Creating the solution

- Create a new solution in visual studio by going File->New->Project…
- Select “ASP.NET Web Application (.NET Framework)”
- Set the name to “ClientApplicationMVC”
- Set the “Solution name” to “Call It Local”
- At the top of the window, make sure that the latest version of the .NET Framework is selected. (Minimum 4.6.1)
- After pressing “OK” Select “MVC” from the list of templates and press “OK”
- We will come back to this project later. For now move onto the Messages Library

Installing the Messages Library

Step 1: Create a Class Library

- In visual studio, right click on your ‘Call It Local’ Solution in the solution explorer and go Add->New Project…->Visual C# ->Class Library (.NET Framework) and name it “Messages”. At the top of the window, make sure that the latest version of the .NET Framework is selected. (Minimum 4.6.1)
- Delete the auto-generated “Class1.cs” file.

Step 2: Add source files

- copy and paste the source files from the “Messages” folder in the unzipped file to the directory where your project was created.
- Then, just drag each of the added folders and files onto the “Messages” Project in the solution explorer (drag into the Visual Studio UI) so that visual studio becomes aware of the files.

Step 3: Add the missing dependencies to the the project.

- Open the “Package Manager Console” in visual studio by navigating to Tools->NuGet Package Manager->Package Manager Console and type the following commands:

- `Install-Package NServiceBus -ProjectName Messages`

- Right click on the “References” section under the “Messages” project in the solution explorer and select “Add Reference…”
- In the window that appears, select “Assemblies->Extensions”, scroll until you find “MySql.Data”. Select the checkbox beside it and press ‘OK”
- Right click on the “References” section under the “Messages” project in the solution explorer and select “Add Reference…”
- In the window that appears, select “Assemblies->Framework” on the left menu, and scroll until you find “System.Runtime.Serialization”. Select the checkbox beside it and press “OK”.

Installing the Echo Service

Step 1: Create a Console App (.NET Framework).

- In visual studio, right click on your ‘Call It Local’ Solution in the solution explorer and go Add->New Project…->Console App (.NET Framework)
- Name it “EchoService”. At the top of the window, make sure that the latest version of the .NET Framework is selected. (Minimum 4.6.1)
- Delete the Auto-Generated Program.cs file

Step 2: Add Echo Service source code from d2l to the project

- First copy and paste the source files from the “EchoService” folder in the unzipped file to the directory where your project was created.
- Then, just drag each of the added folders and files onto the “EchoService” Project in the solution explorer (drag into the Visual Studio UI) so that visual studio becomes aware of the files.

Step 3: Add the missing dependencies to the the project.

- Open the “Package Manager Console” in visual studio and type the following command:

- `Install-Package NServiceBus -ProjectName EchoService`

- Right click on the “References” section under the “EchoService” project in the solution explorer and select “Add Reference…”
- In the window that appears, select “Projects” on the left menu, and select the checkbox beside “Messages” and press “OK”.
- Right click on the “References” section under the “EchoService” project in the solution explorer and select “Add Reference…”
- In the window that appears, select “Assemblies->Extensions”, scroll until you find “MySql.Data”. Select the checkbox beside it and press ‘OK”.

Installing the Authentication Service

Step 1: Create a Console App (.NET Framework).

- In visual studio, right click on your ‘Call It Local’ Solution in the solution explorer and go Add->New Project…->Console App (.NET Framework)
- Name it “AuthenticationService”. At the top of the window, make sure that the latest version of the .NET Framework is selected. (Minimum 4.6.1)
- Delete the Auto-Generated Program.cs file

Step 2: Add Authentication Service source code from d2l to the project

- First copy and paste the source files from the “AuthenticationService” folder in the unzipped file to the directory where your project was created.
- Then, just drag each of the added folders and files onto the “AuthenticationService” Project in the solution explorer (drag into the Visual Studio UI) so that visual studio becomes aware of the files.

Step 3: Add the missing dependencies to the the project.

- Open the “Package Manager Console” in visual studio and type the following commands:

```
- Install-Package NServiceBus -ProjectName AuthenticationService
- Install-Package NServiceBus.Callbacks -ProjectName AuthenticationService
```

- Right click on the “References” section under the “AuthenticationService” project in the solution explorer and select “Add Reference…”
- In the window that appears, select “Projects” on the left menu, and select the checkbox beside “Messages” and press “OK”.
- Right click on the “References” section under the “AuthenticationService” project in the solution explorer and select “Add Reference…”
- In the window that appears, select “Assemblies->Extensions”, scroll until you find “MySql.Data”. Select the checkbox beside it and press ‘OK”.

Installing the ClientApplicationMVC Project

- You should already have the “ClientApplicationMVC” project created in your solution if you correctly followed the first portion of this document.

Step 1: Remove some autogenerated files and folders.

- Take a look in the source files under “ClientApplicationMVC” folder in the unzipped file from D2L. If there are any files with matching names in the Solution Explorer under the “ClientApplicationMVC” project, delete them from within the solution explorer. They will be replaced with the source files from D2L.

Step 2: Add ClientApplicationMVC source code from d2l to the project

- First copy and paste the source files from the “ClientApplicationMVC” folder in the unzipped file to the directory where your project was created.
- If you are asked if you want to overwrite any files because they already exist, you probably forgot to delete some of the files in step 1.
- Then, just drag each of the added folders and files onto the “ClientApplicationMVC” Project in the solution explorer (drag into the Visual Studio UI) so that visual studio becomes aware of the files.

Step 3: Add missing dependencies to the project.

- Right click on the “References” section under the “ClientApplicationMVC” project in the solution explorer and select “Add Reference…”

- In the window that appears, select “Projects” on the left menu, and select the checkbox beside “Messages” and press “OK”.

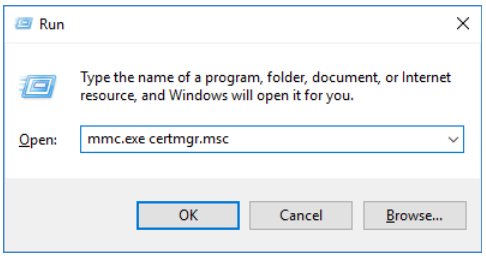
You should be able to compile the program now. In order to run without exceptions we need to change a few more settings.

Enabling Microsoft Message Queue

Microsoft Message queue must be enabled on your machine in order for NServiceBus to run properly. Go to the control panel->Programs->Turn Windows features on or off-> scroll down to find “Microsoft Message Queue (MSMQ) server” and make sure it is selected.

Installing the SSL Certificate.

- Type “mmc.exe certmgr.msc” in the Run window and press OK to open certificate manager:

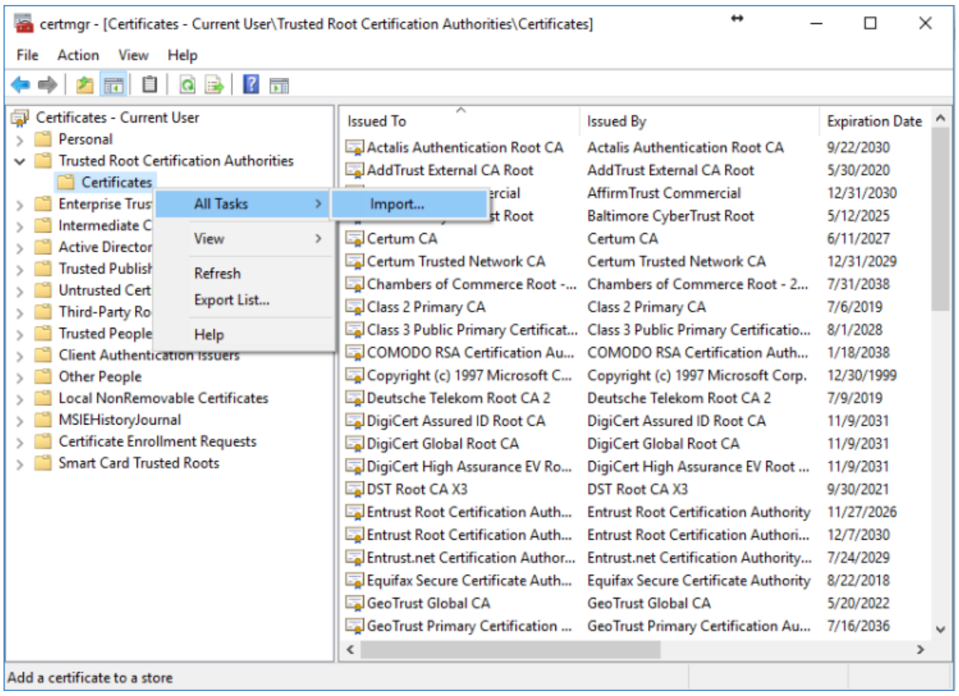


Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: mmc.exe certmgr.msc

- Navigate to “Trusted Root Certification Authorities >> Certificates”. Right-click on Certificates and choose “All Tasks >> Import”:



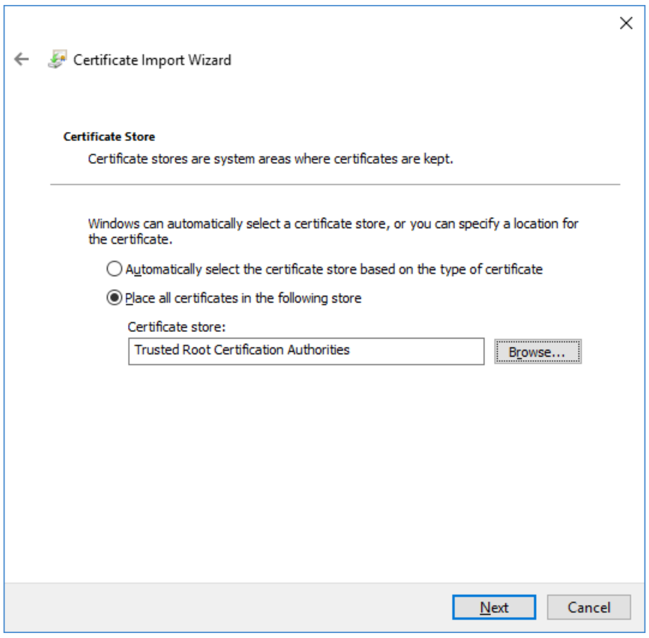
certmgr - [Certificates - Current User\Trusted Root Certification Authorities\Certificates]

Issued To	Issued By	Expiration Date
Actalis Authentication Root CA	Actalis Authentication Root CA	9/22/2030
AddTrust External CA Root	AddTrust External CA Root	5/30/2020
rcial	AffirmTrust Commercial	12/31/2030
t Root	Baltimore CyberTrust Root	5/12/2025
Certum CA	Certum CA	6/11/2027
Certum Trusted Network CA	Certum Trusted Network CA	12/31/2029
Chambers of Commerce Root -...	Chambers of Commerce Root - 2...	7/31/2038
Class 2 Primary CA	Class 2 Primary CA	7/6/2019
Class 3 Public Primary Certificat...	Class 3 Public Primary Certificatio...	8/1/2028
COMODO RSA Certification Au...	COMODO RSA Certification Auth...	1/18/2038
Copyright (c) 1997 Microsoft C...	Copyright (c) 1997 Microsoft Corp.	12/30/1999
Deutsche Telekom Root CA 2	Deutsche Telekom Root CA 2	7/9/2019
DigiCert Assured ID Root CA	DigiCert Assured ID Root CA	11/9/2031
DigiCert Global Root CA	DigiCert Global Root CA	11/9/2031
DigiCert High Assurance EV Ro...	DigiCert High Assurance EV Root ...	11/9/2031
DST Root CA X3	DST Root CA X3	9/30/2021
Entrust Root Certification Auth...	Entrust Root Certification Authority	11/27/2026
Entrust Root Certification Auth...	Entrust Root Certification Authori...	12/7/2030
Entrust.net Certification Author...	Entrust.net Certification Authority...	7/24/2029
Equifax Secure Certificate Auth...	Equifax Secure Certificate Authority	8/22/2018
GeoTrust Global CA	GeoTrust Global CA	5/20/2022
GeoTrust Primary Certification ...	GeoTrust Primary Certification Au...	7/16/2036

Add a certificate to a store

- Click Next. Then Browse to the location of the D2L folder and go to certificates->SENG401TenYears.pfx (make sure to choose All files as the file format filter to show all file types in the Open dialog).
- Click Next.
- You will be asked for a password. However, there is no password and you can simply press enter.

- Select “Place all certificates in the following store” and click Browse and choose “Trusted Root Certification Authorities”:



Certificate Import Wizard

Certificate Store

Certificate stores are system areas where certificates are kept.

Windows can automatically select a certificate store, or you can specify a location for the certificate.

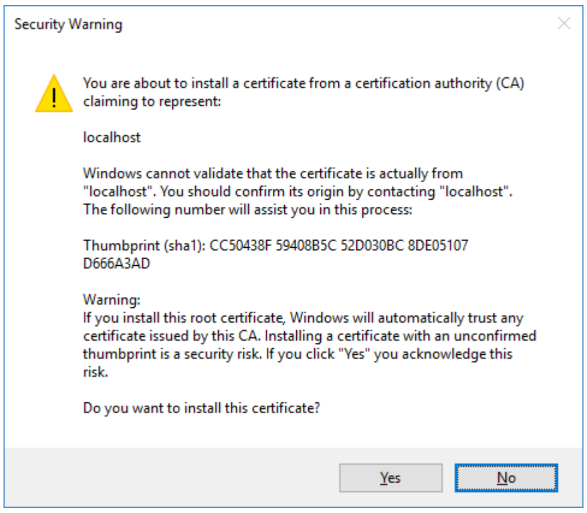
○ Automatically select the certificate store based on the type of certificate

● Place all certificates in the following store

Certificate store: Trusted Root Certification Authorities

- Click Next. In the next step, review the information and if they are correct click Finish.

Note: If you receive a warning after clicking the Finish button, read the warning. Press Yes to acknowledge that the certificate is self-generated.



Security Warning

You are about to install a certificate from a certification authority (CA) claiming to represent:

localhost

Windows cannot validate that the certificate is actually from "localhost". You should confirm its origin by contacting "localhost". The following number will assist you in this process:

Thumbprint (sha1): CC50438F 59408B5C 52D030BC 8DE05107 D666A3AD

Warning:
If you install this root certificate, Windows will automatically trust any certificate issued by this CA. Installing a certificate with an unconfirmed thumbprint is a security risk. If you click "Yes" you acknowledge this risk.

Do you want to install this certificate?

- Now navigate to the Server.cs class in the authentication service. Scroll to the bottom of the file and change the “certificateLocation” property to the location of the SENG401TenYears.pfx file on your machine.

The program should now be able to run without any errors. You will need to repeat this and the above MSMQ steps on your deployment machine before you deploy the project

Other convenient settings:

- For convenience, it is suggested that you change your startup programs so that when you press start all of your projects will be automatically built. Do this by right clicking on your call it local solution in the solution explorer and selecting properties. Then look under Common Properties->Startup Project.
- Select “Multiple Startup Projects” and select the “start” option for every project except for the Messages library.