.NET App Dev Hands-On Lab Prerequisites

Before starting the rest of the workshop, you must have the .NET (Core) SDK, .NET Runtime, access to a local SQL Server Database, an appropriate .NET development IDE, and an SQL Server IDE.

Part 1: Version Options

.NET/IDE Version Choices

This workshop uses .NET 8. If using Visual Studio, reference the minimum version based on the runtime and SDK. This will be necessary when you install (or confirm your installation of) Visual Studio (Mac or Windows)

.NET 8 versions:

- SDK => 8.0.100+
- Runtime => 8.0.0+
- Visual Studio =>

.NET Runtime Version	.NET SDK Version	VS Win Version
8.0.5	8.0.3XX	2022 17.9+
8.0.4	8.0.2XX	2022 17.9+
8.0.0	8.0.1XX	2022 17.8+

See https://dotnet.microsoft.com/en-us/download/dotnet/8.0 for the rest of the version correlations.

Supported .NET development IDEs include (one is required):

- [VS Windows] See above for version information.
- [VS Mac] will no longer be supported after
- [Any OS] Visual Studio Code 1.89+

SQL Server

SQL Server options (one is required):

- [Windows] SQL Server LocalDb (installed with Visual Studio 2022), or
- [Windows] SQL Server 2019/2022 Developer Edition (or newer)
- [Any OS] Docker Community with SQL Server 2019
- [ARM-Mac] https://sqlblog.org/2023/03/03/sql-server-apple-silicon

Supported SQL Server IDEs include (one is recommended):

- [Windows] SQL Server Management Studio 18.11+/19.2+/20.0+
  - **NOTE:** SQL Server Management Studio version has no relation to SQL Server version.
- [Any OS] Azure Data Studio 1.48.0+

Part 2: Permissions

You must have admin permissions on your machine to complete this hands-on lab.

Part 3: Installing the Prerequisites

Step 1: Install/Confirm .NET Runtime and SDK

- Download and install the latest .NET 8 SDK, ASP.NET Core 8 Runtime, and .NET 8 Runtime from http://dot.net (the .NET Desktop Runtime is not used for the hands-on lab).
- Check the version of the .NET Runtime by entering (must have 8.0.x or above):

```
dotnet --list-runtimes
```

- Check the version of the .NET SDK by entering (must have 8.0.xxx or above):

```
dotnet --list-sdks
```

- You can check for updates to your versions by using the following:

```
dotnet sdk check
```

Step 2: Install a Development IDE (one is required)

Option 1: [Windows] Download and install any edition of Visual Studio 2022

- If you already have the correct version of Visual Studio 2022 installed or don't plan on using VS 2022, continue to the next step.
- Download Visual Studio 2022 (any edition) from the Visual Studio home page: https://www.visualstudio.com
- The Community Edition is free and has everything you need to complete this Hands-On Lab
    - Start the installer
- The new installation experience has separate workloads based on what type of work you intend to do. For this lab, select the "**ASP.NET and web development**" workload as well as the "**Data storage and processing**" workloads.

Option 2: [Any OS] Download and install Visual Studio Code

- If you already have the latest version of VS Code installed or don't plan on using VS Code, continue to the next step.
- Download Visual Studio Code from https://visualstudio.microsoft.com/.

Install the "C#" and "C# Dev Kit" extensions.

Step 3: Install SQL Server IDE (SSMS or ADS)

Neither of these is required for the workshop but having one installed makes it easier to work with the database. You only need to install one.

Option 1: SQL Server Management Studio (Windows only)

Download/Install SQL Server Management Studio (SSMS) from https://aka.ms/ssmsfullsetup

Option 2: Azure Data Studio (Mac/Windows)

Download/Install the free Azure Data Studio from https://learn.microsoft.com/en-us/azure-data-studio/download-azure-data-studio

Step 4: SQL Server

You must have access to SQL Server 2016+ for this workshop. If you are on a Windows machine, the easiest option is to use `LocalDb`, installed with Visual Studio 2022. If you have another edition of SQL Server installed on your local machine, you have completed the prerequisites for this workshop.

Check the version of the SQL Server instance:

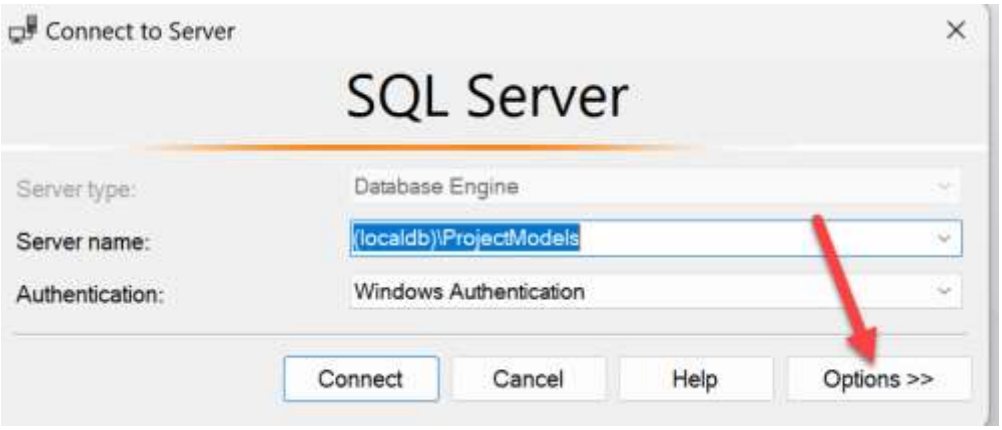
Connect to your local SQL Server instance using SSMS or Azure Data Studio. When connecting to `LocalDb`, use Windows Authentication (in on Windows) and the Server name to use is:

```
(localdb)\mssqllocaldb
```

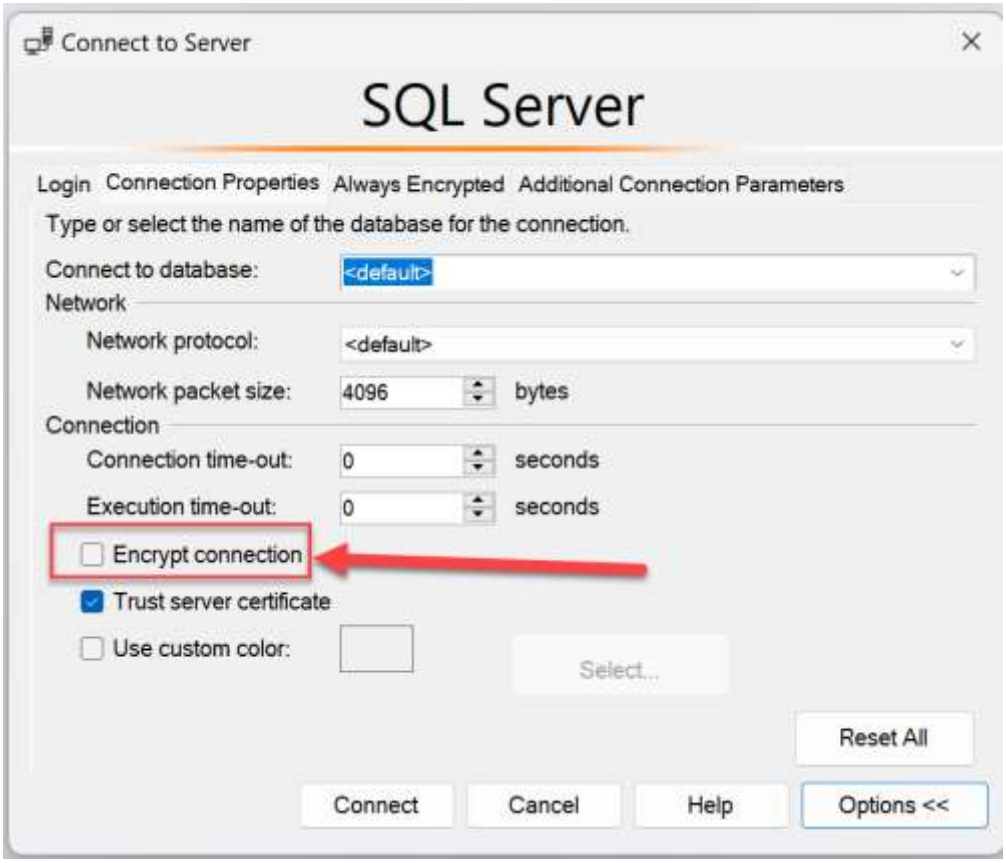
If that server name doesn't work, you can also try the following, which is another location where VS2022 will install LocalDb:

```
(localdb)\ProjectModels
```

If the connection to either of these does not work and you get an error about encryption, click the options button on the connection screen:



On the options page, uncheck the "Encrypt the connection" check box:



Check that the version of SQL Server is 2016 or greater by running the following in a query window:

```
SELECT @@Version
```

The command should return the following (or higher):

```
Microsoft SQL Server 2019
```

If your version is not 2019 (or above), or you do not have SQL Server installed, then follow either option below to get SQL Server installed.

NOTE: This workshop is built using SQL Server 2019, but SQL Server 2019 and 2022 will both work.

Option 1: Download and install SQL Server 2022 Developer (Windows)

- Download/Install SQL Server 2022 Developer Edition: https://go.microsoft.com/fwlink/p/?linkid=2215158

Option 2: Install Docker Desktop (Windows/Mac/Linux)

Docker is a containerization platform that runs on Windows, MacOS, and Linux.

NOTE: If you are using a Windows-based machine and have SQL Server installed, Docker is optional for this workshop. If you are not on a Windows/Linux machine or can't have SQL Server 2019 installed on your laptop, Docker is required.

- Download and install Docker Desktop from https://www.docker.com/products/docker-desktop
    a) Select the edition for your operating system (Windows/Mac)
    b) During installation, when prompted about what type of containers to use, select Linux containers (and not Windows containers), even if you are on a Windows machine. This type of container will be loaded and is not related to your computer's operating system.
    c) This is a free tool but requires you to have a Docker user id and password

A Docker image is like a class definition, while a Docker Container is like an instance of that class. To run SQL Server in Docker, you must first pull the image from Docker Hub and create a container using that image.

- Pull the SQL Server 2019 for Linux image. Enter the following command:

```
docker pull mcr.microsoft.com/mssql/server:2019-latest
```

- When creating an image, there are two required environment variables, "ACCEPT_EULA" and "SA_PASSWORD". An optional environment variable "MSSQL_PID" sets the product version. The host port mapping to the image port needs to be set, and a friendly name added. Create the container using the following command:
    a) **NOTE:** On Windows, use double quotes ("). On Mac and Linux, use single quotes (').

```
docker run -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=P@ssw0rd" -p 5433:1433 --name AutoLot -d
mcr.microsoft.com/mssql/server:2019-latest
```

Summary

These are all the tools you need to complete this Hands-on Lab. On lab day you will get the URL for the repo to clone/download.