### **Data Developer Center**

Search Data with Bing

United States (English) ▼Sign in

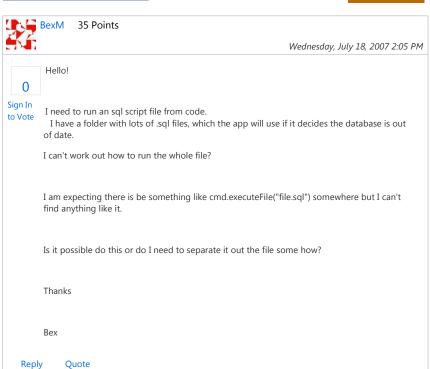
Home Library Learn Downloads Support Community Forums

Data Developer Center > Data Platform Development Forums > ADO.NET Managed Providers > Run a .sql script file in C#

# Run a .sql script file in C#

Search ADO.NET Managed Providers





### Answers

boban.s Datasoft (Partner)

Wednesday, July 18, 2007 2:42 PM You can't run files directly, with just specifing file name. You must first read it's content in

6,605 Points

17 Sian In to Vote

a string and then execute CommandText command. But this will work as long you don't have GO statement in your sql script. From your post

it looks like that SqlCommand is not usefull because scripts that include DDL command are completed with batch finalizer GO command. Luckily, since SQL Server 2005, you can use smo library to do this operation and not use batch command and osql connection as previously. Here is how to do it: using System.Data.SqlClient;

```
using System.IO;
using Microsoft.SqlServer.Management.Common;
using Microsoft.SqlServer.Management.Smo;
namespace ConsoleApplication1
{
   class Program
        static void Main(string[] args)
           string sqlConnectionString = "Data Source=(local);Initial
Catalog=AdventureWorks;Integrated Security=True";
           FileInfo file = new FileInfo("C:\\myscript.sql");
           string script = file.OpenText().ReadToEnd();
           SqlConnection conn = new SqlConnection(sqlConnectionString);
           Server server = new Server(new ServerConnection(conn));
            server.ConnectionContext.ExecuteNonQuery(script);
```

My Forum Links

Sign In To Forums Forums Home **Browse Forums Users** 

#### **Related Topics**

= Unanswered = Answered

how to run multiple .sql file c# - Microsoft Home Page | Devices ...

FAQ Item: How to run a .sql script file from C#?

How to run a .sql script file? - Microsoft Home Page | Devices and ...

How to run a sql script file on Sql server Express edition 2005 ...

Executing sql script using C# - Microsoft Home Page | Devices and ...

Change .Sql File Script format to Execute using

How to execute a sql script from C# - Microsoft Home Page ..

run a sql script file save the result in a text file and email it

 $\mbox{C\#}$  - How can I Execute a complex SQL file and getting the results?

How to run Oracle Script file using C#. -

Microsoft Home Page ...

# Statistics

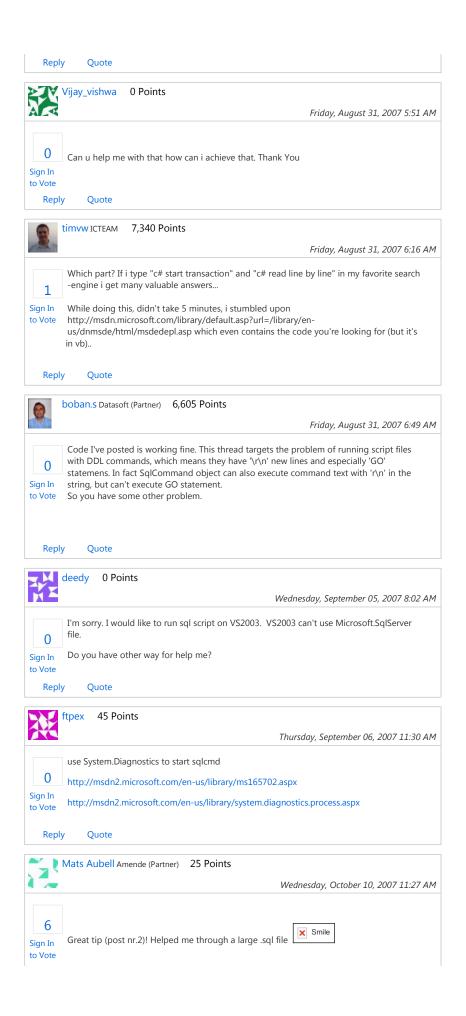
Started: 7/18/2007 Last Reply: 8/27/2012 Helpful Votes: 31 Replies: 51

Views: 121,499

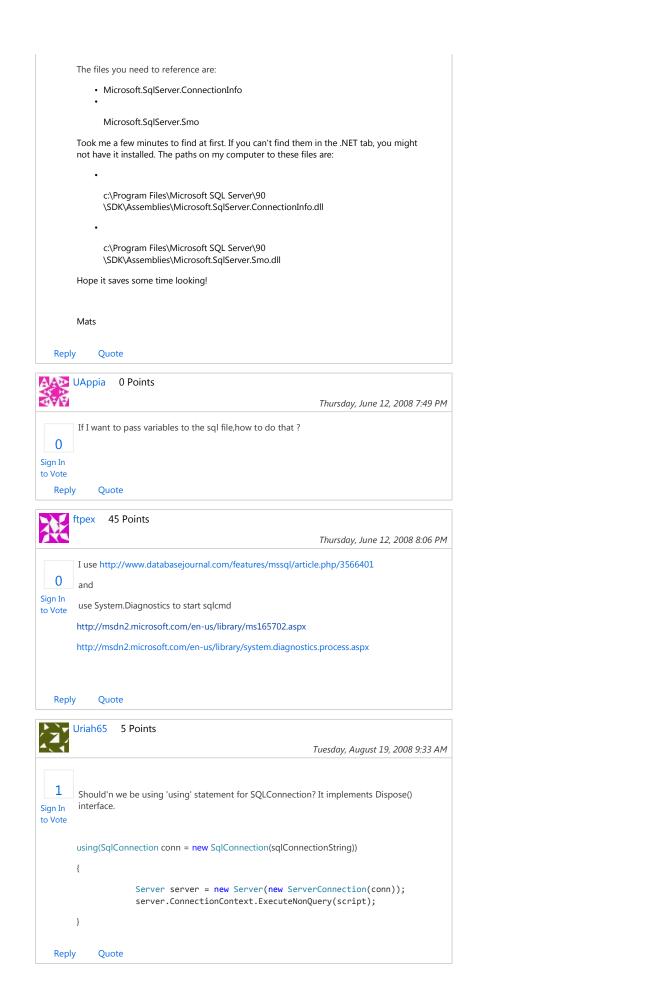
Frequently asked questions - Forums help

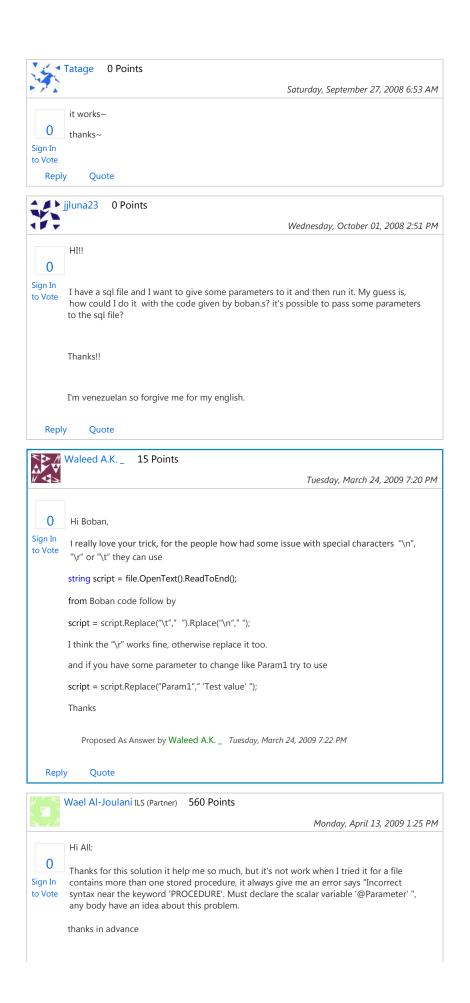
```
Reply
           Quote
```



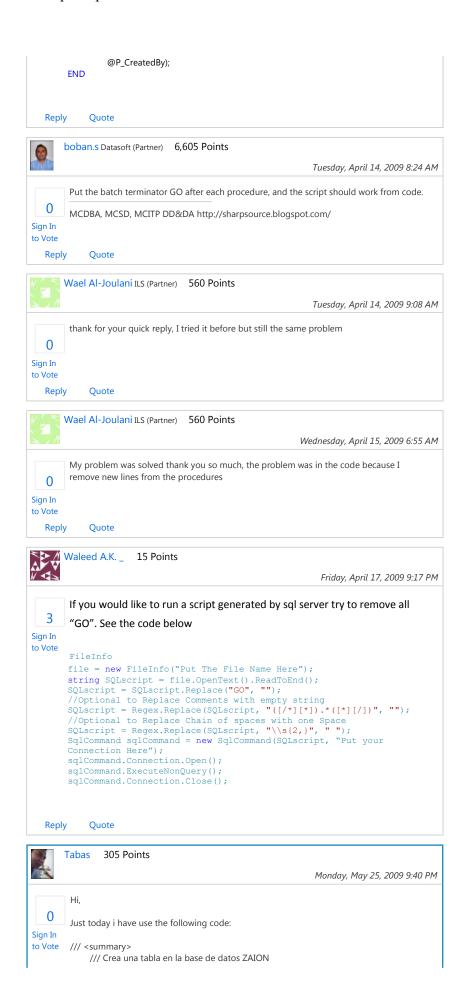


©

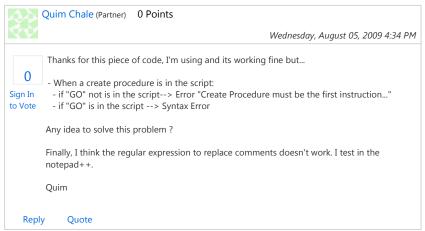


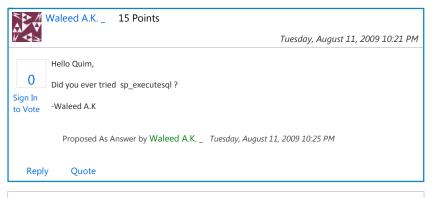


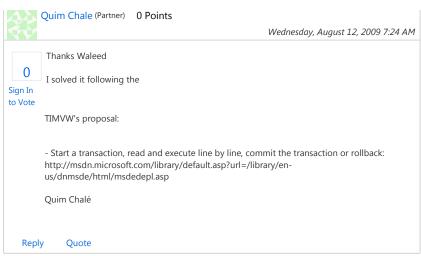




```
/// </summary>
          /// <param name="script">El script SQL generado por Zaion</param>
          /// <param name="conexion">La cadena de conexión a usar</param>
          public static void CreaTablaZaion(string script, string conexion)
            Microsoft. Practices. Enterprise Library. Data. Database \_db =
     DatabaseFactory.CreateDatabase(conexion);
            DbConnection _cmd = _db.CreateConnection();
            using (SqlConnection conn = new SqlConnection(_cmd.ConnectionString))
               try
                 Server server = new Server(new ServerConnection(conn));
                 server.ConnectionContext.ExecuteNonQuery(script);
              catch (Exception ex)
                  _lasterror = ex.Message;
                 throw new ApplicationException(String.Format("Ocurrió un error al intentar
     crear el esquema de plantilla.\r\nMás detalles:\r\n{0}", _lasterror));
          }
     You will need to add the following references:
     using System.Data.SqlClient;
     using Microsoft.SqlServer.Management.Smo;
     using Microsoft.SqlServer.Management.Common;
     using Microsoft.Practices.EnterpriseLibrary.Data;
     using System.Data.Common;
     Regards
     TESTED
        Proposed As Answer by Tabas Monday, May 25, 2009 9:40 PM
Reply
          Quote
```









```
IsItScienceFiction.Tests.UserTest.TestMethod1() in C:\Users\J. Pablo
     Fernández\Documents\Visual Studio 2008
     \Projects\IsItScienceFiction\IsItScienceFiction.Tests\UserTest.cs: line 75
     I am running Visual Studio 2008 with SQL Server 2008. What am I doing wrong?
         Proposed As Answer by 205ger Monday, September 21, 2009 1:29 PM
Reply
           Quote
```

```
205ger
                   0 Points
                                                                Monday, September 21, 2009 1:30 PM
          Hi See this solution: Console
   0
Sign In
               static void Main(string[] args)
to Vote
                 Console.WriteLine("1-Hola");
                Console.WriteLine("2-Genera Conextion String");
                string sqlConnectionString = @"....'
                SqlConnection cn = new SqlConnection(sqlConnectionString);
                Console.WriteLine("3-Verificando esta de conexion");
                    if (cn.State==ConnectionState.Closed)
                      Console.WriteLine("4-Por abrir conexion");
                     cn.Open():
                     Console.WriteLine("5-Conxion Abierta");
                 SqlCommand cmd;
                 FileInfo file = new FileInfo("C:\\apolo2.sql");
                 Console.WriteLine("6-Por leer archivo");
                 string sql = file.OpenText().ReadToEnd();
                 Console.WriteLine("7-archivo ya leido y paso a reemplazo GO");
                 string scriptText = sql;
                 //split the script on "GO" commands
                 string[] splitter = new string[] { "\r \nGO\r \n" };
                 string[] commandTexts = scriptText.Split(splitter,
                  StringSplitOptions.RemoveEmptyEntries);
                 int i=0:
                   Console.WriteLine("8-Actualizando");
                 foreach (string commandText in commandTexts)
                   if (cn.State == ConnectionState.Closed)
                      cn.Open();
                   string x = commandText.Replace("GO", "");
                   cmd = new SqlCommand(x, cn);
                   try
                      Console.Write(i.ToString() + "-");
                      cmd.ExecuteNonQuery();
                   catch (SqlException sqlexception)
                      Console.WriteLine(sqlexception.Message);
                   catch (Exception ex)
                      Console.WriteLine(ex.Message);
                   finally
```

```
cn.Close();
                  //execute commandText
            Console.WriteLine("10-Cerrada la base OK");
            Console.ReadKey();
            Proposed As Answer by 205ger Monday, September 21, 2009 1:45 PM
   Reply
              Quote
                 0 Points
                                                               Saturday, October 03, 2009 7:44 AM
         Cool! But how to carry out this on production server?
   0
         using Microsoft.SqlServer.Management.Common;
Sign In
        using Microsoft.SqlServer.Management.Smo;
to Vote
         are not supported by webforms..
              Quote
   Reply
        Nitin J. Jain BatchMaster Software Inc... 1,710 Points
                                                            Wednesday, October 07, 2009 12:07 PM
   0
         I also have the same requirement i.e. I have to create database and all table using the C#
Sign In
        application. I have all command in a
to Vote
         single .sql file. Currently I am using sqlCommand.ExecuteNonQuery() to perform execute
        sql commands. But as you mention another
        way to do the same. Can you please help me which one is better in context of Application
        performance etc?
   Reply
              Quote
        torohm
                  0 Points
                                                                Tuesday, October 20, 2009 4:16 PM
         Hello
         I have similar issues.
   0
         SQL Server 2008
Sign In
to Vote
         Using all the correct references I dare say:
         System.Data.SqlClient;
        using Microsoft.SqlServer.Management.Smo;
        using Microsoft.SqlServer.Management.Common;
         using Microsoft.SqlServer.Management.Sdk.Sfc;
        All compiles with no errors.
        I have stripped code down to almost zero for easy debugging.
```

Connecting to server alright and so on.

Excuting following code: SqlConnection connection = new SqlConnection(sqlConnectionString); Server server = new Server(new ServerConnection(connection)); server.ConnectionContext.ExecuteNonQuery(sqlDBQuery);

Where "sqlDBQuery" is a string: "USE [master] GO ALTER DATABASE [Cassiopeia] SET ANSI NULL DEFAULT OFF GO ALTER DATABASE [Cassiopeia] SET ANSI NULLS OFF GO"

But it doesn't matter what "sqlDBQuery" is, I always get the same error, like "incorrect syntax near GO".

I was in belief that SMO would take care of this, when I look at my ConnectionContext is says Batchseparator = "GO"

If I remove GO it's a go... so to speak but I really need to know why my SMO doesn't

Hope I didn't post something not applicable to this thread, but everywhere I look it just says "use smo like this and you're off fine". Well... doesn't work for me.

Regards/

Proposed As Answer by Malakin Monday, November 16, 2009 6:35 PM

Reply

Quote



#### 5 Points

Monday, November 16, 2009 6:38 PM



I don't know if this will help any, but i find this way much more reliable. Have fun and take what you need.

Sign In /// <summary>

```
to Vote /// Run an .sql script trough sqlcmd.
        /// </summary>
        /// <param name="fileName">the .sql script</param>
        /// <param name="machineName">The name of the server.</param>
        /// <param name="databaseName">The name of the database to connect to.</param>
        /// <param name="trustedConnection">Use a trusted connection.</param>
```

/// <param name="args">The arguments passed to the sql script.</param> public void RunSqlScript(string fileName, string machineName, string databaseName, bo ol trustedConnection, string[] args)

 $if \ (!Path.GetExtension(fileName). Equals (".sql", StringComparison. Invariant Culture))\\$ throw new Exception("The file doesn't end with .sql."); // check for used arguments

foreach (var shortArg in new[] { "S", "d", "E", "i" })  $var\ tmpArg = args.SingleOrDefault(a => a.StartsWith(string.Format("-{0}", shortArg), and also are the started are the start$ StringComparison.InvariantCulture)); if (tmpArg != null) throw new ArgumentException(string.Format("Cannot pass -{0} argument to sqlc

md for a second time.", shortArg));

// check the params for trusted connection.

 $var\ user Arg = args. Single Or Default (a => a. Starts With ("-U", String Comparison. Invariant) (a => b. Starts With ("-U", String Comparison) (b => b. Starts Wit$ 

 $var\ passwordArg = args. Single Or Default (a => a. Starts With ("-P", String Comparison. Invariant of the comparison of the comparison$ riantCulture));

if (trustedConnection) if (userArg != null)

throw new ArgumentException("Cannot pass -H argument when trustedConnecti

if (passwordArg != null) throw new ArgumentException("Cannot pass -P argument when trustedConnectio n is used.");

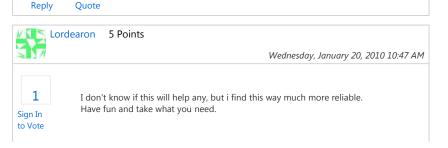
else if (userArg == null)

throw new ArgumentException("Exspecting username(-H) argument when trusted Connection is not used.");

if (passwordArg == null)

```
throw new ArgumentException("Exspecting password(-P) argument when trusted
     Connection is not used.");
        // set the working directory. (can be needed with ouputfile)
        // TODO: Test if the above statement is correct
        var tmpDirectory = Directory.GetCurrentDirectory();
        var directory = Path.IsPathRooted(fileName) ? Path.GetDirectoryName(fileName) : Path.
      Combine(this.ProjectRoot, fileName);
        var file = Path.GetFileName(fileName);
        Directory.SetCurrentDirectory(directory);
        // create cmd line
        var cmd = string.Format(string.Format("SQLCMD -S {0} -d {1} -i \"{2}\\"", machineName,
      databaseName, file));
        foreach (var argument in args.Where(a => a.StartsWith("-", StringComparison.Invariant
     CultureIgnoreCase)))
cmd += " " + argument;
        if (trustedConnection)
          cmd += " -E":
        // create the process
        var process = new System.Diagnostics.Process();
        process.StartInfo.FileName = "cmd":
        process.StartInfo.CreateNoWindow = true;
        process.StartInfo.UseShellExecute = false;
        process.StartInfo.RedirectStandardOutput = true;
        process.StartInfo.RedirectStandardInput = true;
        // start the application
        process.Start();
        process.StandardInput.WriteLine("@ECHO OFF");
        process.StandardInput.WriteLine(string.Format("cd {0}", directory));
        process.StandardInput.WriteLine(cmd);
        process.StandardInput.WriteLine("EXIT");
        process.StandardInput.Flush();
        process.WaitForExit();
        // write the output to my debug folder and restore the current directory
        Debug.Write(process.StandardOutput.ReadToEnd());
        Directory.SetCurrentDirectory(tmpDirectory);
         Proposed As Answer by Malakin Monday, November 16, 2009 6:39 PM
         Edited by Malakin Monday, November 16, 2009 7:18 PM Added 'EXIT' Command. (this will
         end the process when its finished)
Reply
           Ouote
```





```
/// <summarv>
/// Run an .sql script trough sqlcmd.
/// </summary>
/// <param name="fileName">the .sql script</param>
/// <param name="machineName">The name of the server.</param>
/// <param name="databaseName">The name of the database to connect t
o.</param>
/// <param name="trustedConnection">Use a trusted connection.</param
/// <param name="args">The arguments passed to the sql script.</param>
public void RunSqlScript(string fileName, string machineName, string datab
aseName, bool trustedConnection, string[] args)
  // simple checks
  if (!Path.GetExtension(fileName).Equals(".sql", StringComparison.Invariant
    throw new Exception("The file doesn't end with .sql.");
  // check for used arguments
  foreach (var shortArg in new[] { "S", "d", "E", "i" })
     var tmpArg = args.SingleOrDefault(a => a.StartsWith(string.Format("-{
0}", shortArg), StringComparison.InvariantCulture));
     if (tmpArg != null)
       throw new ArgumentException(string.Format("Cannot pass -{0} argu
ment to sqlcmd for a second time.", shortArg));
  // check the params for trusted connection.
  var userArg = args.SingleOrDefault(a => a.StartsWith("-U", StringCompari
son.InvariantCulture)):
  var\ passwordArg = args.SingleOrDefault(a => a.StartsWith("-P", StringCo
mparison.InvariantCulture));
  if (trustedConnection)
     if (userArg != null)
       throw new ArgumentException("Cannot pass -H argument when trus
tedConnection is used."):
    if (passwordArg != null)
       throw new ArgumentException("Cannot pass -P argument when trus
tedConnection is used."):
  else
    if (userArg == null)
       throw new ArgumentException("Exspecting username(-H) argument
when trustedConnection is not used.");
    if (passwordArg == null)
       throw new ArgumentException("Exspecting password(-P) argument
when trustedConnection is not used.");
  // set the working directory. (can be needed with ouputfile)
  // TODO: Test if the above statement is correct
  var tmpDirectory = Directory.GetCurrentDirectory();
  var directory = Path.IsPathRooted(fileName) ? Path.GetDirectoryName(file
Name): Path.Combine(this.ProjectRoot, fileName);
  var file = Path.GetFileName(fileName);
  Directory.SetCurrentDirectory(directory);
  // create cmd line
  var\ cmd = string.Format(string.Format("SQLCMD -S {0} -d {1} -i \"{2}\"", m
achineName, databaseName, file));
  foreach (var argument in args.Where(a => a.StartsWith("-", StringCompar
ison.InvariantCultureIgnoreCase)))
    cmd += " " + argument;
  if (trustedConnection)
    cmd += "-E";
  // create the process
  var process = new System.Diagnostics.Process();
  process.StartInfo.FileName = "cmd";
  process.StartInfo.CreateNoWindow = true;
  process.StartInfo.UseShellExecute = false;
  process.StartInfo.RedirectStandardOutput = true;
  process.StartInfo.RedirectStandardInput = true;
```

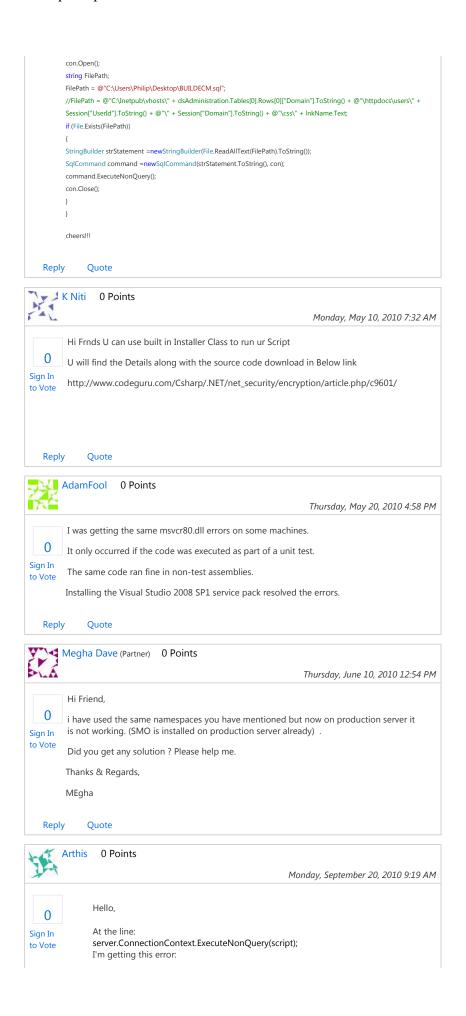
```
// start the application
         process.Start();
         process.StandardInput.WriteLine("@ECHO OFF");
         process.StandardInput.WriteLine(string.Format("cd {0}", directory));
         process.StandardInput.WriteLine(cmd);
         process.StandardInput.WriteLine("EXIT");
         process.StandardInput.Flush();
         process.WaitForExit();
         // write the output to my debug folder and restore the current directory
         Debug.Write(process.StandardOutput.ReadToEnd());
         Directory.SetCurrentDirectory(tmpDirectory);
You have a deadlock when reading the console output after the WaitForExit call as
described here:
http://msdn.microsoft.com/en-us/library/system.diagnostics.process.standardoutput.aspx
I have changed your code to suite my needs (Client is using OSQL, so I couldn't use -v
option)
class Program
     static void Main(string[] args)
       string fileName = @"CreateDatabase-params.sgl";
       SqlConnectionStringBuilder\ connStringBuilder\ =\ new\ SqlConnectionStringBuilder(
);
       connStringBuilder.UserID = "sa";
       connStringBuilder.Password = "Test123";
       connStringBuilder.DataSource = @".\SQLEXPRESS";
       Dictionary<string, string> variables = new Dictionary<string, string>(4);
       variables.Add("AppLogin", "ADMTLogin");
       variables.Add("AppPassword", "Test123");
       variables.Add("DBName", "ADMT");
       variables.Add("DBPath", @"C:\Program Files\Microsoft SQL Server\MSSQL10.SQLE
XPRESS\MSSQL\DATA");
       using(
         TextWriter standard = new StreamWriter("output.txt"),
         error = new StreamWriter("errors.txt"))
         SqlHelper.StandardOutput = standard;
         SqlHelper.ErrorOutput = error;
         Thread th = new Thread(new ThreadStart(delegate
              Sql Helper. Run Sql Script (file Name, conn String Builder, variables); \\
         Console.WriteLine("Creating Database");
         th.Start();
         while (th.IsAlive)
            Console.Write(".");
            Thread.Sleep(1000);
         Console.WriteLine(" Done");
         th.Join();
         standard.Flush();
         error.Flush();
       Console.WriteLine("Database Created");
       Console.ReadKey();
class SqlHelper
    public static TextWriter StandardOutput;
     public static TextWriter ErrorOutput;
    /// <summary>
    /// Parse Parameters then Run a .sql script trough osql.
    /// </summarv>
    /// <param name="fileName">the .sql script</param>
    /// <param name="connStringBuilder">object that holds all connection information.
</param>
    /// <param name="args">The arguments passed to the sql script.</param>
```

```
public static void RunSqlScript(string fileName, SqlConnectionStringBuilder connStri
ngBuilder, Dictionary < string, string > variables)
       // simple checks
       if \ (!Path.GetExtension (fileName). Equals (".sql", StringComparison. Invariant Culture)) \\
          throw new Exception("The file doesn't end with .sql.");
       FileInfo file = ProcessArgs(fileName, variables);
       // create cmd line
       StringBuilder cmd = new StringBuilder(string.Format(
          "OSQL -S \"{0}\" -i \"{1}\" -n",
          /*0*/connStringBuilder.DataSource,
          /*1*/file.ToString()));
       if \ (connStringBuilder.IntegratedSecurity) \\
          cmd.Append(" -E");
       else
          cmd.AppendFormat(" -U {0} -P {1}",
            /*0*/connStringBuilder.UserID,
            /*1*/connStringBuilder.Password);
       // create the process
       var process = new System.Diagnostics.Process();
       process.StartInfo.WorkingDirectory = Environment.CurrentDirectory;
       process.StartInfo.FileName = "cmd":
       process.StartInfo.CreateNoWindow = false;
       process.StartInfo.UseShellExecute = false;
       process.StartInfo.RedirectStandardOutput = true;
       process.StartInfo.RedirectStandardInput = true;
       process.StartInfo.RedirectStandardError = true;
       process.ErrorDataReceived += new DataReceivedEventHandler(process_ErrorData
Received);
       process.OutputDataReceived += new DataReceivedEventHandler(process_Output
DataReceived);
       // start the application
       process.Start();
       process.BeginErrorReadLine();
       process.BeginOutputReadLine();
       process.StandardInput.WriteLine("@ECHO OFF");
       process.StandardInput.WriteLine(cmd.ToString());
       process.StandardInput.WriteLine("EXIT");
       process.StandardInput.Flush();
       process.WaitForExit();
       //delete temporary file
       file.Delete();
       //Reading output after waitforexit creates deadlock as available output buffer is fu
lly filled
       //output.Write(process.StandardOutput.ReadToEnd());
     private static void process_OutputDataReceived(object sender, DataReceivedEventAr
gs e)
       if (e.Data != string.Empty)
          if (StandardOutput == null)
            Console.WriteLine(e.Data);
            StandardOutput.WriteLine(e.Data);
     private static void process_ErrorDataReceived(object sender, DataReceivedEventArgs
e) {
       if (e.Data != string.Empty)
          if (ErrorOutput == null)
            Console.WriteLine(e.Data);
          else
            ErrorOutput.WriteLine(e.Data);
     private static FileInfo ProcessArgs(string fileName, Dictionary<string, string> variable
s)
       FileInfo f = new FileInfo(fileName);
       string SqlScript = f.OpenText().ReadToEnd();
       foreach (KeyValuePair<string,string> kvp in variables)
```

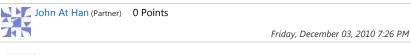
```
SqlScript = SqlScript.Replace(String.Format("$({0})", kvp.Key), kvp.Value);
             FileInfo newFile = new FileInfo("tmpSql.sql");
             using (var w = newFile.CreateText())
                w.Write(SqlScript);
                w.Flush();
             return newFile;
Reply
           Quote
```

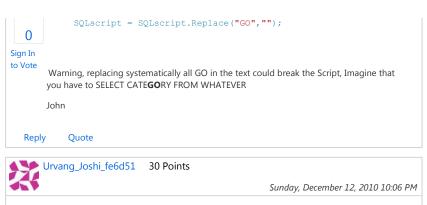


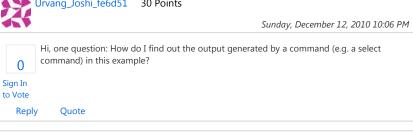




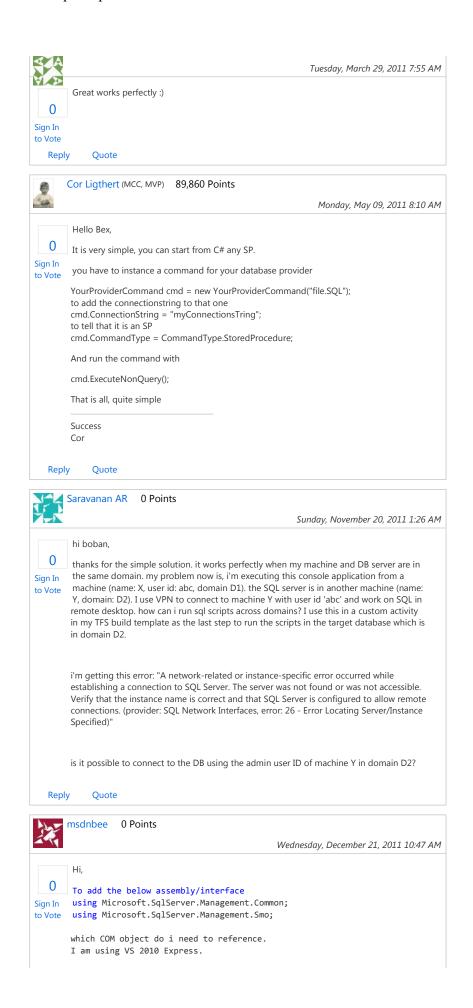
```
Test method IsItScienceFiction.Tests.UserTest.TestMethod1
      threw exception:
      System.Reflection.TargetInvocationException: Exception has
      been thrown by the target of an invocation. --->
       System.TypeInitializationException: The type initializer for
       '<Module>' threw an exception. --->
       <CrtImplementationDetails>.ModuleLoadException: The C++
      module failed to load during appdomain initialization.
         ---> System.DllNotFoundException: Unable to load DLL
       'MSVCR80.dll': The specified module could not be found.
       (Exception from HRESULT: 0x8007007E).
      with this stack trace:
       _encode_pointer(Void*)
       initatexit app domain()
      Language Support. Initialize Per App Domain (Language Support *) \\
      LanguageSupport._Initialize(LanguageSupport*)
      LanguageSupport.Initialize(LanguageSupport*)
       ThrowModuleLoadException(String errorMessage, Exception
      innerException)
       ThrowModuleLoadException(String, Exception)
      LanguageSupport.Initialize(LanguageSupport*)
      cctor()
      Microsoft.SqlServer.Management.Common.ExecuteBatch.GetStatements
      (String sqlCommand)
       System.RuntimeMethodHandle._InvokeMethodFast(Object target, Object[]
       arguments, SignatureStruct& sig, MethodAttributes methodAttributes,
      RuntimeTypeHandle typeOwner)
      System.RuntimeMethodHandle.InvokeMethodFast(Object target, Object[]
       arguments, Signature sig, MethodAttributes methodAttributes,
       RuntimeTypeHandle typeOwner)
      System.Reflection.RuntimeMethodInfo.Invoke(Object obj, BindingFlags
      invokeAttr, Binder binder, Object[] parameters, CultureInfo culture, Boolean
      skipVisibilityChecks)
      System. Reflection. Runtime Method Info. Invoke (Object\ obj,\ Binding Flags
       invokeAttr, Binder binder, Object[] parameters, CultureInfo culture)
      System. Runtime Type. Invoke Member (String\ name,\ Binding Flags
      bindingFlags, Binder binder, Object target, Object[] providedArgs,
      ParameterModifier[] modifiers, CultureInfo culture, String[] namedParams)
      System.Type.InvokeMember(String name, BindingFlags invokeAttr, Binder
       binder, Object target, Object[] args, CultureInfo culture)
      {\it Microsoft. Sql Server. Management. Common. Server Connection. Get Statements}
       (String query, ExecutionTypes executionType, Int32& statementsToReverse)
      Microsoft. Sql Server. Management. Common. Server Connection. Execute Non Query \\
      (String sqlCommand, ExecutionTypes executionType)
      Microsoft.SqlServer.Management.Common.ServerConnection.ExecuteNonQuery
      (String sqlCommand)
      IsItScienceFiction.Tests.UserTest.TestMethod1() in C:\Users\J. Pablo
      Fernández\Documents\Visual Studio 2008
      \Projects\IsItScienceFiction\IsItScienceFiction.Tests\UserTest.cs: line 75
      I am running Visual Studio 2008 with SQL Server 2008. What am I doing
      wrong?
Hi Pablo.
I experience the same issue. Did you ever find a solution?
Yoann
```

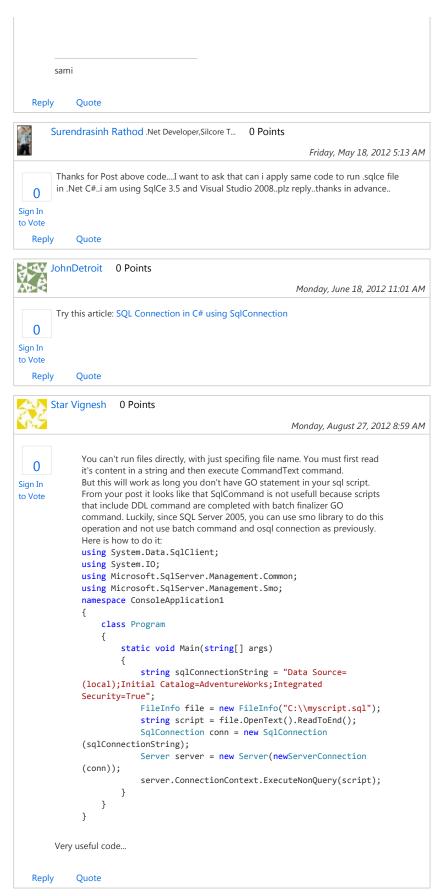












2012 Microsoft. All rights reserved.

Terms of Use | Trademarks | Privacy Statement | Site Feedback