

Use Stack Overflow for Teams at work to find answers in a private and secure environment. Get your first 10 users free. [Sign up.](#)

How to view LINQ Generated SQL statements?

Asked 8 years, 7 months ago Active 1 year, 2 months ago Viewed 56k times



How is it done using the ObjectQuery method?

52

linq



20

asked Feb 4 '11 at 15:53



[nellbryant](#)

1,556 4 15 16

8 Answers



55

You can always attach something to the .Log property of your DataContext. That will show all the SQL commands as they are sent.

I do this in my base for data access objects and output it to the Visual Studio debug console. As the objects create their DataContext I check it see if its debug and attach a TextWriter helper class like this:

```
dbDataContext _dB = new dbDataContext();
_dB.CommandTimeout = 5000;

#if DEBUG
    _dB.Log = new DebugTextWriter();
#endif
```

Here is the helper object for output to the debug console:

```
//utility class for output of TextWriter for the Visual Studio Debug window
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```

    System.Diagnostics.Debug.Write(new String(buffer, index, count));
}

public override void Write(string value)
{
    System.Diagnostics.Debug.Write(value);
}

public override Encoding Encoding
{
    get { return System.Text.Encoding.Default; }
}
}

```

edited Dec 23 '13 at 14:45



Chris Schiffhauer

13.5k 14 67 83

answered Feb 4 '11 at 16:28



Tj Kellie

5,345 2 25 37

5 For those using **Entity Framework 6** check [this](#), hope helps someone. – [stom](#) Feb 1 '16 at 8:39

Here is what I found using ObjectQuery Method. Using console for testing, you can do the following:

26 Create an Extension Method as below, then call it. Say Product product, then SQL prints out as product.ToTraceString.

```

public static class MyExtensions
{
    public static string ToTraceString<T>(this IQueryable<T> t)
    {
        string sql = "";
        ObjectQuery<T> oqt = t as ObjectQuery<T>;
        if (oqt != null)
            sql = oqt.ToTraceString();
        return sql;
    }
}

```

edited Oct 29 '13 at 9:42

answered Feb 4 '11 at 16:39

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

5 It can be used in VS Watch window too: `((ObjectQuery)myQuery).ToTraceString()` . Nice one! – [icebat](#) Mar 21 '13 at 9:51

What about aggregate methods like `Count` ? – [Mat J](#) May 26 '14 at 6:30

6 I don't think works when using `dbContext` as you can't cast `IQueryable<T>` to `ObjectQuery<T>` in this case. – [NER1808](#) Oct 21 '14 at 13:58

You can use [LINQPad](#) for that.

20

answered Feb 4 '11 at 15:56



[Olav Haugen](#)

1,021 2 10 20

They say its using the `ObjectQuery` Method or `SQL Profiler`. How do it programmatically in a console for example? – [nellbryant](#) Feb 4 '11 at 16:05

1 I'm not sure if I understood that last question... At least, to answer the question in your title, "How to view LINQ Generated SQL statements": `LINQPad` has an output window where you can select either "Results", "Lambda" or "SQL". – [Olav Haugen](#) Feb 4 '11 at 16:13 ✎

Hey, that's a nice open source tool!. Thank you. – [nellbryant](#) Feb 4 '11 at 16:33

+1 for `LinqPad` recommendation. This should be on every .Net developer's list of essential tools. – [Doctor Jones](#) Feb 4 '11 at 16:35

I was already using `LinqPad` when I searched for the answer to this question. I had completely missed the 'SQL' button in the Results pane. Thanks – [Greg Woods](#) Mar 29 '12 at 14:20

```
var q = from img in context.Images
```

```
    select img;
```

```
string sql = q.ToString();
```

`sql` will contain the sql select query.

EDIT: disadvantage: parameters won't have any values at this time

edited Jan 26 '15 at 12:31

answered Dec 23 '13 at 14:39



[avozu kurdor](#)

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

1 If you're debugging locally and using SQL Server, run the SQL Profiler. The query with all parameters will show up there. – Rich Aug 11 '17 at 21:24

You could have a look at the [Linq-to-SQL Debug Visualizer](#), or just hover your mouse over your Linq-to-SQL query (tooltip should show generated SQL), or access:

15

```
context.GetCommand(query).CommandText
```

answered Feb 4 '11 at 16:16



[marc_s](#)

605k

137

1158

1291

I haven't tried that GetCommand yet. I like the link from Scott Gu! Is that Visualizer built-in Visual Studio 2010? – nellbryant Feb 4 '11 at 16:31

@nellbryant: no, it's still a separate extension that you need to grab from the VS gallery – marc_s Feb 4 '11 at 16:33

You could run the SQL Server Profiler.

5

answered May 31 '13 at 15:57



[guiomie](#)

2,557

5

32

58

1 If you use SQL profiler, to ensure capture of all LINQ queries from your VS debug session, use SQL Profiler 'RPC Completed' and 'SQL:BatchCompleted' events. – lomm1 Sep 24 '18 at 9:11

This is what I use when setting up the database context:

1

```
this.DbContext.Database.Log += s => Debug.WriteLine(s);
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



just a small update you can now use an Action to log the SQL:

0

```
// test SQL Logger  
Action<string> SQLLogger = (message) => System.Diagnostics.Debug.Write(message);  
_dB.Context().Database.Log = SQLLogger;
```



answered Sep 20 '17 at 10:04



[liminal18](#)

373 3 16