

# Easy way to convert exec sp\_executesql to a normal query?



27



10

When dealing with debugging queries using Profiler and SSMS, its pretty common for me to copy a query from Profiler and test them in SSMS. Because I use parameterized sql, my queries are all sent as exec sp\_executesql queries.

```
exec sp_executesql
N'/*some query here*/',
N'@someParameter tinyint',
@ someParameter =2
```

I'll take this and convert it into a normal query for ease of editing (intellisense, error checking, line numbers, etc):

```
DECLARE @someParameter tinyint
SET @someParameter = 2
```

```
/*some query here*/
```

Of course, the bigger and more complex the query, the harder to do this. And when you're going back and forth multiple times, it can be a pain in the ass and soak up lots of time.

Is there an easy (e.g., macro command) way to convert muh executesql into something more convenient?

[sql-server](#) [macros](#) [format](#) [ssms](#) [sp-executesql](#)

edited Nov 4 '16 at 9:53



Alex Vazhev

1,003 1 10 14

asked Jul 16 '09 at 15:23



Will

114k 47 288 375

+1 I'd love that as well – [Vinko Vrsalovic](#) Jul 16 '09 at 15:33

+1 Seconded on lovin' this – [Adam Caviness](#) Aug 24 '11 at 13:17

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook





1

I am not aware of an existing Add-In that can do this. But you could create one :)

A few regular expressions and some string concatenation and after that sell it to Vinko and other souls looking for this functionality.



If you're feeling like diving into this, here is some information on creating an SSMS addin:

<http://sqlblogcasts.com/blogs/jonsayce/archive/2008/01/15/building-a-sql-server-management-studio-addin.aspx>



answered Jul 18 '09 at 18:56



Zyphrax

13.7k 9 57 79



37

I spent a little time making an simple script that did this for me. It's a WIP, but I stuck a (very ugly) webpage in front of it and it's now hosted here if you want to try it:

<http://execsqlformat.herokuapp.com/>



Sample input:

```
exec sp_executesql
    N'SELECT * FROM AdventureWorks.HumanResources.Employee
    WHERE ManagerID = @level',
    N'@level tinyint',
    @level = 109;
```

And the output:

```
BEGIN
DECLARE @level tinyint;

SET @level = 109;

SELECT * FROM AdventureWorks.HumanResources.Employee
    WHERE ManagerID = @level
```

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook



edited Jul 7 '13 at 20:38

answered Dec 20 '11 at 20:42



Jarrod Dixon ♦

13.7k 9 54 70



Matt Roberts

12.6k 26 85 149

Hi Matt, I am very interested in how you used the api to convert sp\_executesql, can you share your code? Thanks! – wangzq Nov 19 '12 at 13:41

2 Bunged it into a git repo for you to enjoy ;) [github.com/mattwoberts/execsqliformat](https://github.com/mattwoberts/execsqliformat) – Matt Roberts Nov 20 '12 at 20:46

Thanks! It is good to know you are using regex to parse the input first. – wangzq Nov 22 '12 at 20:59

1 [github.com/wangzq/convert-sp\\_executesql/blob/master/...](https://github.com/wangzq/convert-sp_executesql/blob/master/...) – wangzq Nov 22 '12 at 22:07

1 Not to look a gift-horse in the mouth, but the "clipboard" button copies the text with line-feeds, instead of CR/LF, which makes it impossible to paste into anything on Windows, even Notepad++. User has to copy select the HTML and copy. It would have been better just to make that control an HTML text area so that you could Ctrl+A to select-all then copy. Not that I'm complaining or anything. :) – zumalifeguard Sep 4 '14 at 19:17

I was looking for something similar so I use this in LinqPad, just copy sp\_executesql statement to the clipboard and run the code in LinqPad. It outputs the SQL statement.

5

```
void Main()
{
    ConvertSql(System.Windows.Forms.Clipboard.GetText()).Dump();
}

private static string ConvertSql(string origSql)
{
    string tmp = origSql.Replace("'", "~");
    string baseSql;
    string paramTypes;
    string paramData = "";
    int i0 = tmp.IndexOf("'") + 1;
    int i1 = tmp.IndexOf("'", i0);
    if (i1 > 0)
    {
        baseSql = tmp.Substring(i0, i1 - i0);
        i0 = tmp.IndexOf("'", i1 + 1);
        i1 = tmp.IndexOf("'", i0 + 1);
        if (i0 > 0 && i1 > 0)
        {
            paramTypes = tmp.Substring(i0, i1 - i0);
            paramData = tmp.Substring(i1 + 1, tmp.Length - i1 - 1);
        }
    }
    return baseSql + " " + paramTypes + " " + paramData;
}
```

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook



```
else
{
    throw new Exception("Cannot identify SQL statement in first parameter");
}

baseSql = baseSql.Replace("~~", "");
if (!String.IsNullOrEmpty(paramData))
{
    string[] paramList = paramData.Split(",".ToCharArray());
    foreach (string paramValue in paramList)
    {
        int iEq = paramValue.IndexOf("=");
        if (iEq < 0)
            continue;
        string pName = paramValue.Substring(0, iEq).Trim();
        string pVal = paramValue.Substring(iEq + 1).Trim();
        baseSql = baseSql.ReplaceWholeWord(pName, pVal);
    }
}

return baseSql;
}

public static class StringExtensionsMethods
{
    /// <summary>
    /// Replaces the whole word.
    /// </summary>
    /// <param name="s">The s.</param>
    /// <param name="word">The word.</param>
    /// <param name="replacement">The replacement.</param>
    /// <returns>String.</returns>
    public static String ReplaceWholeWord(this String s, String word, String replacement)
    {
        var firstLetter = word[0];
        var sb = new StringBuilder();
        var previousWasLetterOrDigit = false;
        var i = 0;
        while (i < s.Length - word.Length + 1)
        {
            var wordFound = false;
            var c = s[i];
            if (c == firstLetter)
```

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook



```
{
    if (Char.IsLetterOrDigit(s[i + word.Length]))
        wholeWordFound = false;
}

sb.Append(wholeWordFound ? replacement : word);

i += word.Length;
}

if (wordFound) continue;

previousWasLetterOrDigit = Char.IsLetterOrDigit(c);
sb.Append(c);
i++;
}

if (s.Length - i > 0)
    sb.Append(s.Substring(i));

return sb.ToString();
}
}
```

answered May 19 '15 at 11:23



Toby

73 1 4

I spent a little time and created a small modification of Matt Roberts / Wangzq solutions without DECLAREs section, you can try it on [.NET Fiddle](#) or [download LINQPad 5 file](#).

5

Input:

```
exec sp_executesql N'UPDATE MyTable SET [Field1] = @@0, [Field2] = @1',N'@@0 nvarchar(max)
,@1 int',@@0=N'String',@1=0
```

Output:

Join Stack Overflow to learn, share knowledge, and build your career.

[Sign up with email](#)[Sign up with Google](#)[Sign up with Facebook](#)

```

using System;
using System.Linq;
using System.Text.RegularExpressions;

public class Program
{
    public static void Main()
    {
        var sql = @"exec sp_executesql N'UPDATE MyTable SET [Field1] = @0, [Field2] =
@1',N'@0 nvarchar(max) ,@1 int',@0=N'String',@1=0";
        Console.WriteLine(ConvertSql(sql));
    }

    public static string ConvertSql(string origSql)
    {
        var re = new Regex(@"exec*\s*sp_executesql\s+N'([\s\S]*)',\s*N'(@[\s\S]*?)',\s*
([\s\S]*)", RegexOptions.IgnoreCase); // 1: the sql, 2: the declare, 3: the setting
        var match = re.Match(origSql);
        if (match.Success)
        {
            var sql = match.Groups[1].Value.Replace("'", "");
            //var declare = match.Groups[2].Value;
            var setting = match.Groups[3].Value + ',';

            // to deal with comma or single quote in variable values, we can use the
            variable name to split
            var re2 = new Regex(@"@[^\s,]*?\s*=");
            var variables = re2.Matches(setting).Cast<Match>().Select(m =>
m.Value).ToArray();
            var values =
re2.Split(setting).Where(s=>!string.IsNullOrEmpty(s)).Select(m =>
m.Trim(',').Trim().Trim(';')).ToArray();

            for (int i = variables.Length-1; i>=0; i--)
            {
                sql = Regex.Replace(sql, "(" + variables[i].Replace("=", "")+")",
values[i], RegexOptions.Singleline);
            }
            return sql;
        }

        return @"Unknown sql query format.";
    }
}

```

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook





477 6 15



1,003 1 10 14

Another solution which replaces the parameter values directly in the query (not exactly what you asked for but it might prove useful to others):

3

<https://code.msdn.microsoft.com/windowsdesktop/spExecuteSql-parser-1a9cd7bc>

It goes from:

```
exec sp_executesql N'UPDATE Task SET Status = @p0, Updated = @p1 WHERE Id = @p2 AND
Status = @p3 AND Updated = @p4',N'@p0 int,@p1 datetime,@p2 int,@p3 int,@p4
datetime',@p0=1,@p1='2015-02-07 21:36:30.313',@p2=173990,@p3=2,@p4='2015-02-07
21:35:32.830'
```

to:

```
UPDATE Task SET Status = 1, Updated = '2015-02-07 21:36:30.313' WHERE Id = 173990 AND
Status = 2 AND Updated = '2015-02-07 21:35:32.830'
```

which makes it easier to understand.

The console application on that page can be used by passing a file parameter or copying the `sp_executesql` in the clipboard, running the app and then pasting the resulting SQL from the clipboard.

### Update:

An SQL formatter can also be added to that solution for easier readability:

<http://www.nuget.org/packages/PoorMansTSQLFormatter/>

```
newSql = ConvertSql(Clipboard.GetText());
var formattedSql = SqlFormattingManager.DefaultFormat(newSql);
Clipboard.SetText(formattedSql);
```

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook



Sql Prompt got this feature recently (2017-02-06). Select the text and look for "Inline EXEC" in the context menu. Gotta love Prompt :)

2

answered Feb 6 '17 at 8:48



buckley

10.2k 2 40 52

1

I faced with this problem too and wrote simple application for solving it - [ClipboardSqlFormatter](#). This is a tray application that listens clipboard input events and tries to detect and convert dynamic sql to static sql.

Anything you need is to copy dynamic sql (from sql profiler for example) and paste to text editor - pasted sql will be a static sql :)

For example, if copied sql is:

```
exec sp_executesql N' SELECT "obj"."CreateDateTime", "obj"."LastEditDateTime" FROM LDERC
"doc" INNER JOIN LDOBJECT "obj" ON ("doc"."ID" = "obj"."ID") LEFT OUTER JOIN LDJournal
"ContainerID.jrn" ON ("doc"."JournalID" = "ContainerID.jrn"."ID") WHERE ( "doc"."ID"
= @V0 AND ( "doc"."StateID" <> 5 AND "ContainerID.jrn"."Name" <> 'Hidden journal'
) ) ',N'@V0 bigint',@V0=6815463'
```

then pasted sql will be:

```
SELECT "obj"."CreateDateTime"
      ,"obj"."LastEditDateTime"
FROM LDERC "doc"
INNER JOIN LDOBJECT "obj" ON ("doc"."ID" = "obj"."ID")
LEFT OUTER JOIN LDJournal "ContainerID.jrn" ON ("doc"."JournalID" = "ContainerID.jrn"."ID")
WHERE (
    "doc"."ID" = 6815463
    AND (
        "doc"."StateID" <> 5
        AND "ContainerID.jrn"."Name" <> 'Hidden journal'
    )
)
```

Join Stack Overflow to learn, share knowledge, and build your career.

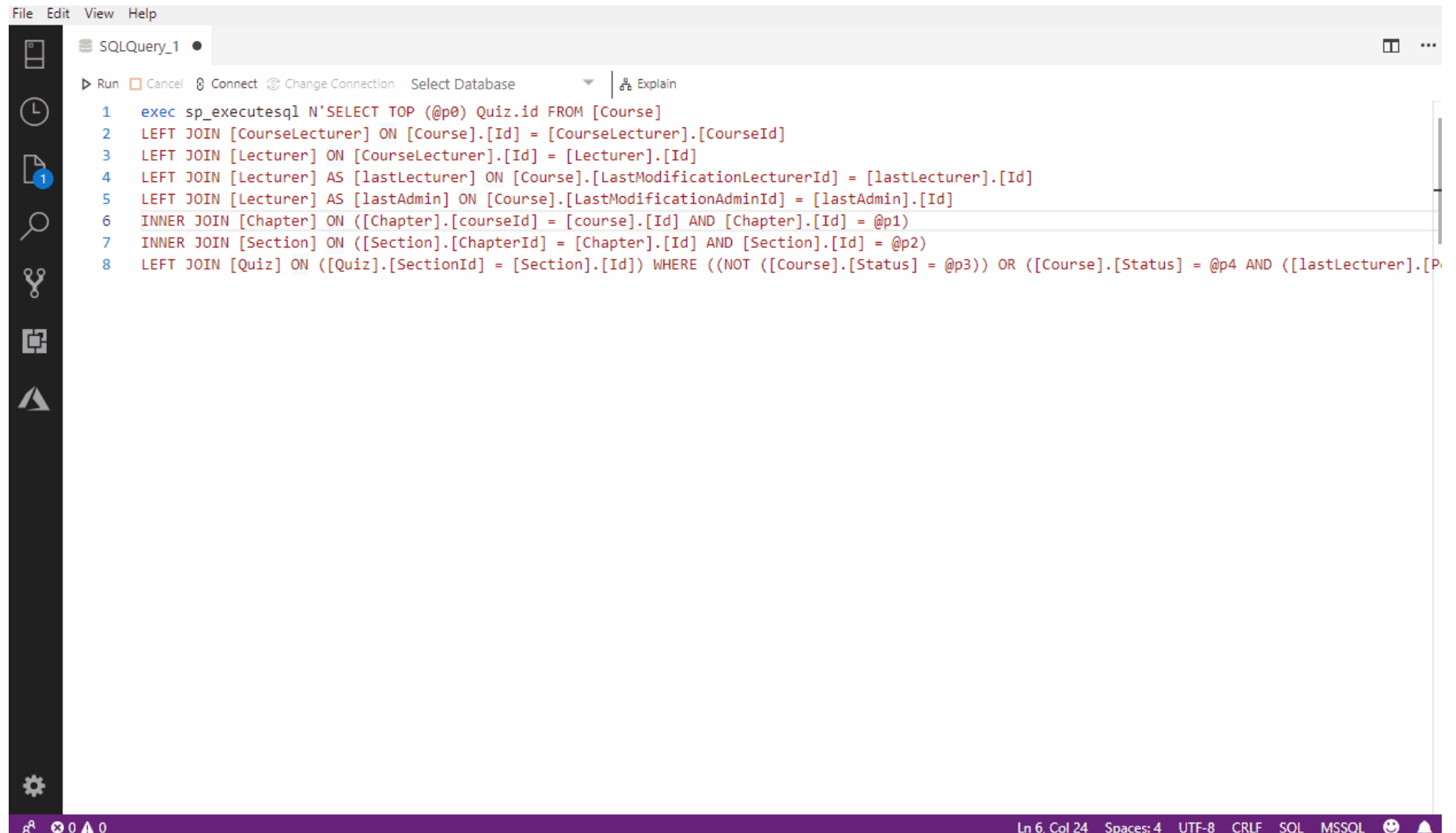
[Sign up with email](#)[Sign up with Google](#)[Sign up with Facebook](#)





You can use this Azur data studio extension. it based on @Matt Roberts repo. <https://github.com/PejmanNik/sqlops-spexecutesql-to-sql/releases/tag/0.0.1>

1



```
1 exec sp_executesql N'SELECT TOP (@p0) Quiz.id FROM [Course]
2 LEFT JOIN [CourseLecturer] ON [Course].[Id] = [CourseLecturer].[CourseId]
3 LEFT JOIN [Lecturer] ON [CourseLecturer].[Id] = [Lecturer].[Id]
4 LEFT JOIN [Lecturer] AS [lastLecturer] ON [Course].[LastModificationLecturerId] = [lastLecturer].[Id]
5 LEFT JOIN [Lecturer] AS [lastAdmin] ON [Course].[LastModificationAdminId] = [lastAdmin].[Id]
6 INNER JOIN [Chapter] ON ([Chapter].[courseId] = [course].[Id] AND [Chapter].[Id] = @p1)
7 INNER JOIN [Section] ON ([Section].[ChapterId] = [Chapter].[Id] AND [Section].[Id] = @p2)
8 LEFT JOIN [Quiz] ON ([Quiz].[SectionId] = [Section].[Id]) WHERE ((NOT ([Course].[Status] = @p3)) OR ([Course].[Status] = @p4 AND ([lastLecturer].[P
```

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook



Conclusion: I note this still gets a little attention so I'll add details here for what my eventual solution was.

0

It turns out that nothing beats doing it for yourself. I created a simple console app that parsed my stored procedure and spit out what I wanted. By adding it to the list of external tools, and passing the current filename as an argument, I could use the following to strip out and rearrange what I needed.

In use, I'd add a new sql file, paste in the sql, save it, then run the external tool. After it completes, the IDE asks me to reload the file. Poof, no more stored procedure.

I do note that this may not work with *every* executesql statement, so you'll have to modify if it does not meet your needs.

```
class Program
{
    const string query = "query";
    const string decls = "decls";
    const string sets = "sets";
    static void Main(string[] args)
    {
        try
        {
            var text = File.ReadAllText(args[0]);
            if(string.IsNullOrEmpty(text))
            {
                Console.WriteLine("File is empty. Try saving it before using the
hillbilly sparc decoder");
            }
            var regex = new Regex(@"exec sp_executesql N'(?<" + query + ">.*')',N'(?<" +
decls + ">[^\']*')',(?<" + sets + ">.*')", RegexOptions.Singleline);
            var match = regex.Match(text);

            if(!match.Success || match.Groups.Count != 4)
            {
                Console.WriteLine("Didn't capture that one. Shit.");
                Console.Read();
                return;
            }

            var sb = new StringBuilder();
```

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook



```
        File.WriteAllText(args[0], sb.ToString());
    }
    catch (Exception ex)
    {
        Console.WriteLine("S*t blew up, yo");
        Console.WriteLine(ex.ToString());
        Console.WriteLine("Press a key to exit");
        Console.Read();
    }
}
```

[edited Mar 20 at 18:04](#)

community wiki  
[4 revs, 2 users 97%](#)  
[Will](#)

---

Jeez, @marc, do you hate sprocs that much? – [Will](#) Feb 21 '17 at 21:11

---

Yes, Will, I do. – [marc\\_s](#) Feb 21 '17 at 21:34

---

---

**Join Stack Overflow** to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook



---

**Join Stack Overflow** to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook

