

What is SqlCommand Objects in ADO.NET

Published 7/12/2009 by [Vijendra Shakya](#)

To execute the SQL Query or Stored procedure, we use the SqlCommand object.

We use the SqlCommand object as follows:

// create the SqlCommand object

```
String sqlQuery="Select * from User";  
SqlCommand cmd = new SqlCommand (sqlQuery, con);
```

another way is:

```
SqlCommand cmd = new SqlCommand ();  
cmd.CommandText = "SELECT Count(*) from users";  
cmd.Connection = con;  
cmd.CommandType = CommandType.Text;
```

Here con is the object of the SqlConnection.

We have three basic methods to execute the Sql Query.

1. **ExecuteScalar ()**: Execute the query and returns the single value (first column of the first row in the result set). It is very useful with aggregate functions like Count (*) or Sum () etc.

Example:

```
SqlCommand cmd = new SqlCommand ();  
cmd.CommandText = "SELECT Count(*) from users";  
cmd.Connection = con;  
cmd.CommandType = CommandType.Text;  
int count= Convert.ToInt32(cmd.ExecuteScalar());
```

this will return only single value.

2. **ExecuteReader ()**: Executes the query, and returns a SqlDataReader object. SqlDataReader is a forward-only, read-only.

Example:

```
SqlCommand cmd = new SqlCommand ();
```

```
cmd.CommandText = "Select * from User";  
cmd.Connection = con;  
cmd.CommandType = CommandType.Text;  
SqlDataReader dr= cmd.ExecuteReader();
```

3. **ExecuteNonQuery ()**: Executes the query, and does not return any result set. It is generally used for Update, Insert and Delete command.

Example:

```
SqlCommand cmd = new SqlCommand ();  
cmd.CommandText = "Delete from User where userId=@UserId ";  
cmd.Connection = con;  
cmd.CommandType = CommandType.Text;  
cmd.ExecuteNonQuery();
```

tags : ado.net, c#, ExecuteNonQuery, ExecuteReader, ExecuteScalar, Sql, SqlCommand