



**US Army Corps  
of Engineers®**  
Engineer Research and  
Development Center



## **Setting up a Local Database From an Existing .mdf File**

Abdías I. Santiago Lugo, Katherine E. Mixon, Amy E. Bednar,  
and Barry C. White

September 2019



**The U.S. Army Engineer Research and Development Center (ERDC)** solves the nation's toughest engineering and environmental challenges. ERDC develops innovative solutions in civil and military engineering, geospatial sciences, water resources, and environmental sciences for the Army, the Department of Defense, civilian agencies, and our nation's public good. Find out more at [www.erdcd.usace.army.mil](http://www.erdcd.usace.army.mil).

To search for other special reports published by ERDC, visit the ERDC online library at <http://acwc.sdp.sirsi.net/client/default>.

# **Setting up a Local Database From an Existing .mdf File**

Abdías I. Santiago Lugo, Katherine E. Mixon, Amy E. Bednar,  
and Barry C. White

*Information Technology Laboratory  
U.S. Army Engineer Research and Development Center  
3909 Halls Ferry Road  
Vicksburg, MS 39180-6199*

Final Report

Approved for public release; distribution is unlimited.

Prepared for Headquarters, U.S. Army Corps of Engineers  
Washington, DC 20314-1000

Under Project 46, "Sustainability/Logistics - Transportation and Distribution  
Management"

## Abstract

The Planning Logistics Analysis Network System (PLANS) team needed a repeatable process for setting up an accessible local database for development using the .mdf database file format on developers' machines. Online sources for this procedure are available, but they must be adapted due to security restrictions imposed by the Engineer Research and Development Center-Defense Research and Engineering Network (ERDC-DREN). Access to the server's database is restricted, requiring copies of the database files to be handled directly, rather than through web-based database calls. This special report provides step-by-step instructions for setting up a local development database using the .mdf database file format on computers on the ERDC-DREN network.

**DISCLAIMER:** The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products. All product names and trademarks cited are the property of their respective owners. The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

**DESTROY THIS REPORT WHEN NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.**

# Contents

Abstract.....	ii
Figures .....	iv
Preface.....	v
Acronyms and Abbreviations .....	vi
<b>1 Introduction.....</b>	<b>1</b>
1.1 Background.....	1
1.2 Objectives.....	1
1.3 Approach .....	1
1.4 Scope.....	2
<b>2 Installing SQL Server Software .....</b>	<b>3</b>
2.1 Step 1: download SQL server .....	3
2.2 Step 2: install and setup SQL Server.....	3
<b>3 Installing SQL Server Management Studio (SSMS).....</b>	<b>6</b>
3.1 Step 1: download SQL Server Management Studio (SSMS) .....	6
3.2 Step 2: install SMSS .....	6
<b>4 Setting Up a Database Using the Structured Query Language Command Prompt (SQLCMD) .....</b>	<b>7</b>
4.1 Step 1: administrator command prompt (CMD).....	7
4.2 Step 2: SQLCMD prompt .....	7
4.3 Step 3: verify server version.....	8
4.4 Step 4: create new database .....	8
4.5 Step 5: verify new database.....	9
4.6 Step 6: exit SQLCMD.....	9
<b>5 Viewing the Database in Microsoft SQL Server Management Studio .....</b>	<b>10</b>
5.1 Step 1: server name .....	10
5.2 Step 2: verify database.....	10
5.3 Step 3: verify database permissions .....	10
<b>6 Accessing the Database from Visual Studio .....</b>	<b>11</b>
6.1 Step 1: create new connection .....	11
6.2 Step 2: choose data source .....	11
6.3 Step 3: server information .....	11
6.4 Step 4: add connection string.....	12
<b>7 Conclusion.....</b>	<b>14</b>
References.....	15

# Figures

- Figure 1. Screenshot of SQL Server Installation Center. .... 3
- Figure 2. Screenshot of SQL Server Setup Feature Selection. .... 4
- Figure 3. Screenshot of SQL Server Setup Instance Configuration. .... 5
- Figure 4. Screenshot of SQL Server Setup Database Engine Configuration..... 5
- Figure 5. Screenshot of the SQLCMD prompt. .... 7
- Figure 6. Screenshot of settings. .... 10
- Figure 7. Screenshot of Choose Data Source in Visual Studio. .... 11
- Figure 8. Screenshot of Add Connection in Visual Studio. .... 12

## Preface

This study was conducted for the Military Engineering Business Area under Project 46, “Sustainability/Logistics - Transportation and Distribution Management.” The technical monitor was Dr. Amy E. Bednar.

The work was performed by the Computational Analysis Branch (CAB) of the Computational Science and Engineering Division (CSED), U.S. Army Engineer Research and Development Center (ERDC), Information Technology Laboratory (ITL). At the time of publication, Dr. Jeffrey L. Hensley was Chief, CEERD-CAB, Dr. Jerrell R. Ballard was Chief, CEERD-CSED, and the Technical Director was Mr. Nicholas R. Boone, CEERD-GZT. The Deputy Director of ITL was Ms. Patti S. Duett, and the Director of ITL was Dr. David A. Horner.

The Commander of ERDC was COL Teresa A. Schlosser and the Director was Dr. David W. Pittman.

## Acronyms and Abbreviations

CAB	Computational Analysis Branch
CMD	Command Prompt
CSED	Computational Science and Engineering Division
DoD	Department of Defense
EPE	Environmental Processes and Engineering
EPR	Environmental Risk Assessment Branch
EQT	Environmental Quality and Technology
ERDC	Engineer Research Development Center
ERDC-DREN	Engineer Research and Development Center-Defense Research and Engineering Network
ITL	Information Technology Laboratory
PLANS	Planning Logistics Analysis Network System
SSMS	SQL Server Management Studio
SQL	Structured Query Language
SR	Special Report
USACE	U.S. Army Corps of Engineers

# **1 Introduction**

## **1.1 Background**

The Planning Logistics Analysis Network System (PLANS) team needed a repeatable process for setting up a local development database using the .mdf database file format on developers local machines. .mdf is the primary database file format used by Microsoft Structured Query Language (SQL) Server, an enterprise-level database program. Using a local database allows the developer to test their code without modifying the database on the active server. Using the server-side database to test code can cause concurrency problems when working with a team of developers and people who give presentations/demonstrations using the system. This can result in system downtime when potential bugs (i.e., changes to database formats) are introduced. A local database provides each developer with their own sandbox environment for testing and implementing changes. Additionally, running each scenario in PLANS produces large amounts of data, this slows the system and monopolizes the limited server space. The test data on the server had to be manually deleted and on a case-by-case basis. With each developer adding their own test data to the same database, the interface became cluttered, and the task of removing unnecessary or outdated scenario data was time-consuming.

## **1.2 Objectives**

Several online resources are available that describe this process. However, details were sparse or needed to be modified locally due to the use of the Engineer Research and Development Center's (ERDC) more restricted Defense Research and Engineering Network (DREN), this requires that databases be transferred via file copies rather than duplicated through the internet.

## **1.3 Approach**

Each developer needs a local database on their development system. Providing a local database for the developer creates their own sandbox environment for testing and implementing changes. Additionally, running each scenario in PLANS produces large amounts of data, this slows the system and monopolizes the limited server space. Many online recourses for setting up and accessing local database exist in the .mdf format which

is the required database format for PLANS. However, these resources had to be modified due to security restrictions imposed by the Engineer Research and Development Center-Defense Research and Engineering Network (ERDC-DREN). Copies of the database files need to be handled directly, rather than through web-based database calls. Due to developers joining the team throughout the development lifecycle of PLANS, it was imperative there was a repeatable process for setting up an accessible local database for development.

#### **1.4 Scope**

This special report (SR) provides step-by-step instructions for setting up a local development database using the .mdf database file format on computers on the ERDC-DREN.

## 2 Installing SQL Server Software

### 2.1 Step 1: download SQL server

Download Microsoft SQL Server Express by searching for “download SQL Server <version>” (<version> is the relevant version) in the preferred internet web browser or by going to the following link to download SQL Server 2014 Express (Microsoft 2014; Rich Developer 2018):

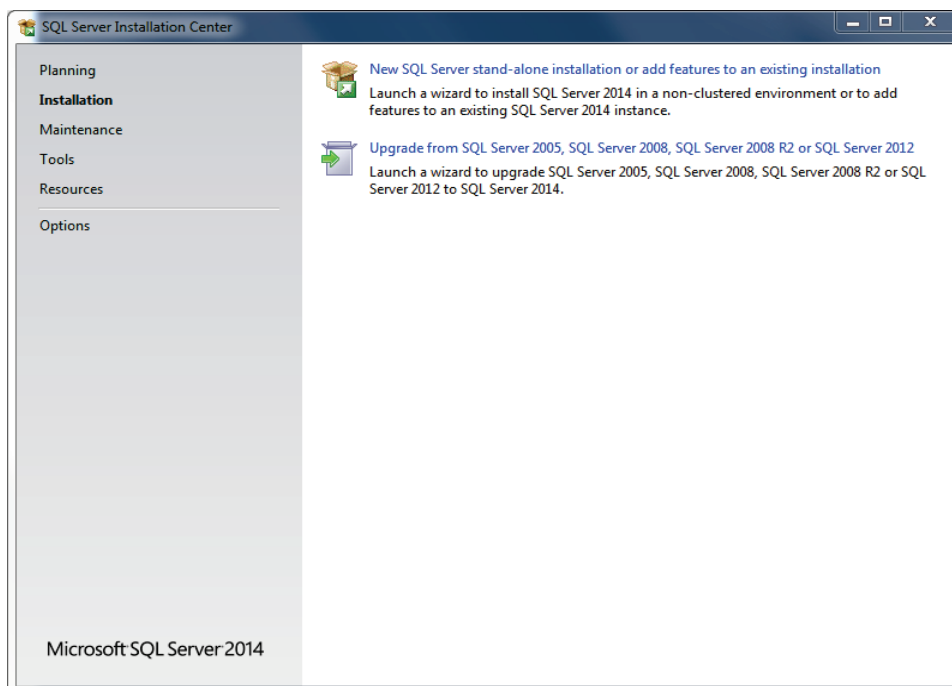
<https://www.microsoft.com/en-us/download/details.aspx?id=42299>

### 2.2 Step 2: install and setup SQL Server

The following steps are instructions for installing and setting up SQL Server 2014 Express (Figures 1–4). It is important to note that these steps may vary for other versions of SQL Server.

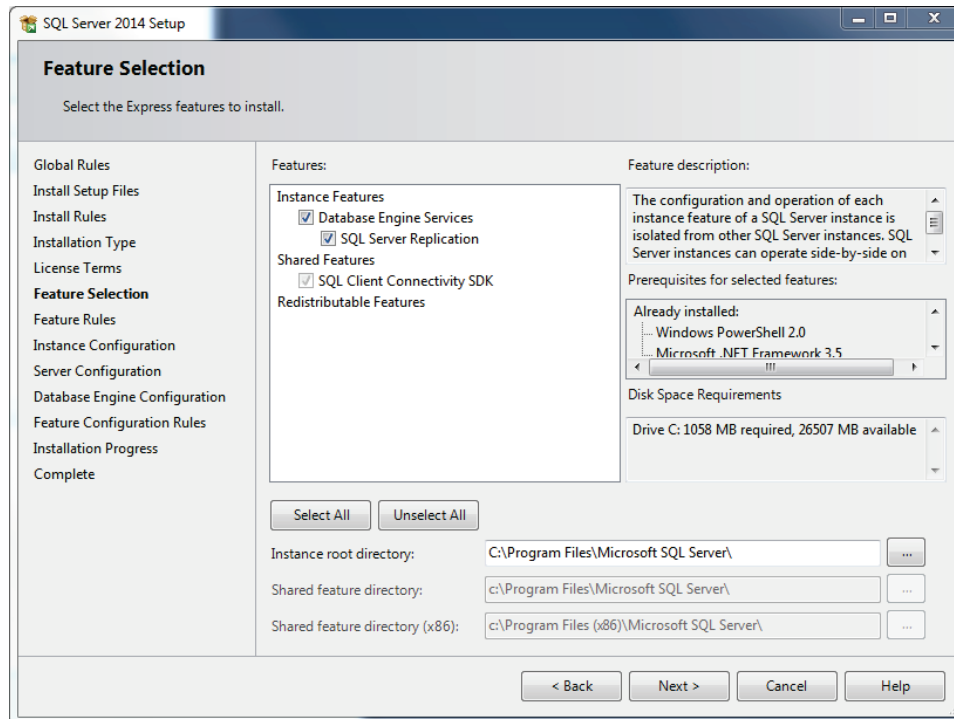
1. Open the file once the download completes.
2. Select a location to extract the file.
3. Click “OK.”
4. Go to the extracted folder in the selected file location.
5. Double-click on the “SETUP.EXE” file.
6. Select “Installation” and “New SQL Server Installation ...” (or similar).

Figure 1. Screenshot of SQL Server Installation Center.



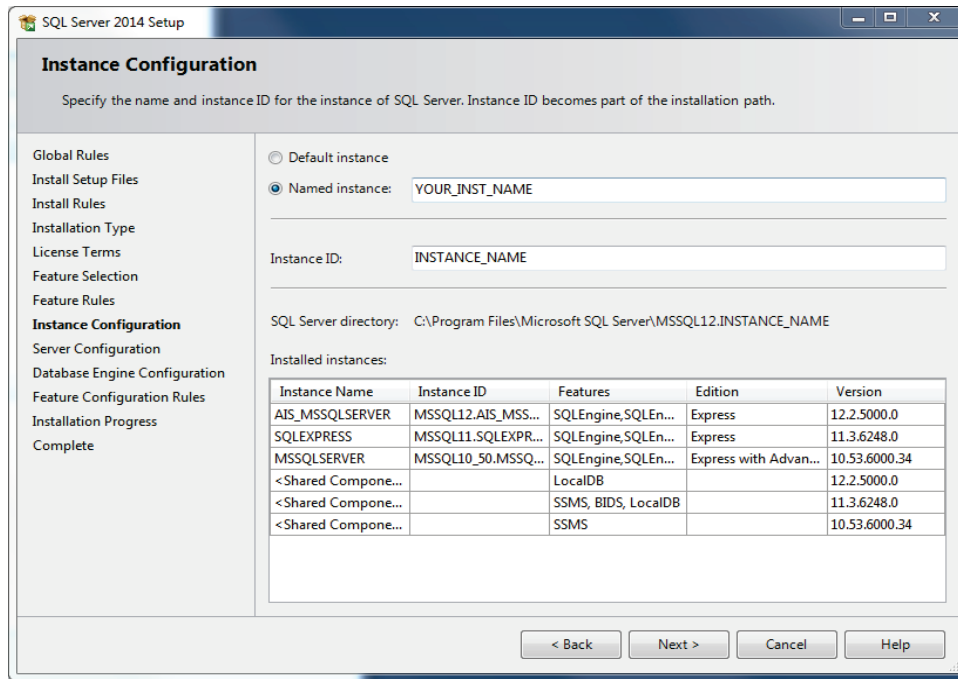
7. Read the license terms.
8. Accept the license terms.
9. Click “Next.”
10. Select all features.
11. Choose the directory for the applicable server in the “Instance root directory” or any of the “Shared feature directories.”
  - a. Note: Preferred directories are:
    - (1) C:\Program Files.
    - (2) C:\Program Files (x86) or similar.
12. Click “Next.”

Figure 2. Screenshot of SQL Server Setup Feature Selection.



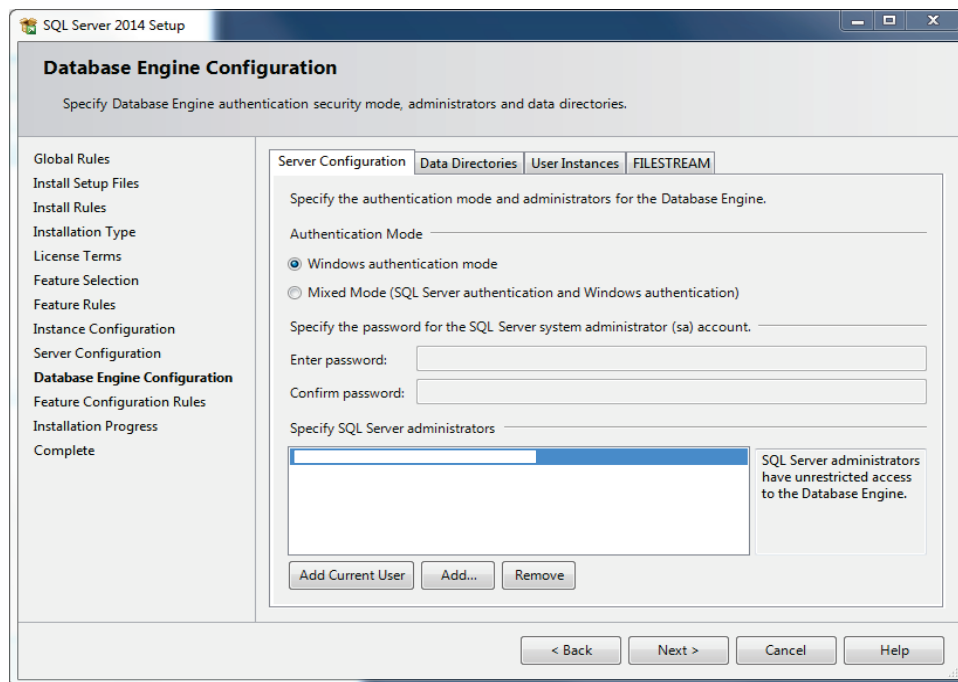
13. Enter the preferred name instance under “Name instance.”
  - a. The name instance will be the “instance” in “server\instance” used for accessing the database through SQL Server Management Studio or the SQLCMD prompt.
14. Click “Next.”

Figure 3. Screenshot of SQL Server Setup Instance Configuration.



15. Click "Next."
16. Select the authentication mode.
17. Click "Next."

Figure 4. Screenshot of SQL Server Setup Database Engine Configuration.



18. Exit the SQL Server Setup once the installation process has completed.

## **3 Installing SQL Server Management Studio (SSMS)**

### **3.1 Step 1: download SQL Server Management Studio (SSMS)**

SQL Server Management Studio (SSMS) is a software application that is used for configuring, managing, and administering all components within Microsoft SQL Server. Download SSMS from the following link (Microsoft, 2017):

<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms>

### **3.2 Step 2: install SMSS**

Run and install the software using the setup wizard provided once the SSMS setup file has been downloaded.

## 4 Setting Up a Database Using the Structured Query Language Command Prompt (SQLCMD)

### 4.1 Step 1: administrator command prompt (CMD)

1. Open Windows command prompt.
2. Select the “RUN AS ADMINISTRATOR” option.
  - a. This is to avoid insufficient permissions when accessing the directory where the database might be located.

### 4.2 Step 2: SQLCMD prompt

1. Enter the following command in the command prompt:

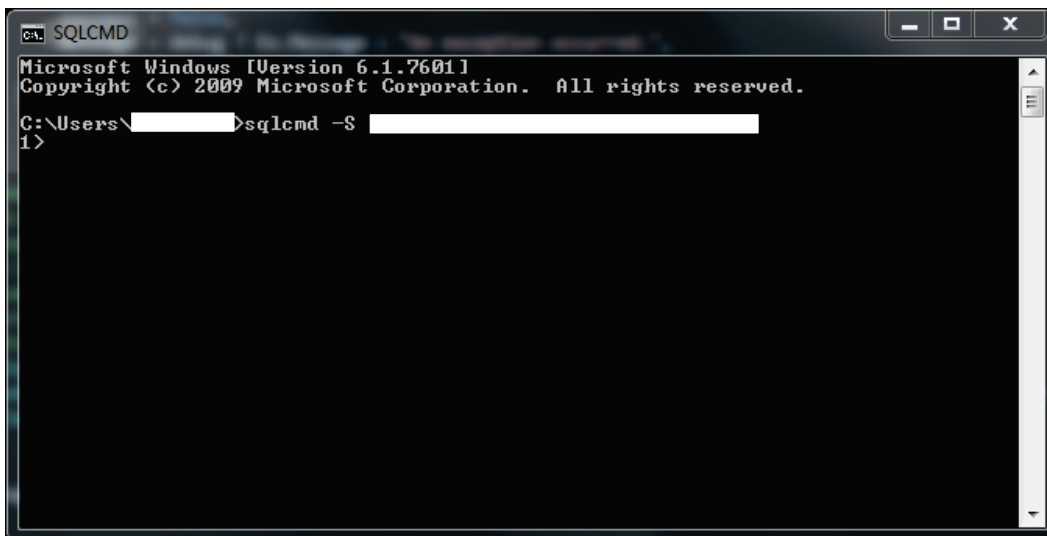
```
> sqlcmd -S Server\Instance
```

Note that “Server” is the name of the computer and “Instance” is the name of the MSSQL instance that was created when installing the SQL server (Microsoft 2012a, 2012b; Thompson Reuters nd).

For example, Computer1\MSSQL is server name “Computer1,” which uses the MSSQL server.

The user should now be in the SQLCMD prompt (Figure 5).

Figure 5. Screenshot of the SQLCMD prompt.



The blanks should be replaced with the “username” followed by the “Server\Instance.”

**Note:** The directory location is irrelevant.

### 4.3 Step 3: verify server version

1. In SQLCMD, enter the following commands:

```
> Select @@version  
Press the <enter> key
```

```
> Go  
Press the <enter> key
```

2. Verify that the server version is correct.

### 4.4 Step 4: create new database

1. Enter the following commands in SQLCMD:

```
1> USE [master]
```

```
2> GO
```

```
3> CREATE DATABASE [database_name] ON
```

```
4> ( FILENAME = N'C:\<path directory>\<database  
name>.mdf' ),
```

```
5> ( FILENAME = N'C:\<path directory>\<database  
name>.ldf' )
```

```
6> FOR ATTACH ;
```

```
7> GO
```

Note: “database\_name” is the name of the database to be created.

“<path directory>” is the path where the .mdf and the .ldf files are located.

“<database name>” is the prefix filename (no extension) of the .mdf file or the .ldf file.

#### **4.5 Step 5: verify new database**

1. To verify that the new database has been created, enter the following:

```
1> select name from sys.databases
```

```
2> go
```

The “database\_name” should appear in the list. This list corresponds to all the databases on the SQL instance.

#### **4.6 Step 6: exit SQLCMD**

1. To exit the SQLCMD, enter the either of the following:

```
> quit
```

```
> exit
```

## 5 Viewing the Database in Microsoft SQL Server Management Studio

### 5.1 Step 1: server name

When opening the Microsoft SQL Server Management Studio, the “Server Name” should be the same as the one entered in the command prompt (Section 2.2 Step 13; Server\Instance).

For Server Name enter: “Server\Instance.”

### 5.2 Step 2: verify database

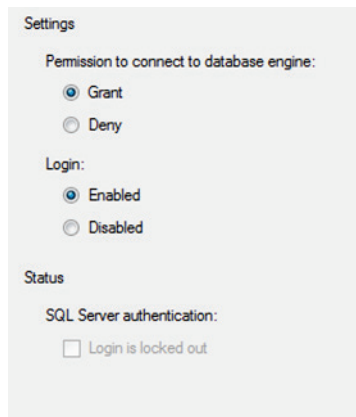
Make sure that the database has been added.

### 5.3 Step 3: verify database permissions

Verify that the right permissions for the database are displayed.


1. In the object explorer, expand “Security.”
2. Expand “Logins.”
3. Right-click the username and select “properties.”
  - a. The “Login Properties” window should appear.
4. Go to “Server Roles” and check both “public” and “sysadmin” roles.
5. Go to “User Mapping” and ensure that the correct database is checked.
6. Go to “Status.”
7. Verify that “grant” and “enable” are selected (Figure 6).
8. Ensure that Windows Authentication is enabled for this login.

Figure 6. Screenshot of settings.



## 6 Accessing the Database from Visual Studio

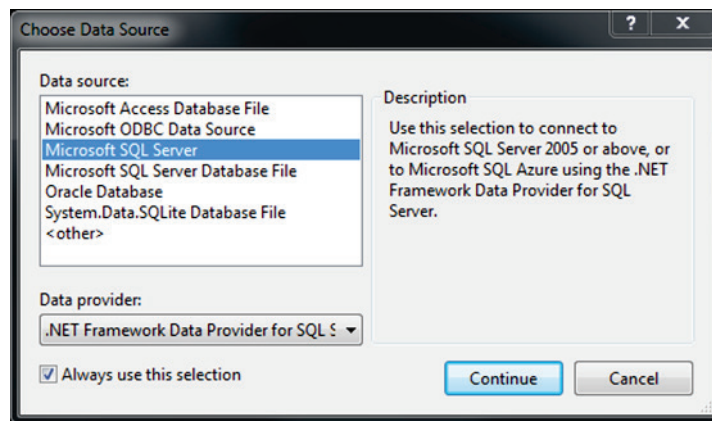
### 6.1 Step 1: create new connection

On the project's "Server Explorer", create a new connection by selecting the "Connect to Database" icon .

### 6.2 Step 2: choose data source

Choose the data source (Microsoft SQL server) and continue (Figure 7).

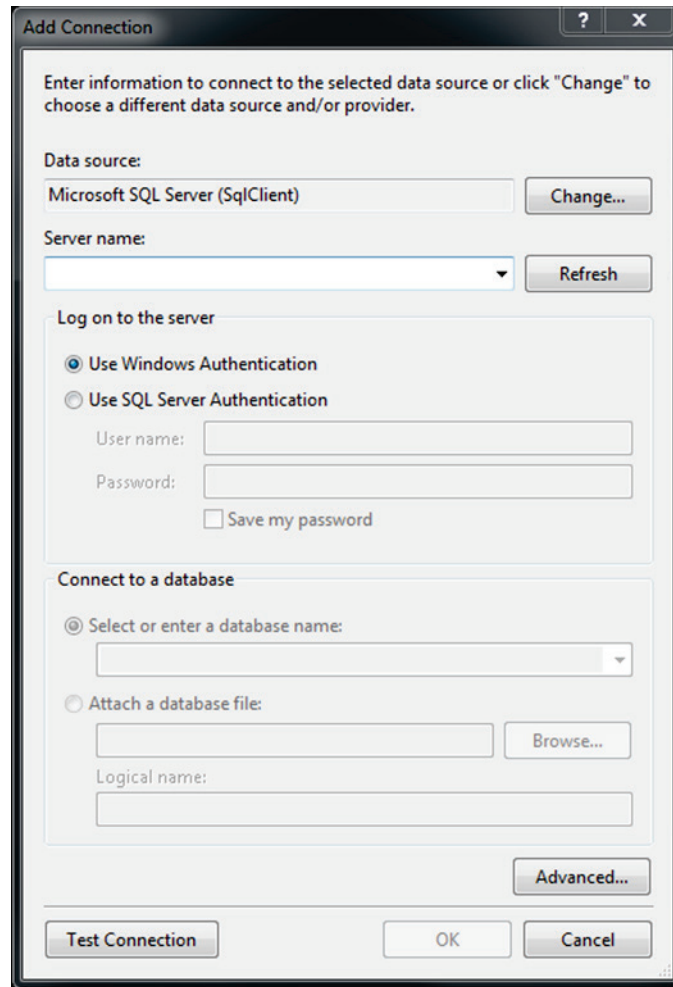
Figure 7. Screenshot of Choose Data Source in Visual Studio.



### 6.3 Step 3: server information

On the "Server Name" enter the "Server\Instance" (e.g., Section 2.2 Step 13). For log on to the server select "Use Windows Authentication" and then select the database from the dropdown (Figure 8).

Figure 8. Screenshot of Add Connection in Visual Studio.



#### 6.4 Step 4: add connection string

1. Next, the connection string will need to be added.
2. Go to the Server Explorer and click on the instance you are going to connect.
3. Right-click and select “Properties.”
4. Along the right-hand side of the Visual Studio window, a properties window should appear. Copy the Connection String that is listed.
5. Open up the Web.config file and paste the connection string between the `<connectionStrings>` tags. It should look like the below example, replacing the server\instance with the own local server and instance, and database with the name of the database. Integrated Security should be set to True if using Windows Authentication.

```
<add name="PLANSConnectionString" connectionString="Data
Source=server\instance;Initial
Catalog=database_name;Integrated Security=True"
providerName="System.Data.SqlClient" />
```

## **7 Conclusion**

This special report shows step-by-step instructions for setting up a local development database on a developer's machine. Using a local database allows the developer to test their code without modifying the database on the active server. Using the server-side database to test code can cause concurrency problems when working with a team of developers and presentation providers. This can result in system downtime when breaking-changes are introduced. A local database provides each developer with their own sandbox environment for testing and implementing changes.

## References

- Microsoft. 2012a. "How to: Attach a Database File to SQL Server Express."  
[https://technet.microsoft.com/en-us/library/ms165673\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms165673(v=sql.105).aspx).
- Microsoft. 2012b. "Using the sqlcmd Utility (SQL Server Express)."  
[https://technet.microsoft.com/en-us/library/ms165702\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms165702(v=sql.105).aspx)
- Microsoft. 2014. "Microsoft SQL Server 2014 Express." Last modified 25 June 2014.  
<https://www.microsoft.com/en-us/download/details.aspx?id=42299>.
- Microsoft. 2017. "Download SQL Server Management Studio."  
<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms>.
- Rich Developer. 2018. "How to Install SQL Server 2008 R2 using Windows 10."  
<https://www.youtube.com/watch?v=DUCbvfE034>.
- Thompson Reuters. nd. "How to identify your SQL Server Version and Edition." Accessed on 17 June 2018. [http://cs.thomsonreuters.com/ua/practice/cs\\_us\\_en/kb/how-to-identify-your-sql-server-version-and-edition.htm](http://cs.thomsonreuters.com/ua/practice/cs_us_en/kb/how-to-identify-your-sql-server-version-and-edition.htm).

