by calling Close() method.

SqlConnection Signature

```
public sealed class SqlConnection : System.Data.Common.DbConnection, ICloneable, IDisposable
```

SqlConnection Constructors

Constructors	Description
SqlConnection()	It is used to initializes a new instance of the SqlConnection class.
SqlConnection(String)0	It is used to initialize a new instance of the SqlConnection class and takes connection string as an argument.
SqlConnection(String, SqlCredential)	It is used to initialize a new instance of the SqlConnection class that takes two parameters. First is connection string and second is sql credentials.

SqlConnection Methods

Method	Description
BeginTransaction()	It is used to start a database transaction.
ChangeDatabase(String)	It is used to change the current database for an open SqlConnection.
ChangePassword(String, String)	It changes the SQL Server password for the user indicated in the connection string.
Close()	It is used to close the connection to the database.
CreateCommand()	It enlists in the specified transaction as a distributed transaction.
GetSchema()	It returns schema information for the data source of this SqlConnection.
Open()	It is used to open a database connection.
ResetStatistics()	It resets all values if statistics gathering is enabled.

SqlConnection Example

Now, let's create an example that establishes a connection to the SQL Server. We have created a **Student** database and will use it to connect. Look at the following C# code.

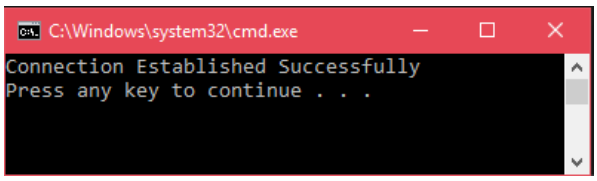
```
using (SqlConnection connection = new SqlConnection(connectionString))
{
  connection.Open();
}
```

Using block is used to close the connection automatically. We don't need to call close () method explicitly, **using** block do this for ours implicitly when the code exits the block.

// Program.cs

```
using System;
using System.Data.SqlClient;
namespace AdoNetConsoleApplication
{
   class Program
   {
      static void Main(string[] args)
      {
         new Program().Connecting();
      }
      public void Connecting()
      {
         using (
               // Creating Connection
               SqlConnection con = new SqlConnection("data source=.; database=student; integrated security=SSPI")
            )
         {
            con.Open();
            Console.WriteLine("Connection Established Successfully");
         }
      }
   }
}
```

Output:



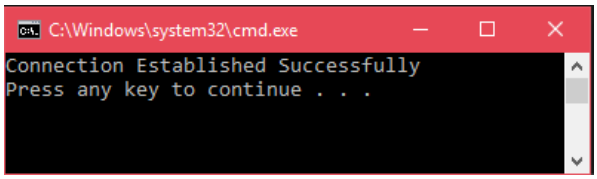
What, if we don't use **using** block.

If we don't use using block to create connection, we have to close connection explicitly. In the following example, we are using try-block instead of using block.

// Program.cs

```
using System;
using System.Data.SqlClient;
namespace AdoNetConsoleApplication
{
   class Program
   {
      static void Main(string[] args)
      {
         new Program().Connecting();
      }
      public void Connecting()
      {
         SqlConnection con = null;
         try
         {
            // Creating Connection
            con = new SqlConnection("data source=.; database=student; integrated security=SSPI");
            con.Open();
            Console.WriteLine("Connection Established Successfully");
         }
         catch (Exception e)
         {
            Console.WriteLine("OOPs, something went wrong.\n"+e);
         }
         finally
         {  // Closing the connection
            con.Close();
         }
      }
   }
}
```

Output:



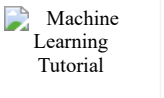
← prev next →

Help Others, Please Share

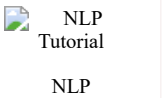
Join Javatpoint Test Series

Placement Papers	AMCAT	Bank PO/Clerk	GATE
TCS	eLitmas	UPSSSC	NEET
HCL	Java	Government Exams	CAT
Infosys	Python	SSC	Railway
IBM	C Programming	Civil Services	CTET
Accenture	Networking	SBI	IIT JEE

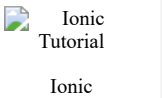
Learn Latest Tutorials



ML

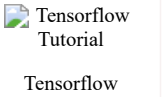


NLP

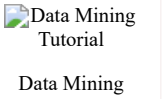


Ionic

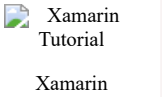
VHDL



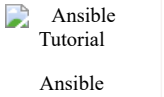
Tensorflow



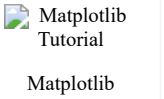
Data Mining



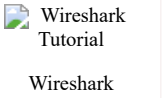
Xamarin



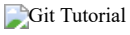
Ansible



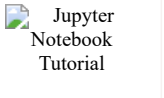
Matplotlib



Wireshark

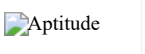


Git

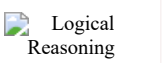


Jupyter

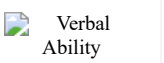
Preparation



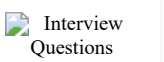
Aptitude



Reasoning



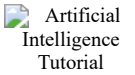
Verbal A.



Interview

Company

Trending Technologies


Artificial Intelligence Tutorial

AI


AWS Tutorial

AWS


Selenium tutorial

Selenium


IoT Tutorial

IoT


Cloud tutorial

Cloud


Hadoop tutorial

Hadoop


ReactJS Tutorial

ReactJS


React Native Tutorial

React Native


Node.js tutorial

Node.js


Data Science Tutorial

D. Science


Angular 7 Tutorial

Angular 7


Blockchain Tutorial

Blockchain

B.Tech / MCA


DBMS tutorial

DBMS


Data Structures tutorial

DS


DAA tutorial

DAA


Operating System tutorial

OS


Computer Network tutorial

C. Network


Compiler Design tutorial

Compiler D.


Computer Organization and Architecture

COA


Discrete Mathematics Tutorial

D. Math.


Ethical Hacking Tutorial

E. Hacking


Computer Graphics Tutorial

C. Graphics


Software Engineering Tutorial

Software E.


html tutorial

Web Tech.


Cyber Security tutorial

Cyber Sec.


Automata Tutorial

Automata


C Language tutorial

C


C++ tutorial

C++


Java tutorial

Java


.Net Framework tutorial

.Net


Python tutorial

Python


List of Programs

Programs


Control Systems tutorial

Control S.