WorkshopPLUS - Essentials on Azure DevOps Services and GitHub

Lab Guides

Conditions and Terms of Use

Microsoft Confidential

Copyright and Trademarks

Module 5: Azure Artifacts

Lab 1: Azure Artifacts

Introduction

Azure Artifacts makes it easy to discover, install, and publish NuGet, npm, and Maven packages in Azure DevOps. It's deeply integrated with other hubs like Pipelines so that Azure Artifacts can become a seamless part of your existing workflows.

Exercise 1: Working with the Azure Artifacts service

Objectives

After completing this lab, you will be able to:

- Create a feed in Azure DevOps Services.
- Create and publish a NuGet package to a feed.
- Update an existing NuGet package.

Prerequisites

- Visual Studio 2017/2019
- Lab 2: PartsUnlimited Lab Setup

Estimated Time to Complete This Lab

45 minutes

Module 5: **Azure Artifacts**, **Lab 1: Azure Artifacts**, Exercise 1: Working with Azure Artifacts

---

Exercise 1: Working with the Azure Artifacts service

Objectives

Azure Artifacts is an extension that makes it easy to discover, install, and publish NuGet, npm, and Maven packages in Azure DevOps. It's deeply integrated with other hubs like Build so that Azure Artifacts can become a seamless part of your existing workflows. In this exercise, you will configure and work with Azure Artifacts. You will learn how to:

- Create a feed in Azure DevOps Services.
- Create and publish a NuGet package to a feed.
- Update an existing NuGet package.

Tasks

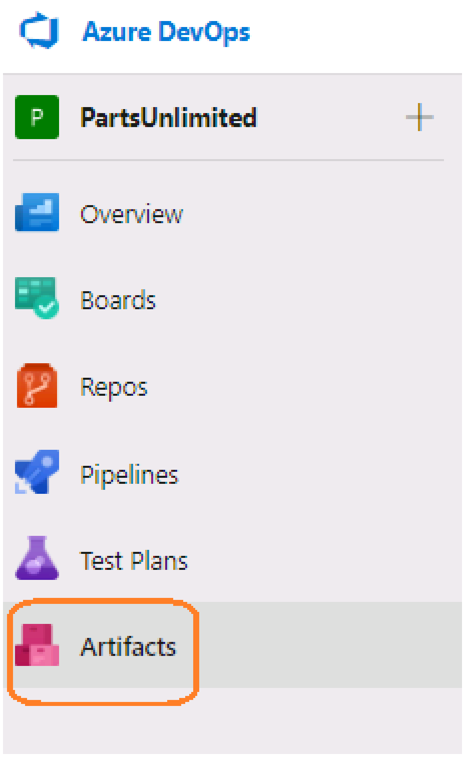
Module 5: **Azure Artifacts**, **Lab 1: Azure Artifacts**, Exercise 1: Working with Azure Artifacts

---

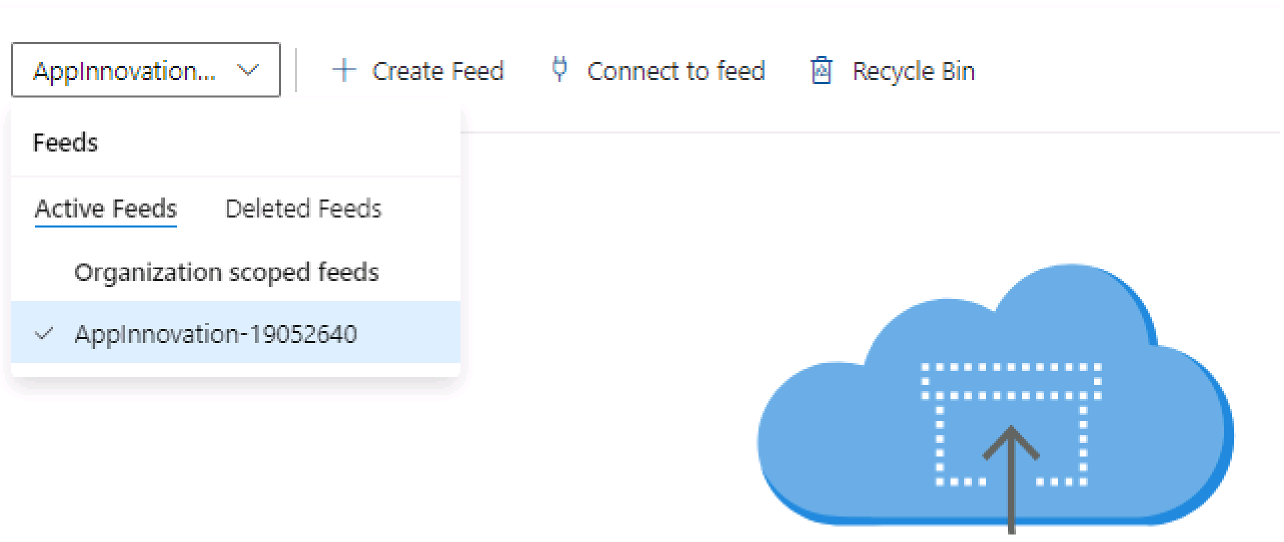
Task 1: Creating and connecting to a feed

1. Navigate to the **PartsUnlimited** project in **Azure DevOps Services**.

2. Navigate to the **Artifacts** hub.



3. You should see a feed with the name of your organization. For the purposes of this lab, we will create a new project-scoped feed.



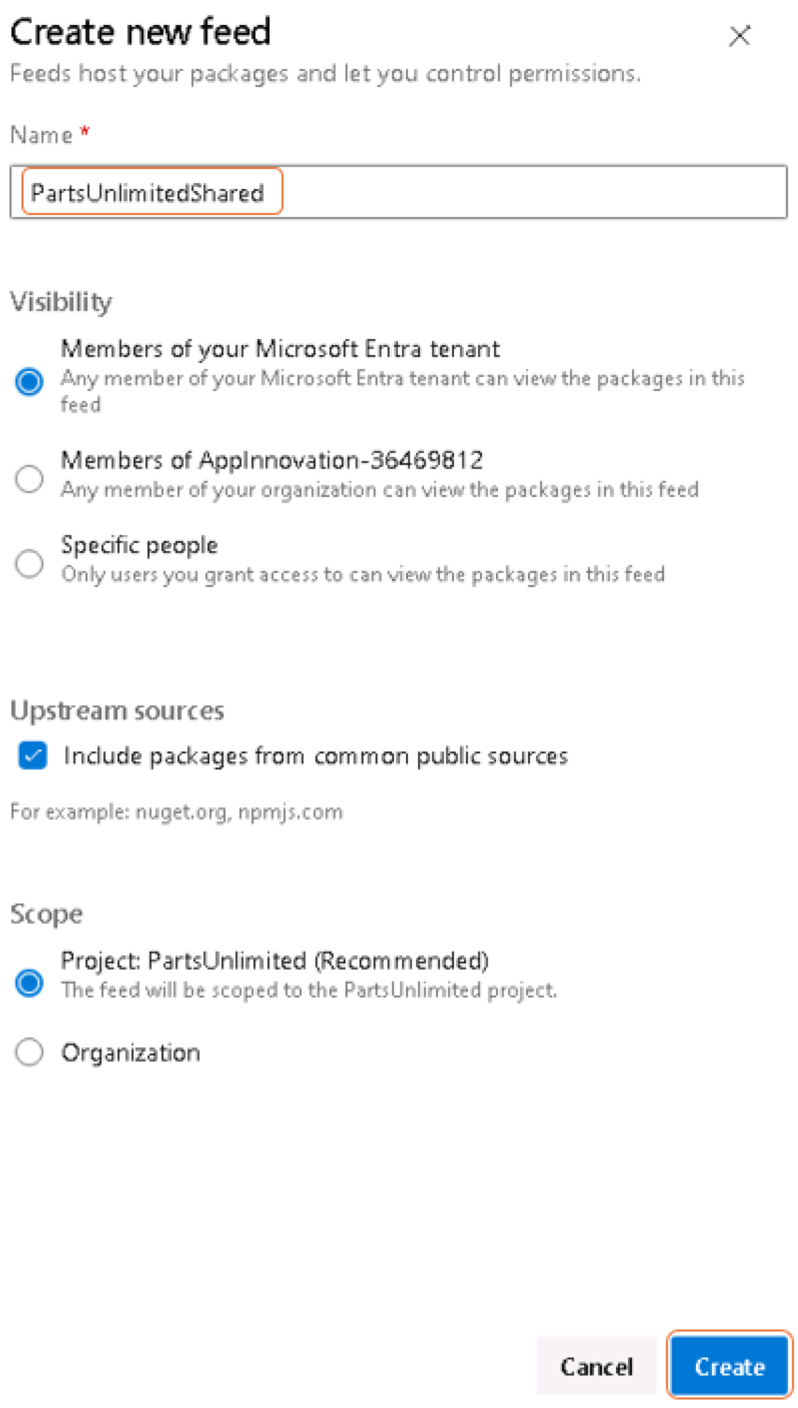
> Historically, all feeds used to be scoped to an organization. However, to enable public feeds and to become more consistent with the rest of Azure DevOps, Feeds created through the new *Create Feed* panel will now be scoped to a project. New organization will automatically have one

> feed scoped to the organization, and all subsequent feeds created will be scoped to a project. All existing organization-scoped feeds will remain organization-scoped.

4. Click **+ Create Feed**.

> This feed will be a collection of NuGet packages available to users scoped to the PartsUnlimited project. The scenario in this lab will focus on the workflow for using Azure Artifacts service, so the actual architectural and development decisions are purely illustrative.

5. In the **Create new feed** pane, set the **Name** to **PartsUnlimitedShared** and click **Create**. Leave other default options.



Create new feed

Feeds host your packages and let you control permissions.

Name *

PartsUnlimitedShared

Visibility

- [✓ Members of your Microsoft Entra tenant] Any member of your Microsoft Entra tenant can view the packages in this feed
- Members of AppInnovation-36469812 Any member of your organization can view the packages in this feed
- Specific people Only users you grant access to can view the packages in this feed

Upstream sources

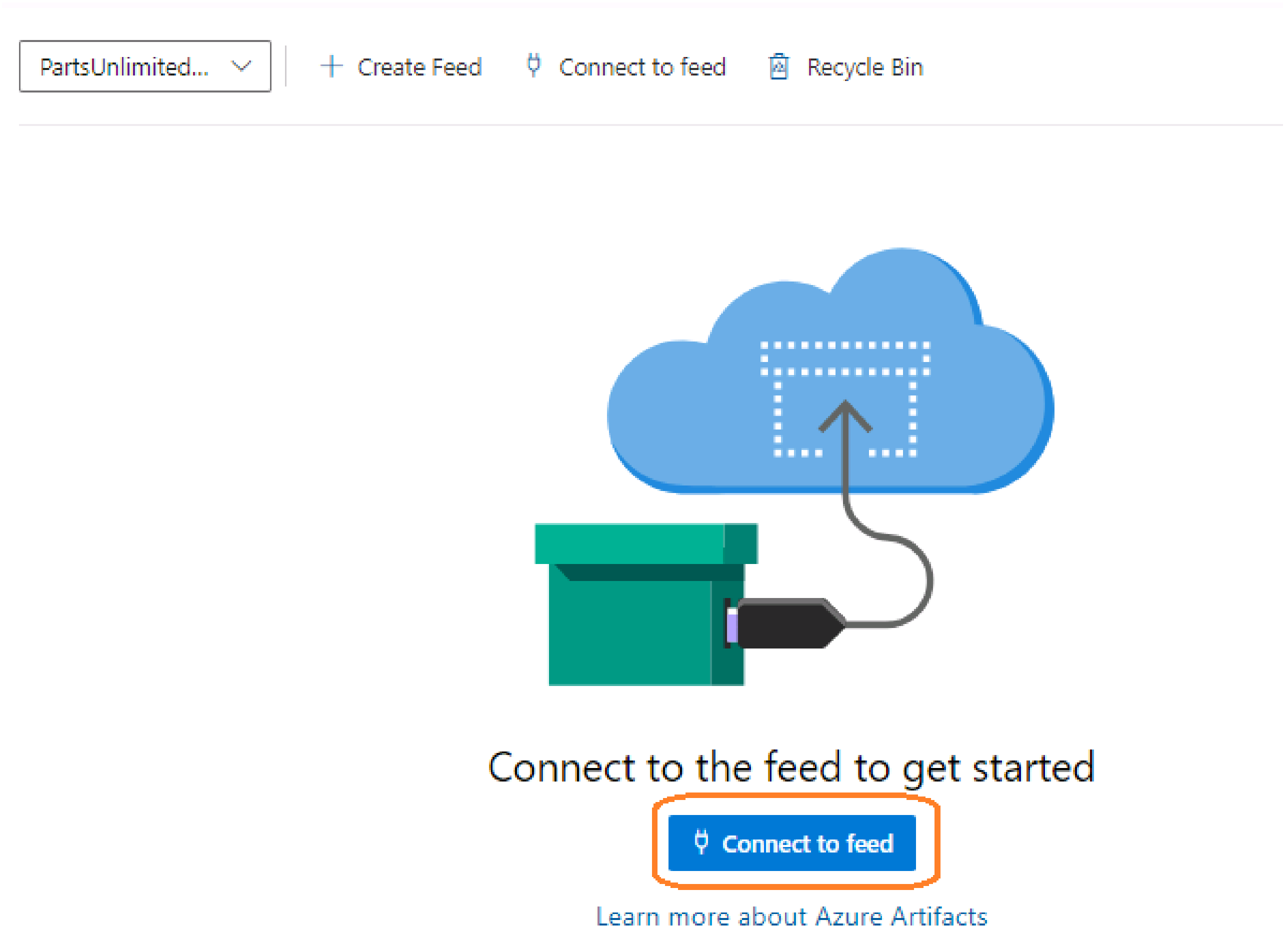
✓ Include packages from common public sources

For example: nuget.org, npmjs.com

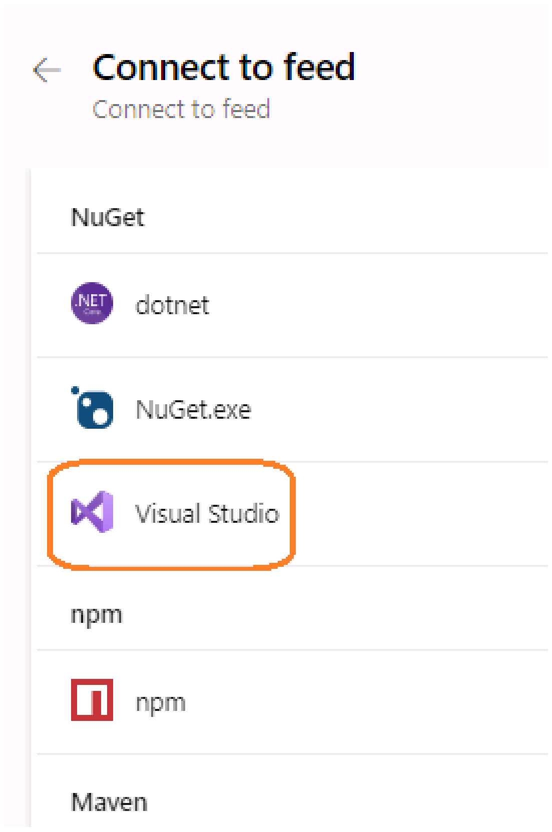
Scope

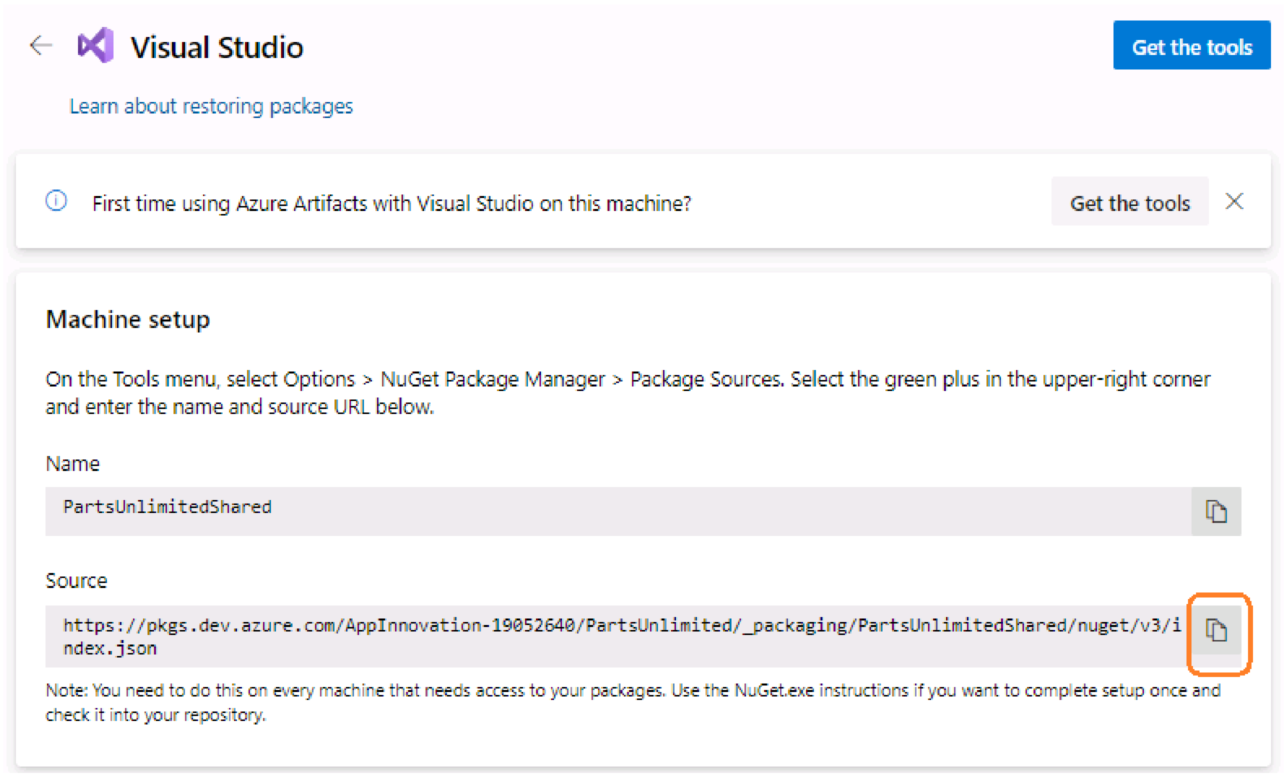
- [✓ Project: PartsUnlimited (Recommended)] The feed will be scoped to the PartsUnlimited project.
- Organization

6. Any user who wants to connect to this NuGet feed must configure their environment. Click **Connect to feed**.



7. Select **Visual Studio** from the **Connect to feed** screen and copy the package **Source URL**. This is the only thing Visual Studio and NuGet need to start taking advantage of the new feed. Leave the dialog open in the browser.





First time using Azure Artifacts with Visual Studio on this machine?

Machine setup

On the Tools menu, select Options > NuGet Package Manager > Package Sources. Select the green plus in the upper-right corner and enter the name and source URL below.

Name

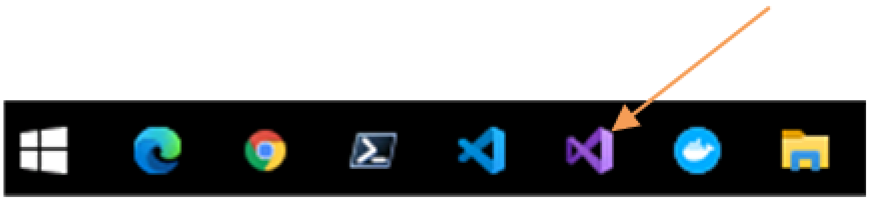
```
PartsUnlimitedShared
```

Source

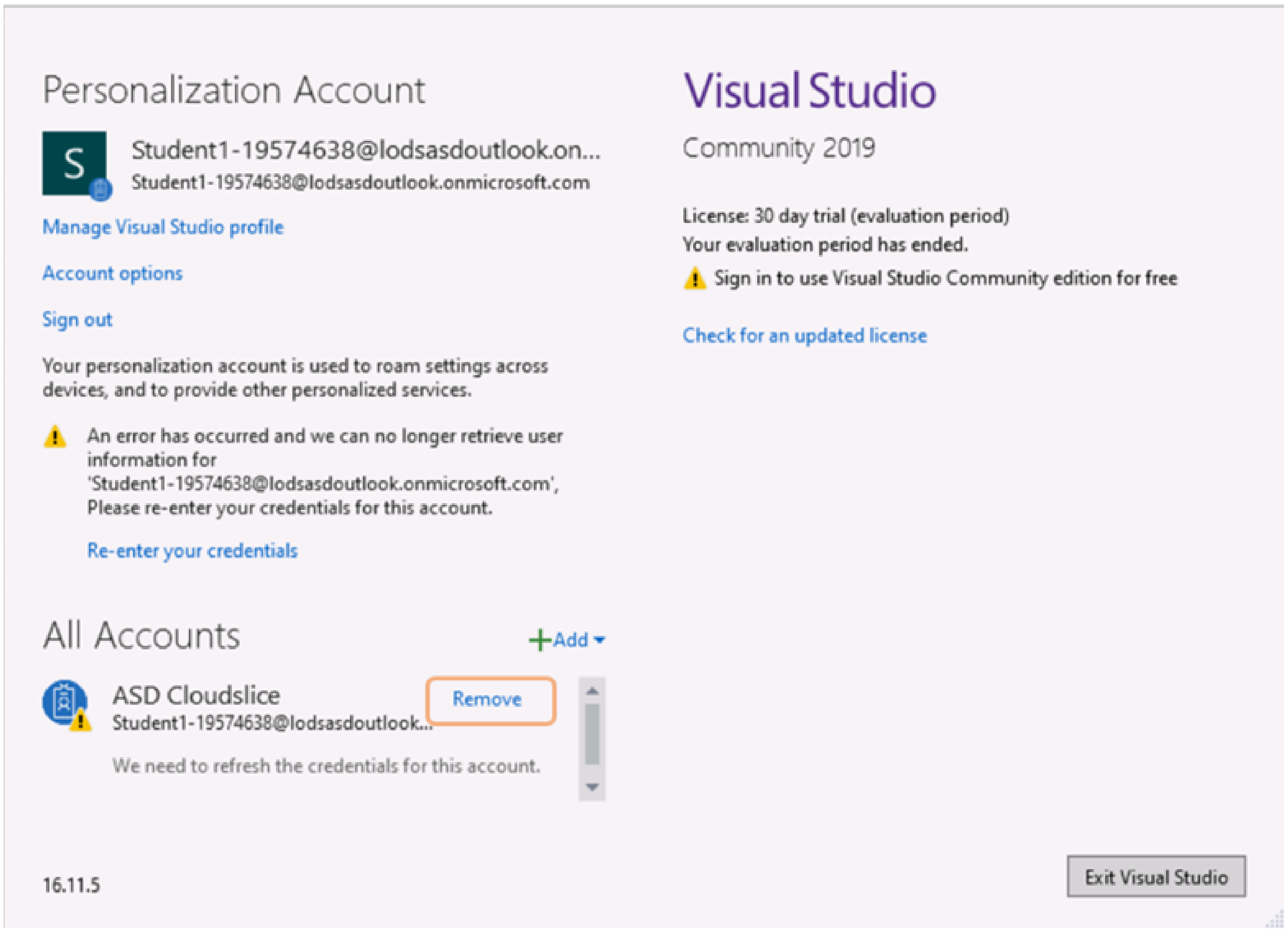
```
https://pkgs.dev.azure.com/AppInnovation-19052640/PartsUnlimited/_packaging/PartsUnlimitedShared/nuget/v3/index.json
```

Note: You need to do this on every machine that needs access to your packages. Use the NuGet.exe instructions if you want to complete setup once and check it into your repository.

8. Start **Visual Studio** from the taskbar.



9. If you see a pop up window asking for an updated license, first **Remove** the existing account (Click on **Yes** if you are asked for a confirmation), and **Sign In** with your student account (You might be asked to enter the username and password one more time). Finally, click on the link to **Check for an updated license**, confirm that the license was successfully updated, and then **Close** the window.



Personalization Account

Student1-19574638@lodsasdoutlook.on...
Student1-19574638@lodsasdoutlook.onmicrosoft.com

Manage Visual Studio profile

Account options

Sign out

Your personalization account is used to roam settings across devices, and to provide other personalized services.

An error has occurred and we can no longer retrieve user information for 'Student1-19574638@lodsasdoutlook.onmicrosoft.com', Please re-enter your credentials for this account.

Re-enter your credentials

All Accounts

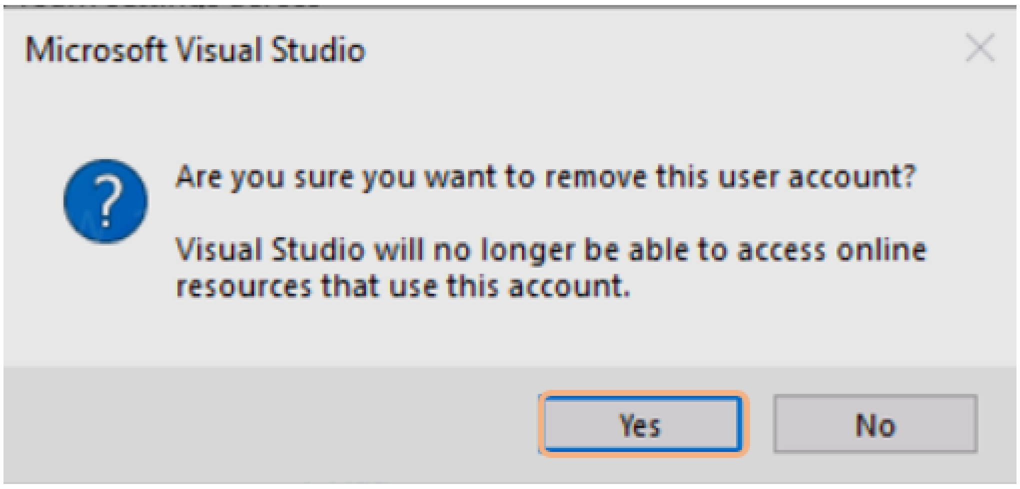
ASD Cloudslice
Student1-19574638@lodsasdoutlook...
We need to refresh the credentials for this account.

Visual Studio
Community 2019

License: 30 day trial (evaluation period)
Your evaluation period has ended.
Sign in to use Visual Studio Community edition for free

Check for an updated license

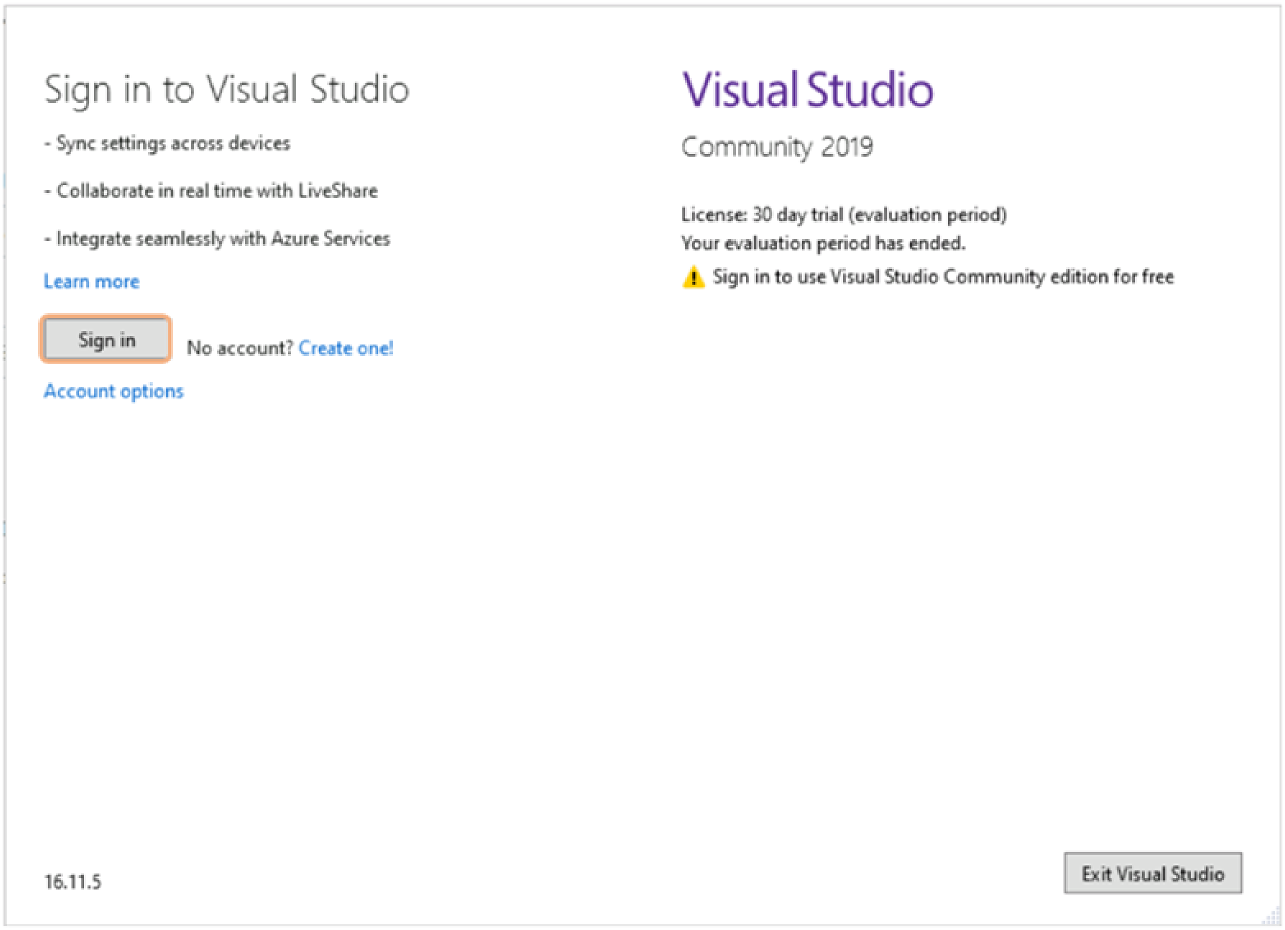
16.11.5



Microsoft Visual Studio

Are you sure you want to remove this user account?

Visual Studio will no longer be able to access online resources that use this account.


Sign in to Visual Studio

- Sync settings across devices
- Collaborate in real time with LiveShare
- Integrate seamlessly with Azure Services

Learn more

Sign in No account? Create one!

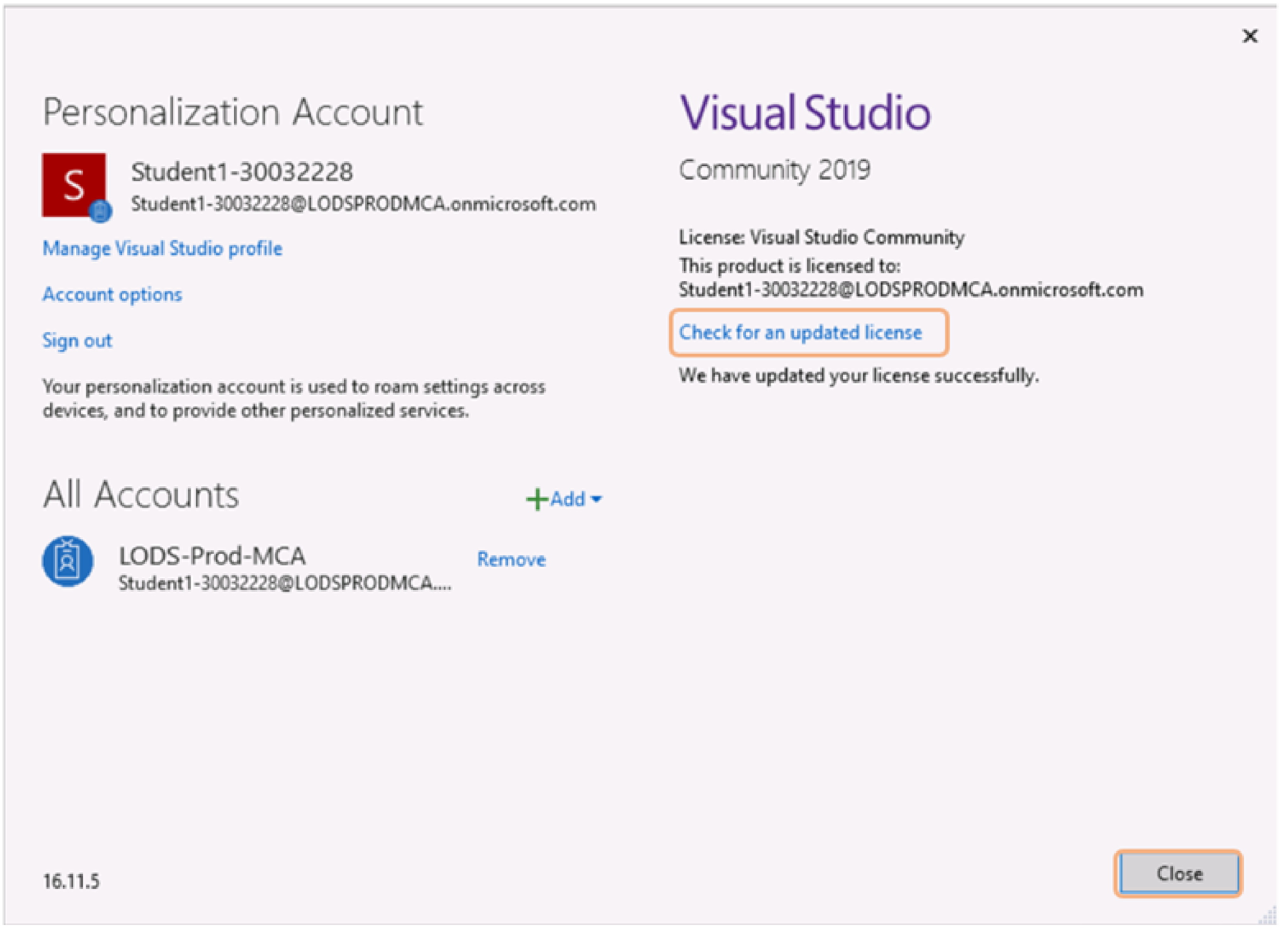
Account options

Visual Studio
Community 2019

License: 30 day trial (evaluation period)
Your evaluation period has ended.
Sign in to use Visual Studio Community edition for free

16.11.5

Exit Visual Studio


Personalization Account

Student1-30032228
Student1-30032228@LODSPRODMCA.onmicrosoft.com

Manage Visual Studio profile

Account options

Sign out

Your personalization account is used to roam settings across devices, and to provide other personalized services.

All Accounts +Add

LODS-Prod-MCA
Student1-30032228@LODSPRODMCA.... Remove

Visual Studio
Community 2019

License: Visual Studio Community
This product is licensed to:
Student1-30032228@LODSPRODMCA.onmicrosoft.com

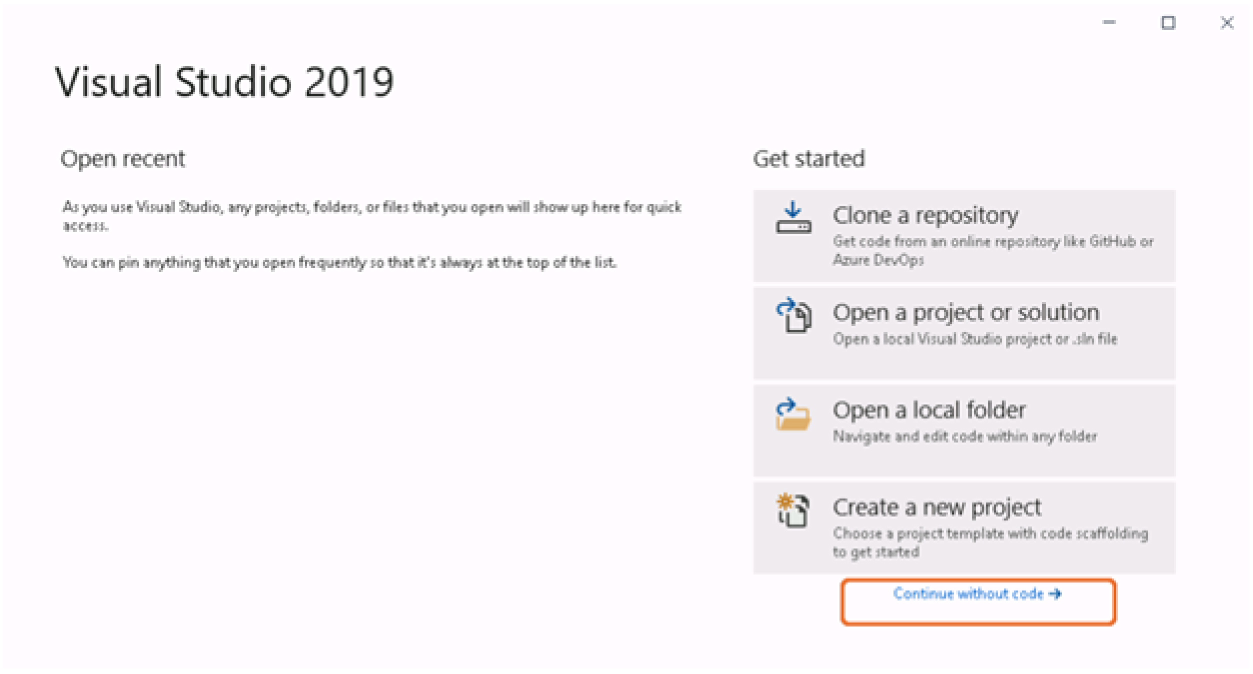
Check for an updated license

We have updated your license successfully.

16.11.5

Close

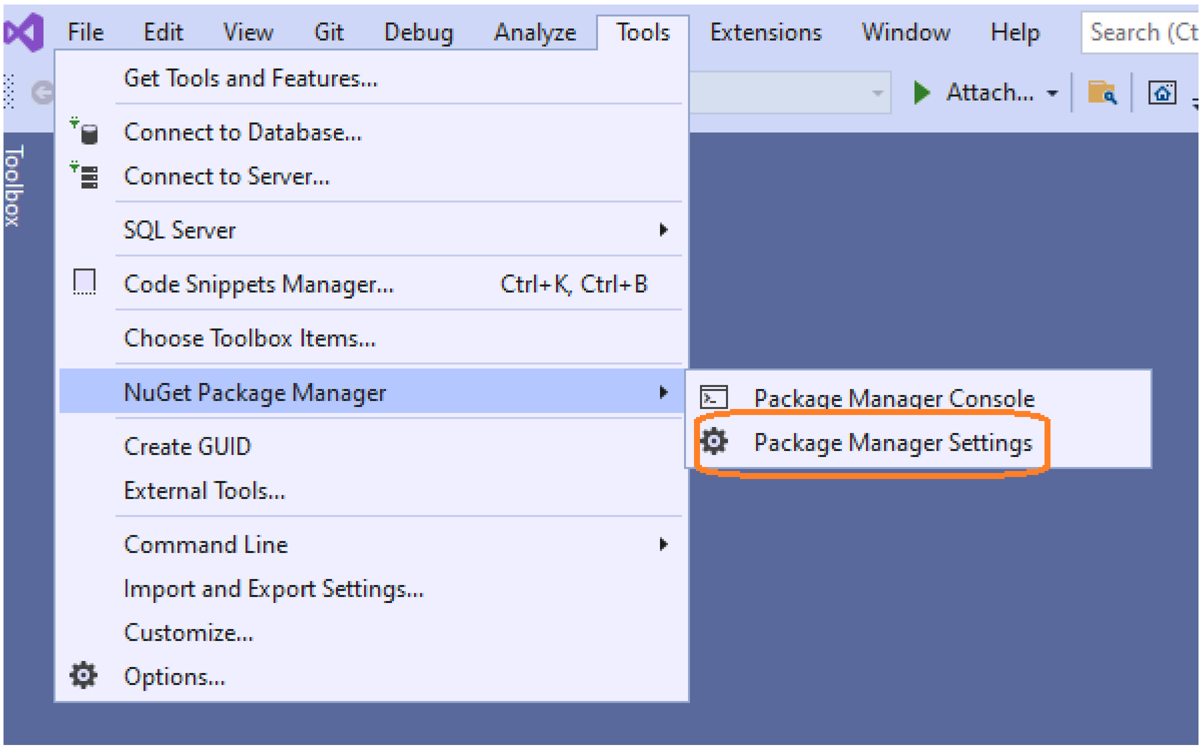
10. Select **Continue without code**.


Visual Studio 2019
Open recent
Get started
Clone a repository
Get code from an online repository like GitHub or Azure DevOps
Open a project or solution
Open a local Visual Studio project or .sln file
Open a local folder
Navigate and edit code within any folder
Create a new project
Choose a project template with code scaffolding to get started
Continue without code →

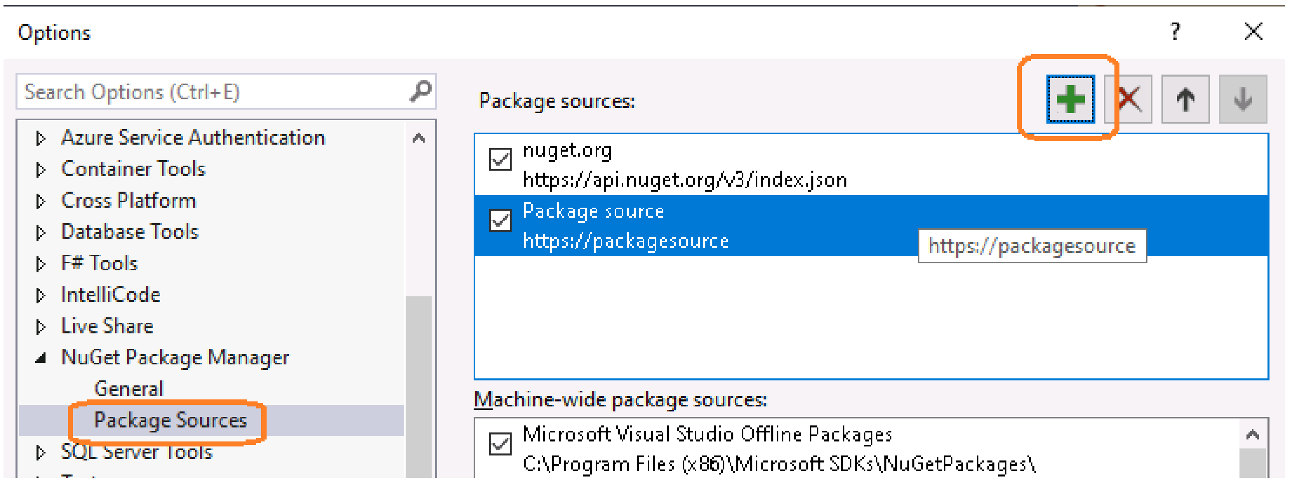
As you use Visual Studio, any projects, folders, or files that you open will show up here for quick access.

You can pin anything that you open frequently so that it's always at the top of the list.

11. Select **Tools | NuGet Package Manager | Package Manager Settings**.



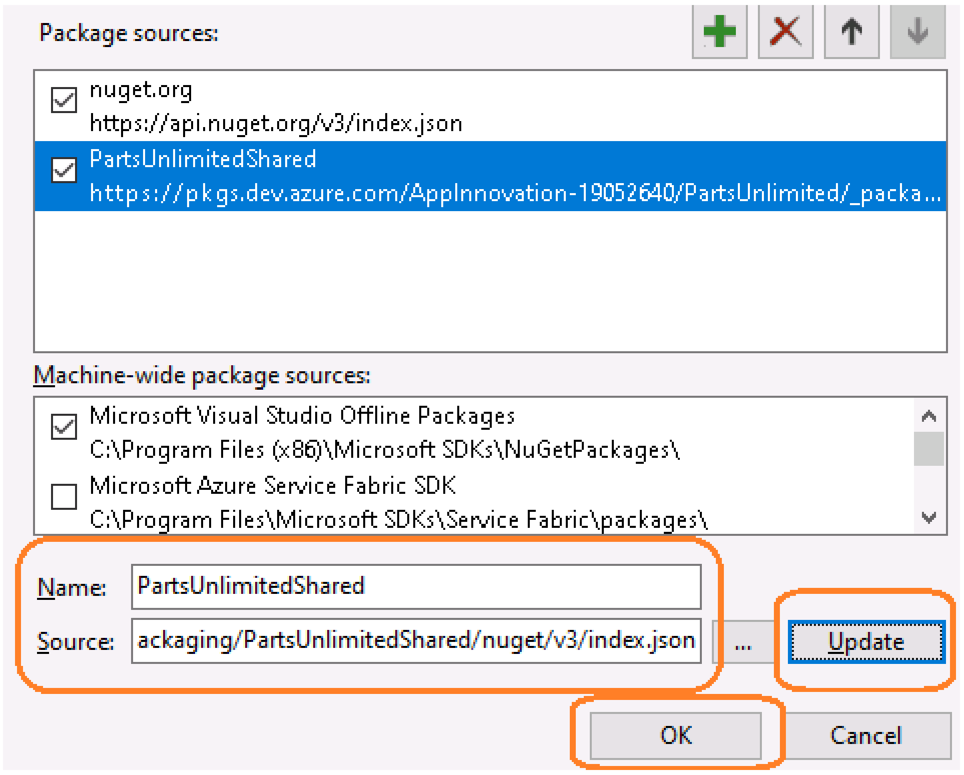
12. Locate the **Package Sources** section and click the **Add button** to add a new package source.



nuget.org
https://api.nuget.org/v3/index.json
Package source
https://packagesource

Microsoft Visual Studio Offline Packages
C:\Program Files (x86)\Microsoft SDKs\NuGetPackages\

13. Set the **Name** to **PartsUnlimitedShared** and paste the **Source URL** copied earlier in the **Source** field. Click **Update** followed by **OK**. Visual Studio is now connected to the new feed.



nuget.org
https://api.nuget.org/v3/index.json
PartsUnlimitedShared
https://pkgs.dev.azure.com/AppInnovation-19052640/PartsUnlimited/_packa...

Microsoft Visual Studio Offline Packages
C:\Program Files (x86)\Microsoft SDKs\NuGetPackages\
Microsoft Azure Service Fabric SDK
C:\Program Files\Microsoft SDKs\Service Fabric\packages\

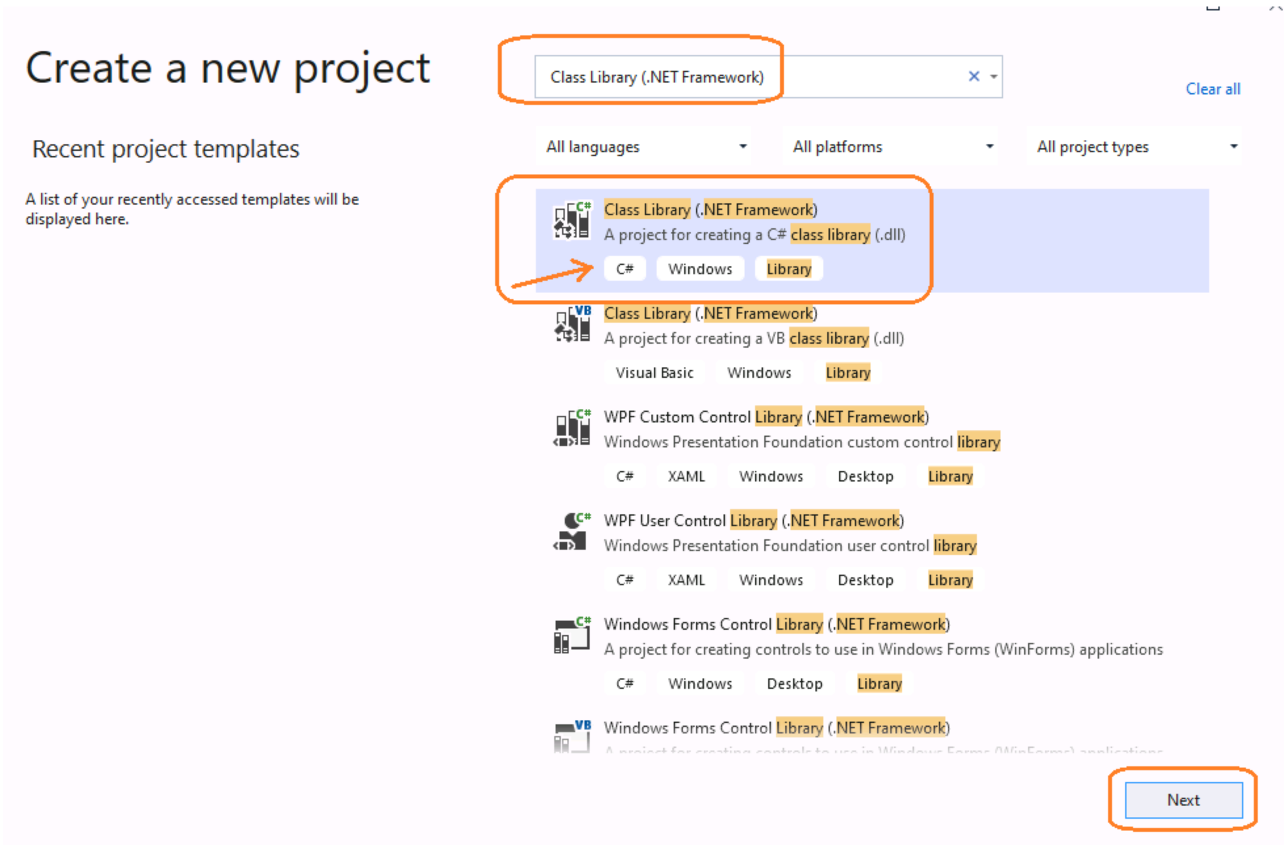
Name: PartsUnlimitedShared
Source: ackaging/PartsUnlimitedShared/nuget/v3/index.json

Module 5: **Azure Artifacts**, **Lab 1: Azure Artifacts**, Exercise 1: Working with Azure Artifacts

Task 2: Creating and publishing a NuGet package

1. In **Visual Studio** (the most recent one where you added the package source), from the main menu, select **File | New | Project**. You will be creating a shared assembly that will be published as a NuGet package so that other teams can integrate it and stay up to date without having to work directly with the project source.

2. In the **Create a new project** window, enter **Class Library (.NET Framework)** in the search bar and select the one with **C#**. Click **Next**.



Create a new project

Class Library (.NET Framework)

Clear all

Recent project templates

A list of your recently accessed templates will be displayed here.

All languages · All platforms · All project types

Class Library (.NET Framework)
A project for creating a C# class library (.dll)
C# · Windows · Library

Class Library (.NET Framework)
A project for creating a VB class library (.dll)
Visual Basic · Windows · Library

WPF Custom Control Library (.NET Framework)
Windows Presentation Foundation custom control library
C# · XAML · Windows · Desktop · Library

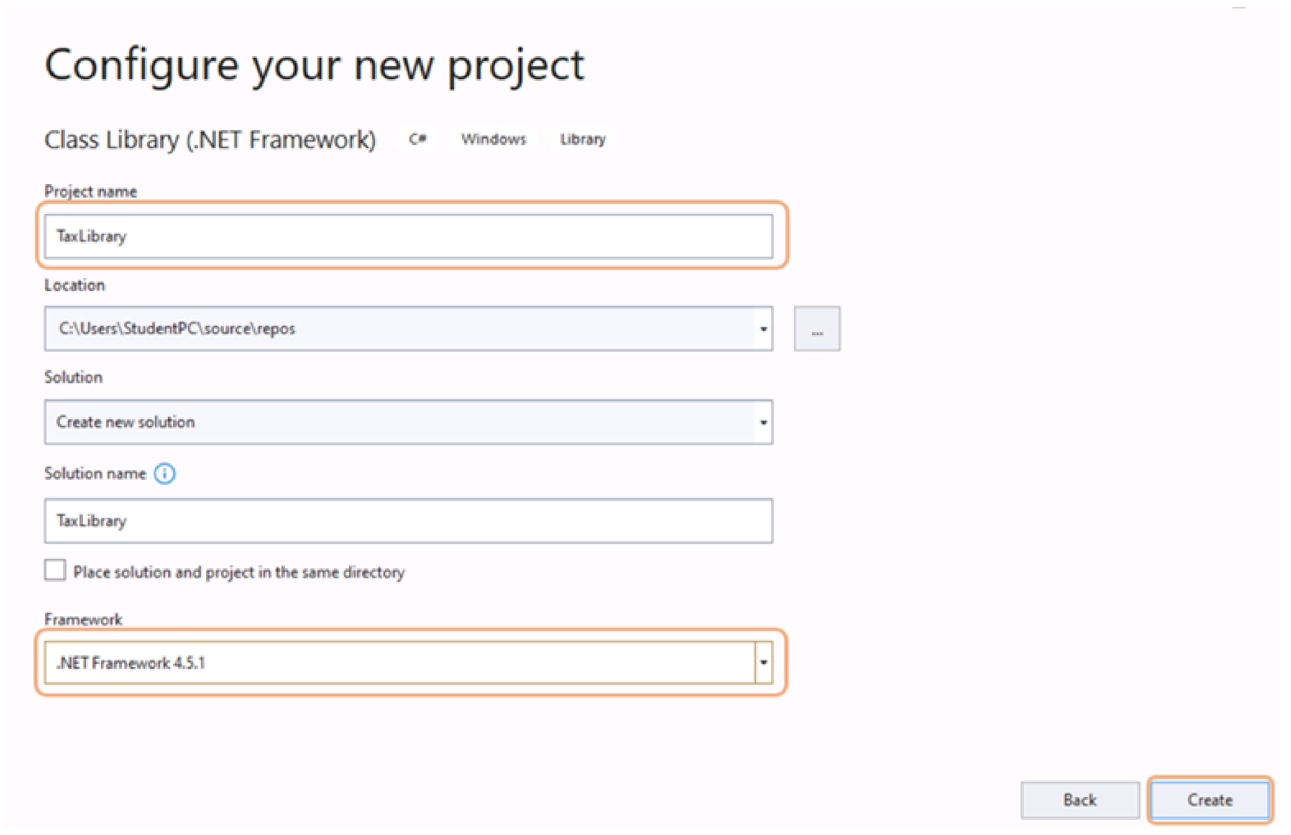
WPF User Control Library (.NET Framework)
Windows Presentation Foundation user control library
C# · XAML · Windows · Desktop · Library

Windows Forms Control Library (.NET Framework)
A project for creating controls to use in Windows Forms (WinForms) applications
C# · Windows · Desktop · Library

Windows Forms Control Library (.NET Framework)

Next

3. Select the **Project name** as **TaxLibrary** and **Framework** as **.NET Framework 4.5.1**. Click **Create**.



Configure your new project

Class Library (.NET Framework)

Project name: TaxLibrary

Location: C:\Users\StudentPC\source\repos

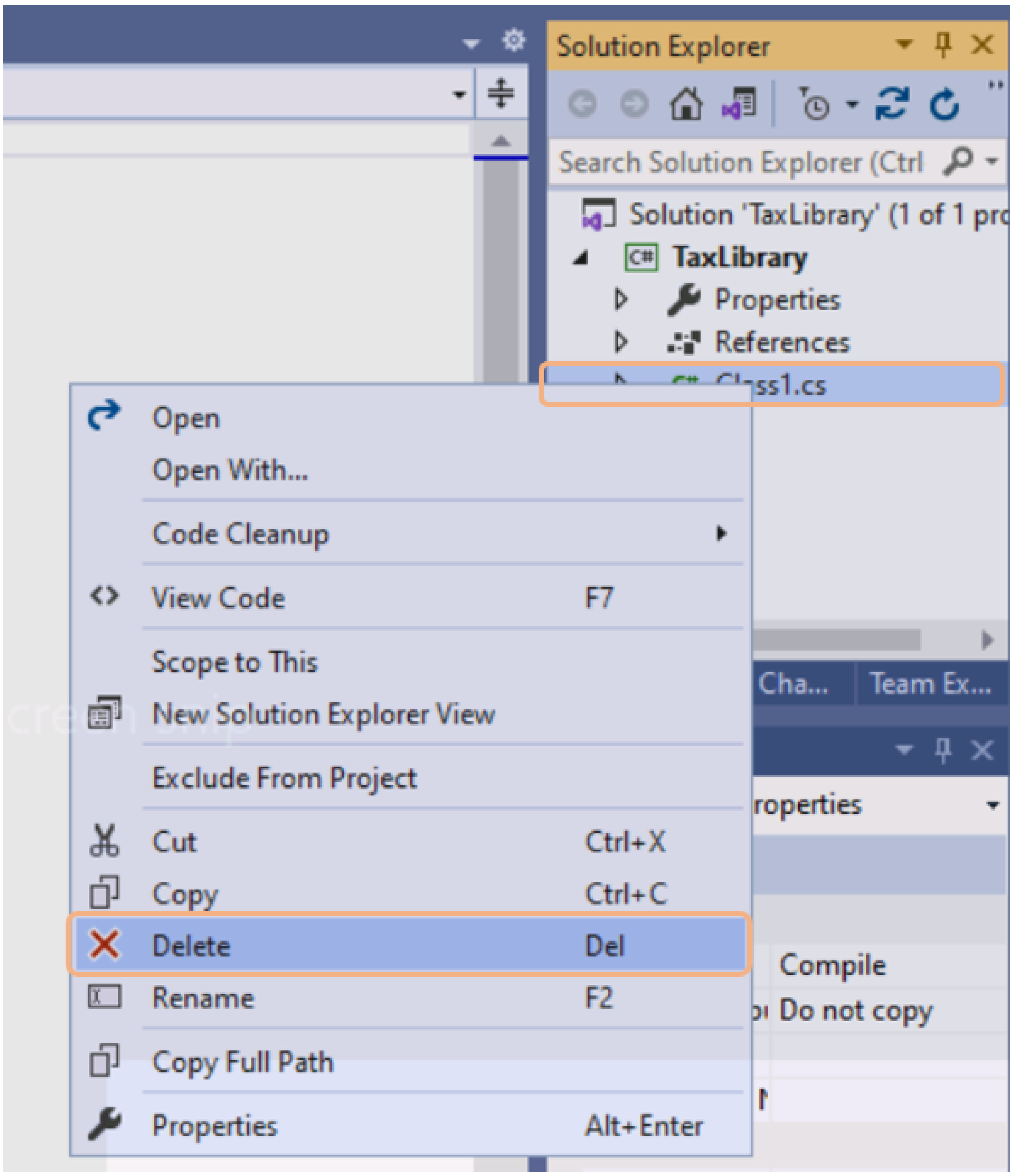
Solution: Create new solution

Solution name: TaxLibrary

☐ Place solution and project in the same directory

Framework: .NET Framework 4.5.1

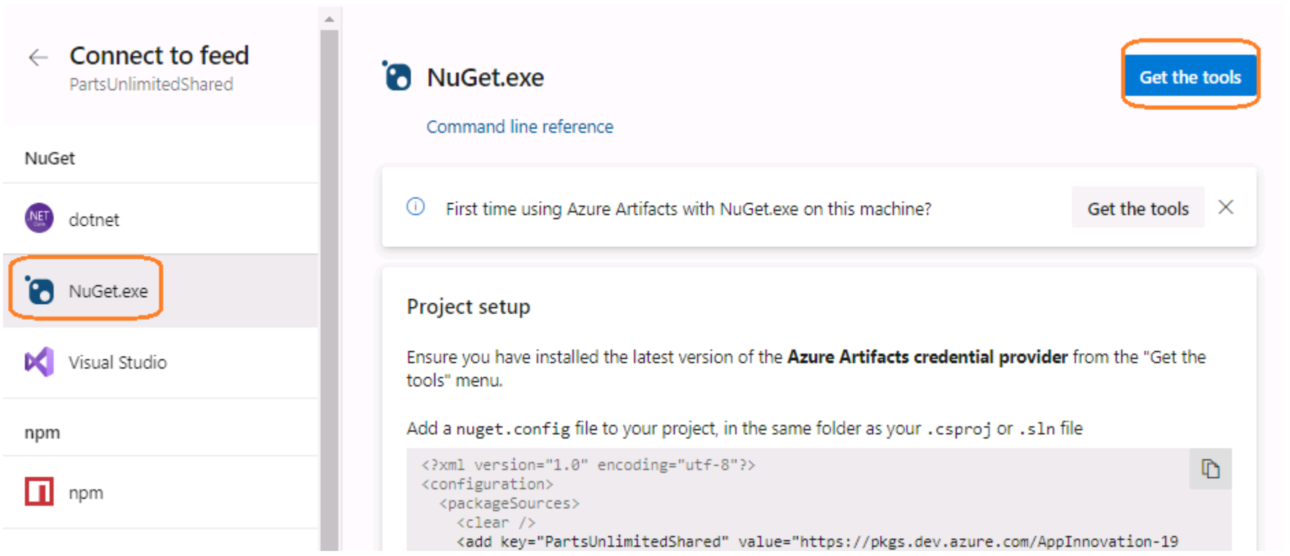
4. In the Solution Explorer, **delete Class1.cs**. Confirm the delete action.



5. Press **Ctrl+Shift+B** to build the project.

> In the next task we'll use NuGet.exe to generate a NuGet package directly from the built project, but it requires the project to be built first.

6. Return back to Azure DevOps Services and select **NuGet.exe**. Select **Get the tools** and click on **Download the latest NuGet**.



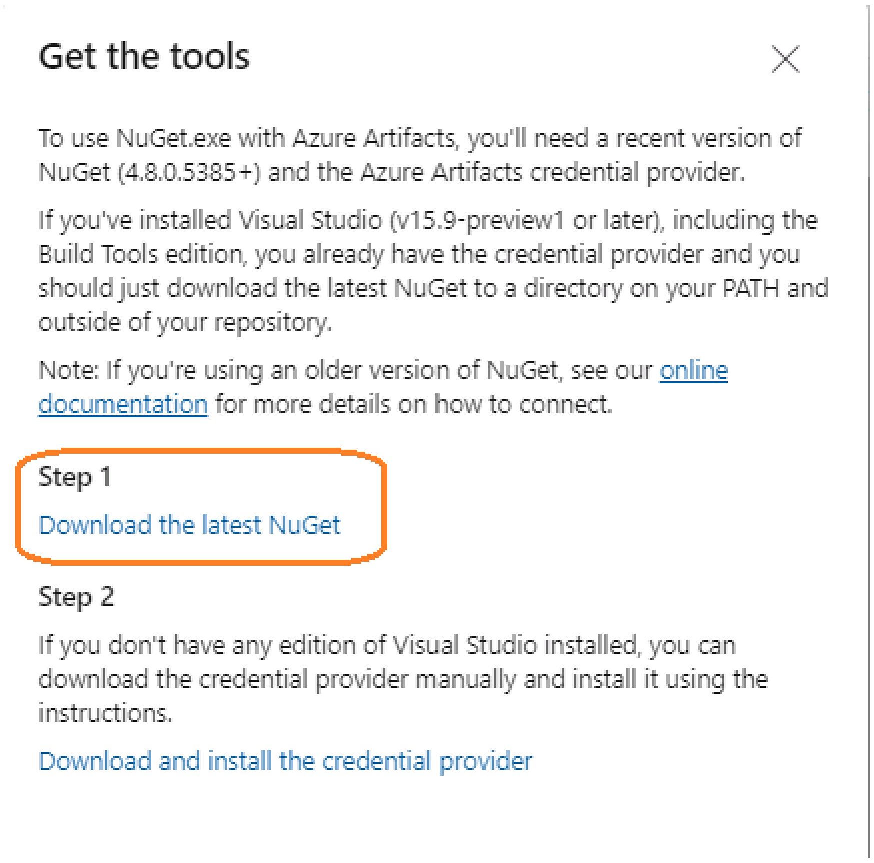
First time using Azure Artifacts with NuGet.exe on this machine?

Project setup

Ensure you have installed the latest version of the **Azure Artifacts credential provider** from the "Get the tools" menu.

Add a `nuget.config` file to your project, in the same folder as your `.csproj` or `.sln` file

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <packageSources>
    <clear />
    <add key="PartsUnlimitedShared" value="https://pkgs.dev.azure.com/AppInnovation-19
```



Get the tools

To use NuGet.exe with Azure Artifacts, you'll need a recent version of NuGet (4.8.0.5385+) and the Azure Artifacts credential provider.

If you've installed Visual Studio (v15.9-preview1 or later), including the Build Tools edition, you already have the credential provider and you should just download the latest NuGet to a directory on your PATH and outside of your repository.

Note: If you're using an older version of NuGet, see our online documentation for more details on how to connect.

Step 1

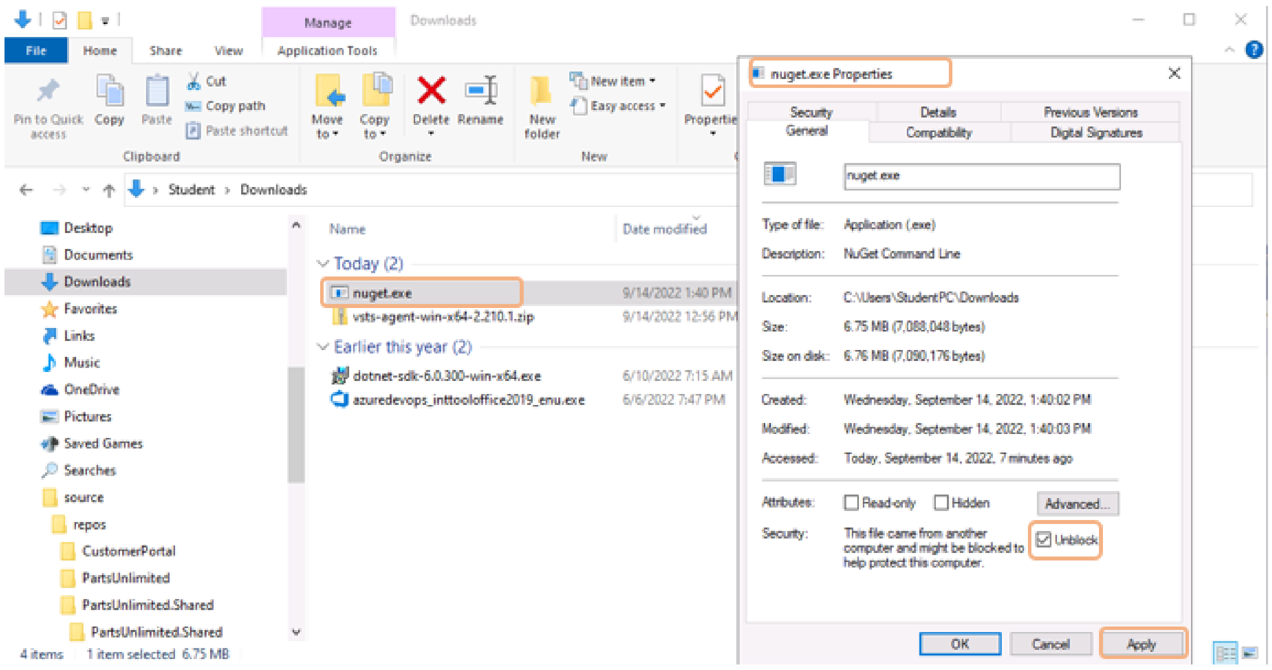
Download the latest NuGet

Step 2

If you don't have any edition of Visual Studio installed, you can download the credential provider manually and install it using the instructions.

Download and install the credential provider

7. Click on the latest nuget.exe to download it. Open the folder where nuget.exe is downloaded. **Right click** to the nuget.exe and select **Properties** from the menu. Select the **Unblock** option and **Apply** the change.

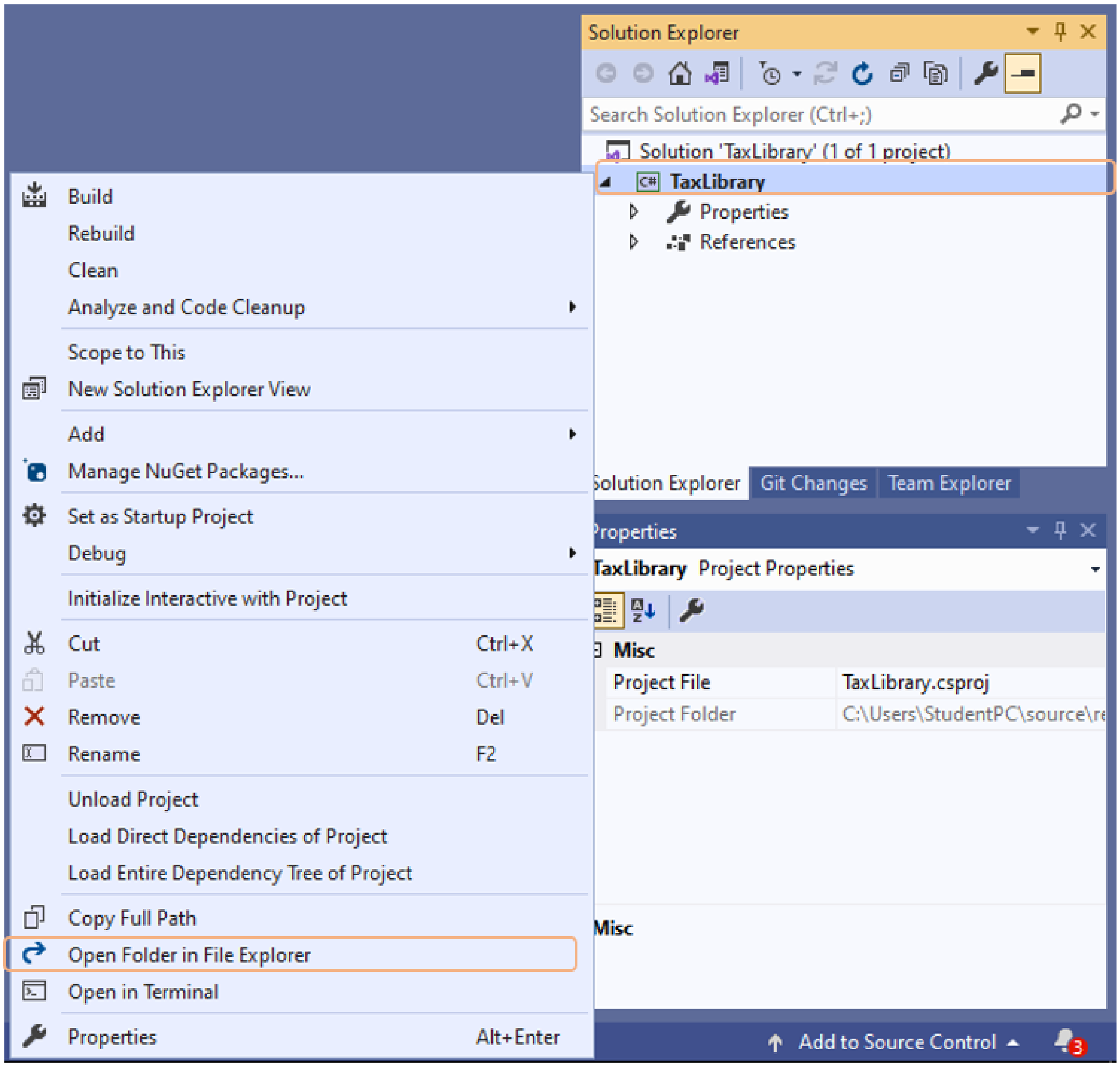


Name	Date modified
nuget.exe	9/14/2022 1:40 PM
vsts-agent-win-x64-2.210.1.zip	9/14/2022 12:56 PM
dotnet-sdk-6.0.300-win-x64.exe	6/10/2022 7:15 AM
azuredevops_inttooloffice2019_enu.exe	6/6/2022 7:47 PM

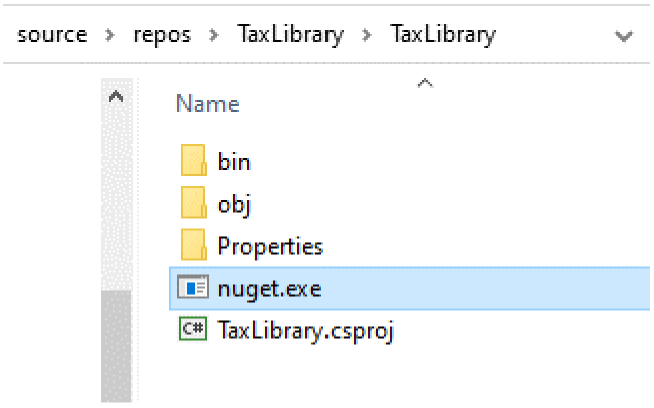
nuget.exe Properties

	nuget.exe
Type of file:	Application (.exe)
Description:	NuGet Command Line
Location:	C:\Users\StudentPC\Downloads
Size:	6.75 MB (7,088,048 bytes)
Size on disk:	6.76 MB (7,090,176 bytes)
Created:	Wednesday, September 14, 2022, 1:40:02 PM
Modified:	Wednesday, September 14, 2022, 1:40:03 PM
Accessed:	Today, September 14, 2022, 7 minutes ago
Attributes:	☐ Read-only ☐ Hidden Advanced...
Security:	This file came from another computer and might be blocked to help protect this computer. ☑ Unblock

8. Return to **Visual Studio** with **TaxLibrary solution** open. From the **Solution Explorer**, right-click the TaxLibrary project node and select **Open Folder in File Explorer**.



9. Move the downloaded nuget.exe file into the folder containing the .csproj file.



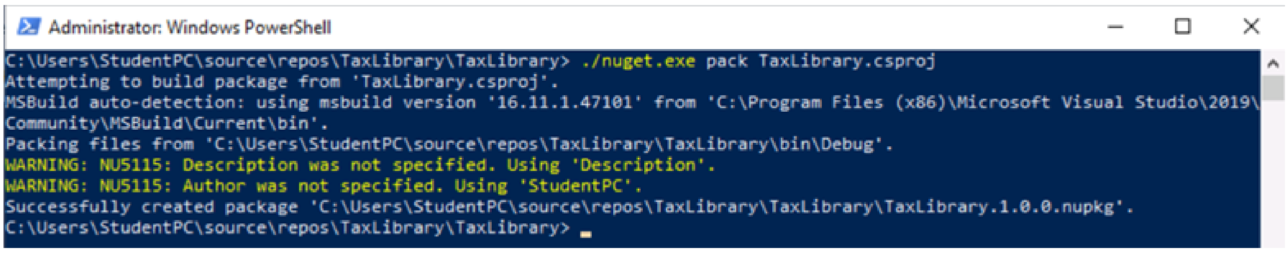
10. In the same Windows Explorer window, select **File | Open Windows PowerShell | Open Windows PowerShell as administrator**.

11. Execute the following command to create a .nupkg file from the project.

> This is a quick shortcut to package the NuGet bits together for deployment. NuGet is very customizable and offers a lot of great flexibility for providing detailed information for

> consumers. You can learn more over on the NuGet package creation page.

```
./nuget.exe pack TaxLibrary.csproj
```


Administrator: Windows PowerShell

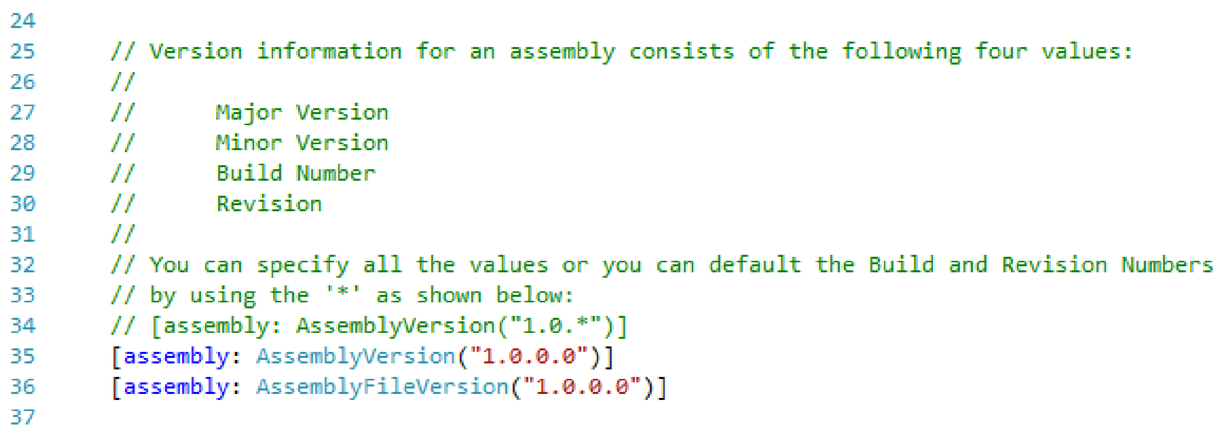
```
C:\Users\StudentPC\source\repos\TaxLibrary\TaxLibrary> ./nuget.exe pack TaxLibrary.csproj
Attempting to build package from 'TaxLibrary.csproj'.
MSBuild auto-detection: using msbuild version '16.11.1.47101' from 'C:\Program Files (x86)\Microsoft Visual Studio\2019\
Community\MSBuild\Current\bin'.
Packing files from 'C:\Users\StudentPC\source\repos\TaxLibrary\TaxLibrary\bin\Debug'.
WARNING: NU5115: Description was not specified. Using 'Description'.
WARNING: NU5115: Author was not specified. Using 'StudentPC'.
Successfully created package 'C:\Users\StudentPC\source\repos\TaxLibrary\TaxLibrary\TaxLibrary.1.0.0.nupkg'.
C:\Users\StudentPC\source\repos\TaxLibrary\TaxLibrary> _
```

> Switching back to the Windows Explorer you can see **TaxLibrary.1.0.0.nupkg** file got created.

> NuGet builds a minimal package based on the information it is able to pull from the project. For example, note that the name is TaxLibrary.1.0.0.nupkg. That version number was pulled from the assembly.

12. Return to **Visual Studio**. From Solution Explorer, open **Properties\AssemblyInfo.cs**.

> The AssemblyVersion attribute specifies the version number to build into the assembly. Each NuGet release requires a unique version number, so if we continue to use this method for creating packages, we will need to remember to increment this before building.

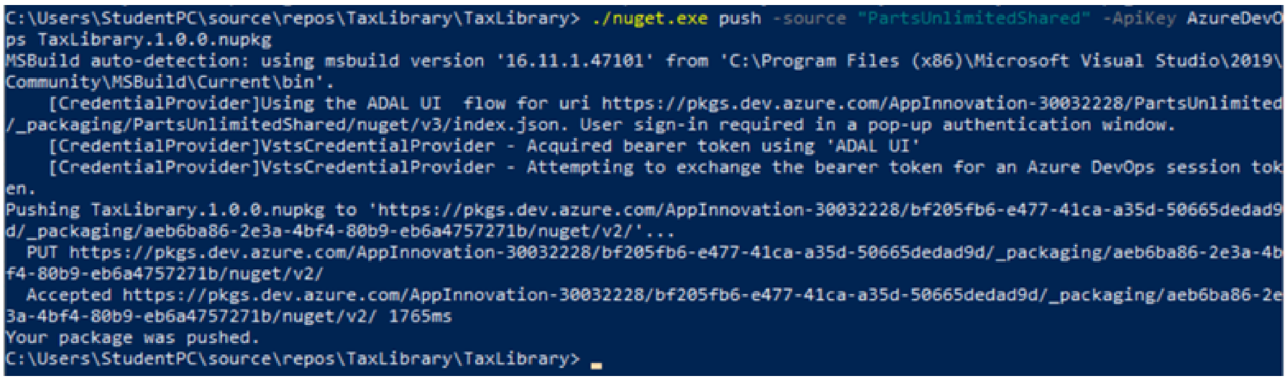


```

    // Version information for an assembly consists of the following four values:
    //
    //      Major Version
    //      Minor Version
    //      Build Number
    //      Revision
    //
    // You can specify all the values or you can default the Build and Revision Numbers
    // by using the '*' as shown below:
    // [assembly: AssemblyVersion("1.0.*")]
    [assembly: AssemblyVersion("1.0.0.0")]
    [assembly: AssemblyFileVersion("1.0.0.0")]

```

13. Return to the PowerShell window and execute the following command (it should be in one line). When prompted, type in your credentials and proceed.

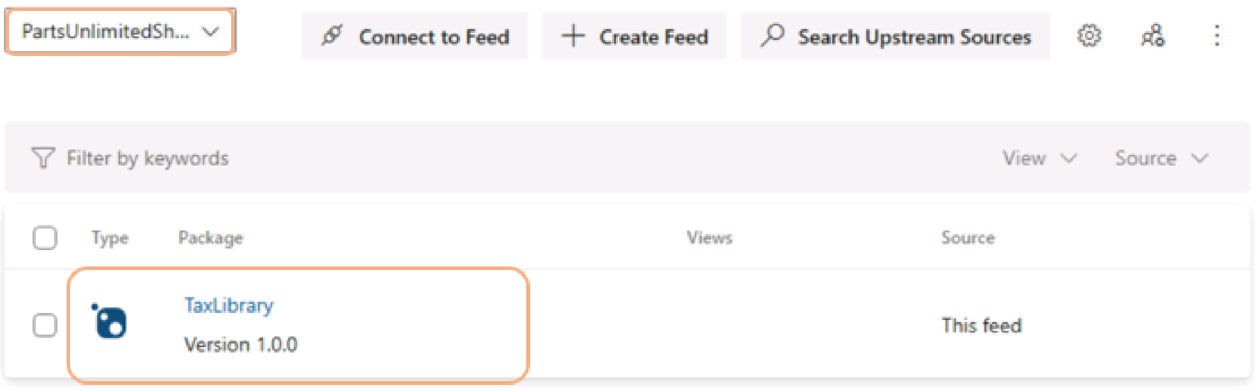
> You need to provide an “API Key”, which can be any non-empty string. We’re using “AzureDevOps” here.

```
./nuget.exe push -source "PartsUnlimitedShared" -ApiKey AzureDevOps
TaxLibrary.1.0.0.nupkg
```

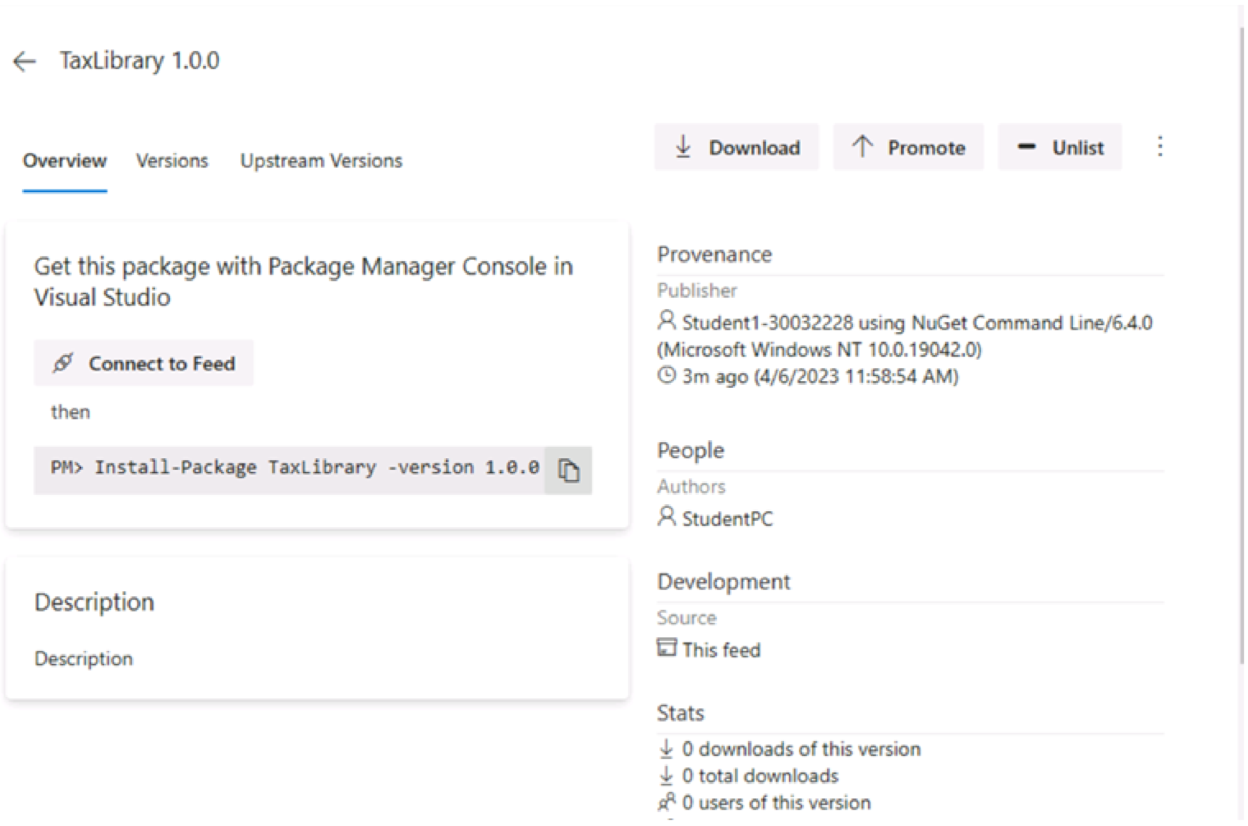


```
C:\Users\StudentPC\source\repos\TaxLibrary\TaxLibrary> ./nuget.exe push -source "PartsUnlimitedShared" -ApiKey AzureDevO
ps TaxLibrary.1.0.0.nupkg
MSBuild auto-detection: using msbuild version '16.11.1.47101' from 'C:\Program Files (x86)\Microsoft Visual Studio\2019\
Community\MSBuild\Current\bin'.
    [CredentialProvider]Using the ADAL UI  flow for uri https://pkgs.dev.azure.com/AppInnovation-30032228/PartsUnlimited
/_packaging/PartsUnlimitedShared/nuget/v3/index.json. User sign-in required in a pop-up authentication window.
    [CredentialProvider]VstsCredentialProvider - Acquired bearer token using 'ADAL UI'
    [CredentialProvider]VstsCredentialProvider - Attempting to exchange the bearer token for an Azure DevOps session tok
en.
Pushing TaxLibrary.1.0.0.nupkg to 'https://pkgs.dev.azure.com/AppInnovation-30032228/bf205fb6-e477-41ca-a35d-50665dedad9
d/_packaging/aeb6ba86-2e3a-4bf4-80b9-eb6a4757271b/nuget/v2/'...
  PUT https://pkgs.dev.azure.com/AppInnovation-30032228/bf205fb6-e477-41ca-a35d-50665dedad9d/_packaging/aeb6ba86-2e3a-4b
f4-80b9-eb6a4757271b/nuget/v2/
  Accepted https://pkgs.dev.azure.com/AppInnovation-30032228/bf205fb6-e477-41ca-a35d-50665dedad9d/_packaging/aeb6ba86-2e
3a-4bf4-80b9-eb6a4757271b/nuget/v2/ 1765ms
Your package was pushed.
C:\Users\StudentPC\source\repos\TaxLibrary\TaxLibrary>
```

14. Return to the browser window open with Azure DevOps Services. Click on **Artifacts** for the PartsUnlimited project. Select the dropdown to select **PartsUnlimitedShared** feed. You should now see the organization's NuGet package is published in the feed. Click **TaxLibrary** to view the details.



Type	Package	Views	Source
	TaxLibrary Version 1.0.0		This feed



Get this package with Package Manager Console in Visual Studio

then

```
PM> Install-Package TaxLibrary -version 1.0.0
```

Description

Description

Provenance
Publisher
Student1-30032228 using NuGet Command Line/6.4.0 (Microsoft Windows NT 10.0.19042.0)
3m ago (4/6/2023 11:58:54 AM)

People
Authors
StudentPC

Development
Source
This feed

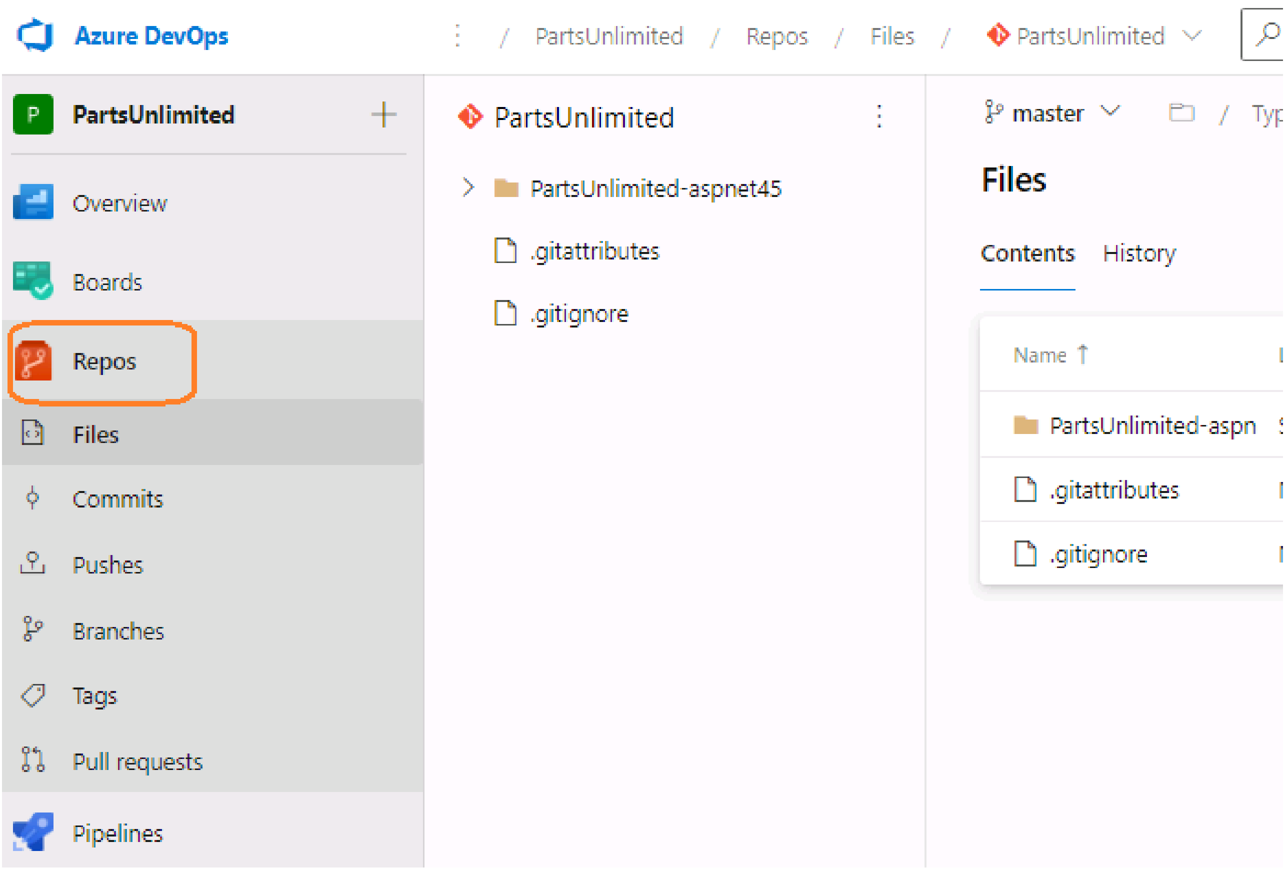
Stats
0 downloads of this version
0 total downloads
0 users of this version

Module 5: **Azure Artifacts**, **Lab 1: Azure Artifacts**, Exercise 1: Working with Azure Artifacts

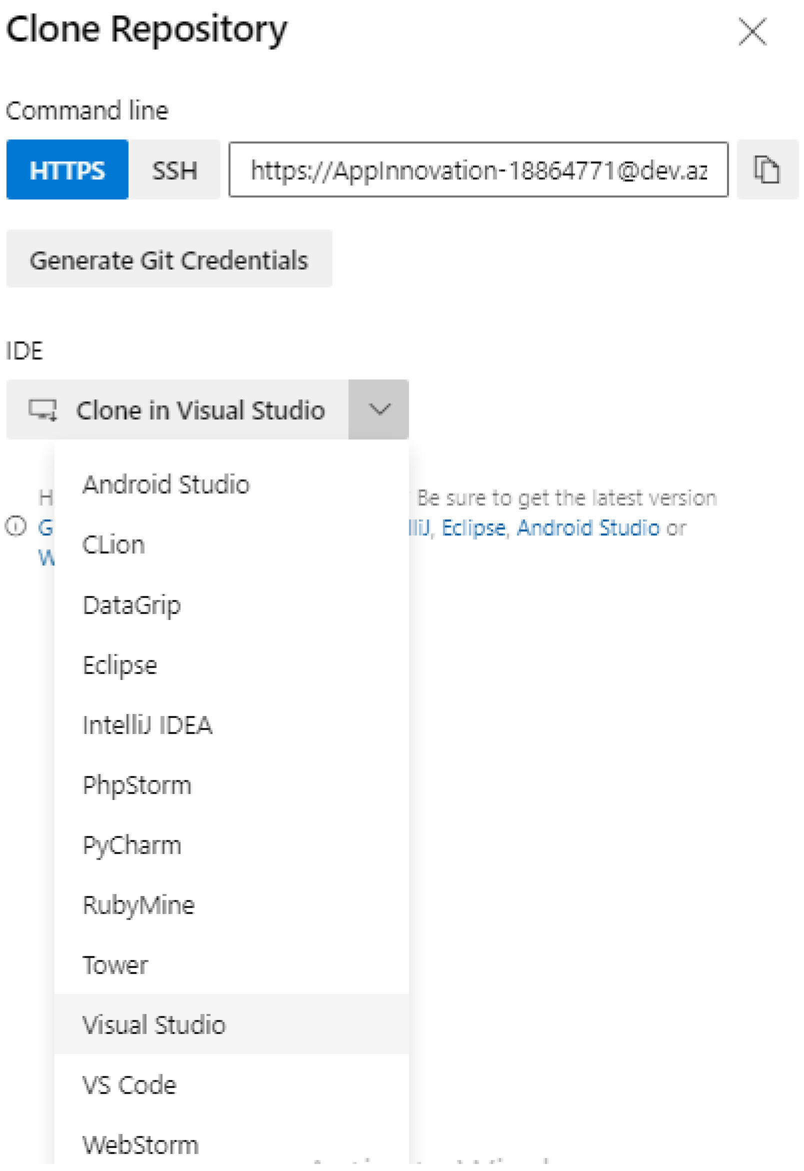
---

Task 3: Cloning PartsUnlimited Repository

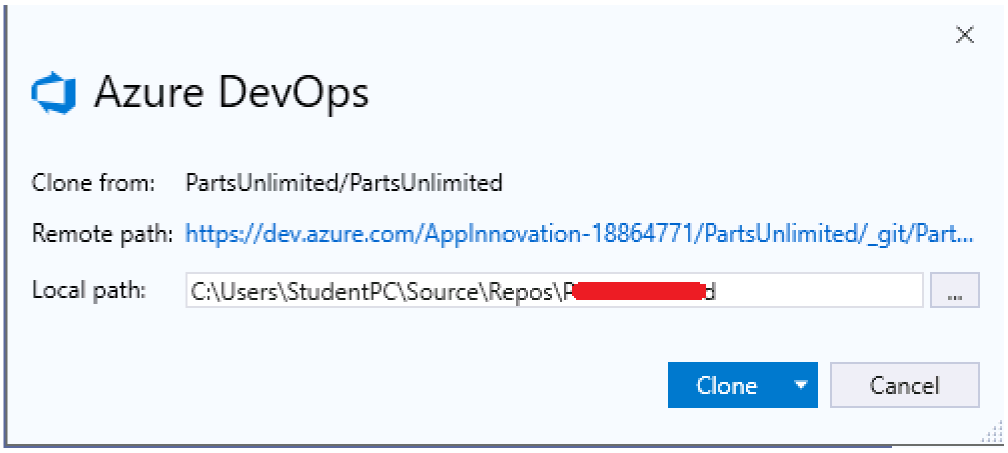
1. Navigate to **PartsUnlimited** project in the browser. You can do this by navigating to https://dev.azure.com/AppInnovation-[YourName]/PartsUnlimited.
2. Navigate to the **Repos hub**. And select **PartsUnlimited** repo from the top-center dropdown.



3. Click **Clone** and select **Clone in Visual Studio**. Follow the workflow that opens Visual Studio.



4. **Clone** the repository at the new local folder.

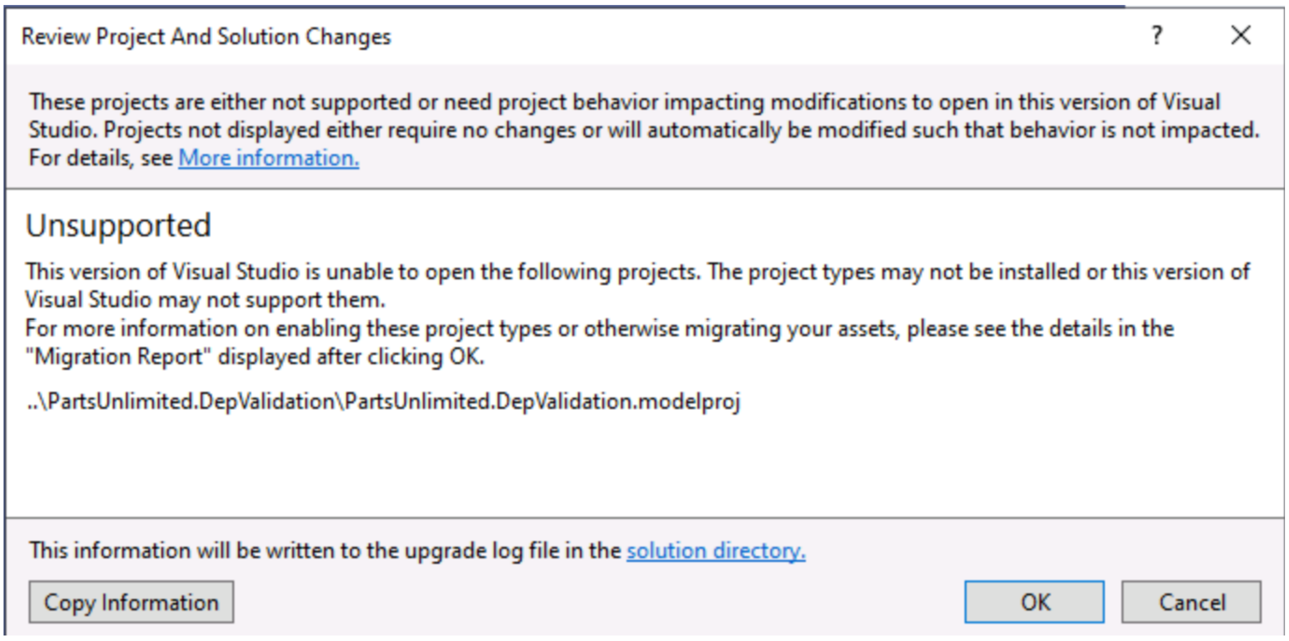


Clone from: PartsUnlimited/PartsUnlimited

Remote path: https://dev.azure.com/AppInnovation-18864771/PartsUnlimited/_git/Part...

Local path: C:\Users\StudentPC\Source\Repos\P[redacted]d

5. In the **Solution Explorer** double-click on **PartsUnlimited.sln**. If displayed with the following message, click on **OK**.



These projects are either not supported or need project behavior impacting modifications to open in this version of Visual Studio. Projects not displayed either require no changes or will automatically be modified such that behavior is not impacted. For details, see More information.

Unsupported

This version of Visual Studio is unable to open the following projects. The project types may not be installed or this version of Visual Studio may not support them.
For more information on enabling these project types or otherwise migrating your assets, please see the details in the "Migration Report" displayed after clicking OK.

..\PartsUnlimited.DepValidation\PartsUnlimited.DepValidation.modelproj

This information will be written to the upgrade log file in the solution directory.

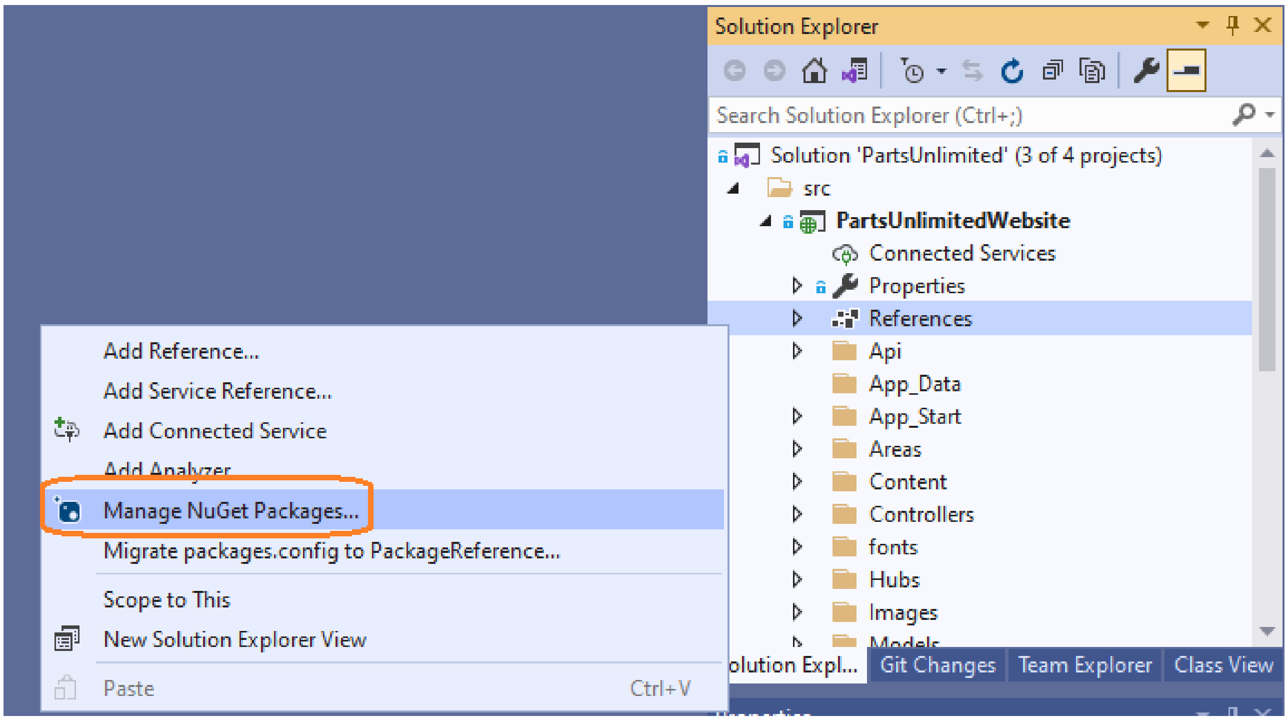
6. Close the **Migration Report** that might have got opened in the browser.

Module 5: **Azure Artifacts**, **Lab 1: Azure Artifacts**, Exercise 1: Working with Azure Artifacts

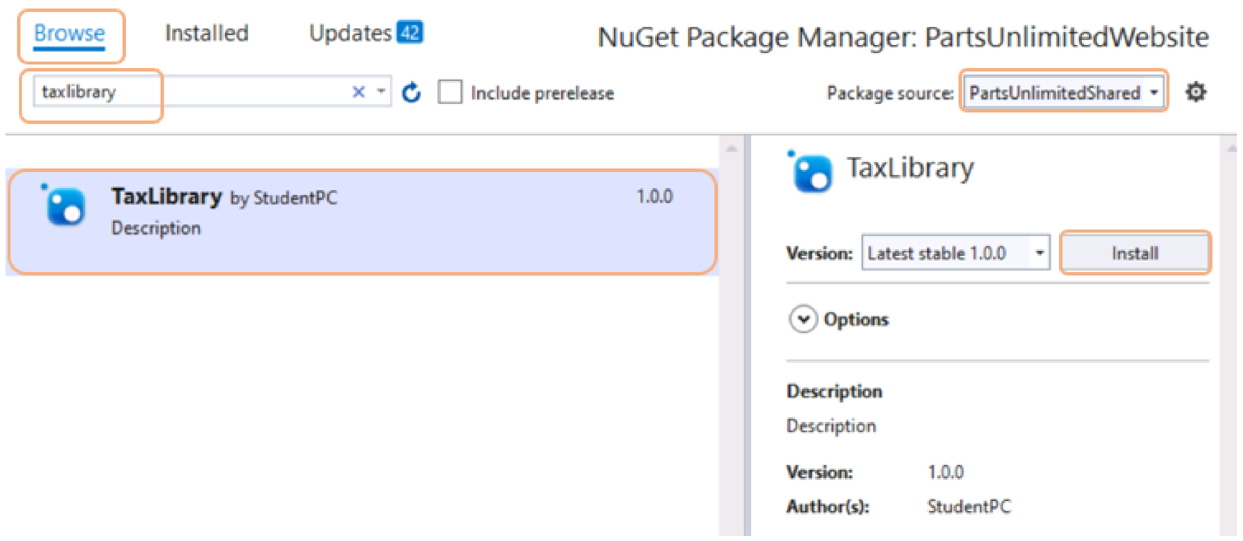
---

Task 4: Importing a NuGet package

1. Inside the **Visual Studio** that has the **PartsUnlimited** solution open, in the **Solution Explorer**, right-click the **References** node under the **PartsUnlimitedWebsite** project and select **Manage NuGet Packages**.



2. Click the **Browse** tab and change the **Package source** to **PartsUnlimitedShared**. Search for the **TaxLibrary** package. Click **Install** to add it to the project.



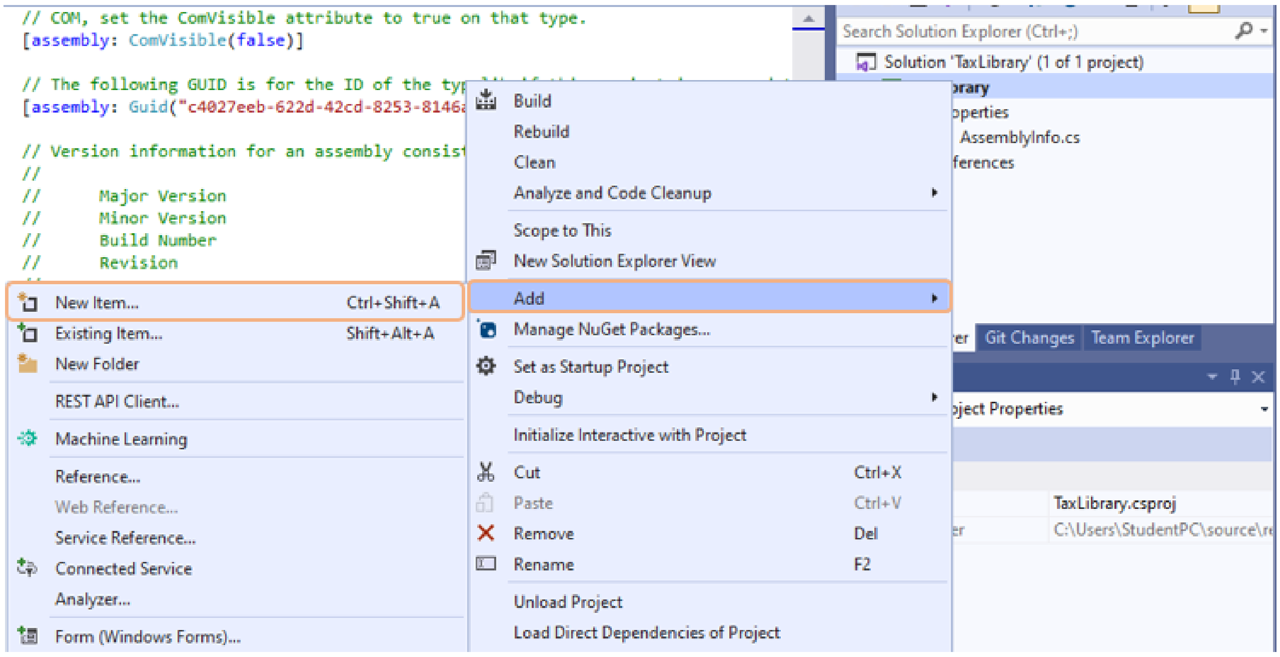
3. If asked, confirm the addition by clicking **OK**.

4. Press **Ctrl+Shift+B** to build the project. It should succeed. The NuGet package doesn't add any value yet, but at least we know it's there.

Task 5: Updating a NuGet package

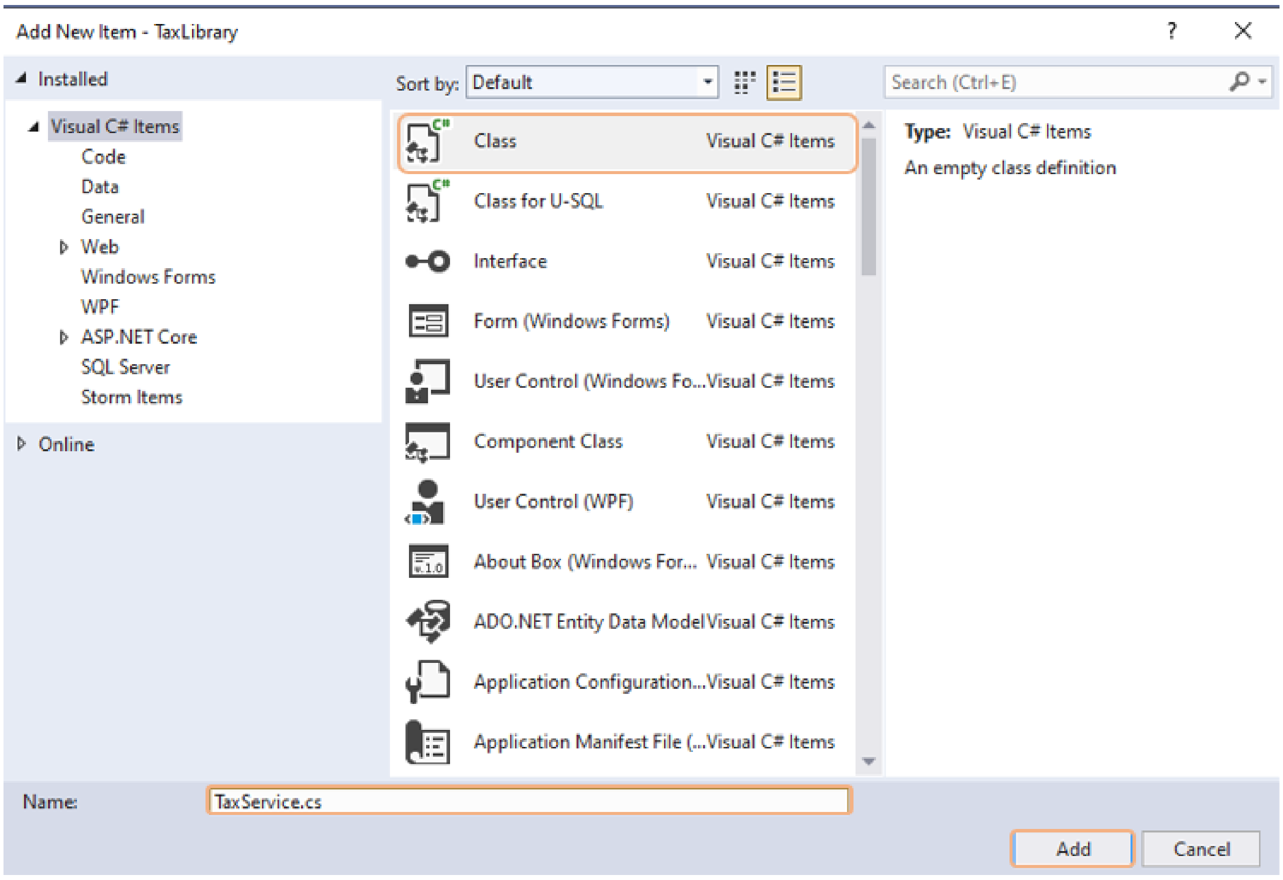
1. Switch to the instance of **Visual Studio** that has the **TaxLibrary** project open (the NuGet source project).

2. In the **Solution Explorer**, right-click the TaxLibrary project node and select **Add | New Item**.



```
// COM, set the ComVisible attribute to true on that type.
[assembly: ComVisible(false)]

// The following GUID is for the ID of the typ
[assembly: Guid("c4027eeb-622d-42cd-8253-8146a

// Version information for an assembly consist
//
//      Major Version
//      Minor Version
//      Build Number
//      Revision
```

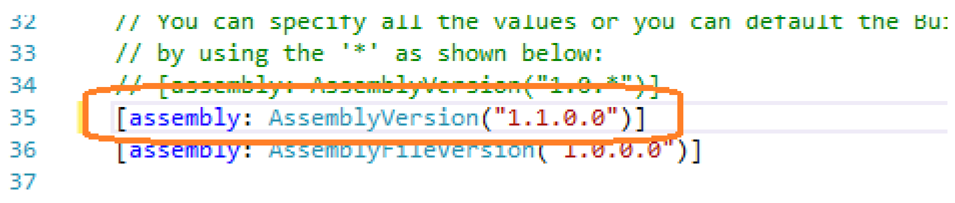
3. Select the Class template and enter the Name **TaxService.cs**. Click **Add** to add the class. We can pretend that tax calculation will be consolidated into this shared class and managed centrally so that other teams can simply work with the NuGet package.


Add New Item - TaxLibrary
Installed
Visual C# Items
Code
Data
General
Web
Windows Forms
WPF
ASP.NET Core
SQL Server
Storm Items
Online
Sort by: Default
Search (Ctrl+E)
Class Visual C# Items
Class for U-SQL Visual C# Items
Interface Visual C# Items
Form (Windows Forms) Visual C# Items
User Control (Windows Fo... Visual C# Items
Component Class Visual C# Items
User Control (WPF) Visual C# Items
About Box (Windows For... Visual C# Items
ADO.NET Entity Data Model Visual C# Items
Application Configuration... Visual C# Items
Application Manifest File (... Visual C# Items
Type: Visual C# Items
An empty class definition
Name: TaxService.cs
Add
Cancel

4. Replace the code in the new file with the code below and save the file with **Ctrl + S**. For now it will just hardcode a 10% rate.

```
namespace TaxLibrary
{
    public class TaxService
    {
        static public decimal CalculateTax(decimal taxable, string
postalCode)
        {
            return taxable * (decimal).1;
        }
    }
}
```

5. Since we're updating the assembly (and package), return to **AssemblyInfo.cs** to update the **AssemblyVersion to 1.1.0.0** and **Ctrl + s** to save the changes.



```
    // You can specify all the values or you can default the Bu:
    // by using the '*' as shown below:
    // [assembly: AssemblyVersion("1.0.*")]
    [assembly: AssemblyVersion("1.1.0.0")]
    [assembly: AssemblyFileVersion("1.0.0.0")]
```

6. Press **Ctrl+Shift+B** to build the project.

7. Return to the PowerShell window and execute the following line to repackage the NuGet package. Note that the new package will have the updated version number.

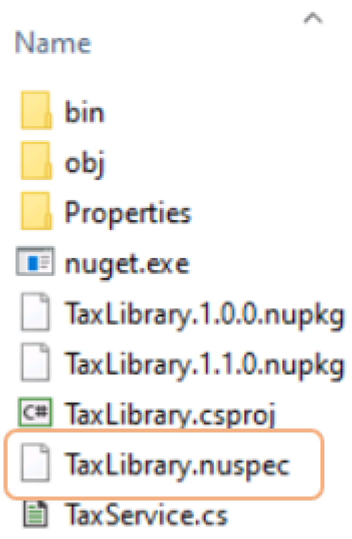
   ```
   ./nuget.exe pack TaxLibrary.csproj
   ```

8. If your AssemblyVersion is not updating as expected, return to your PowerShell window and run the following command

   ```
   ./nuget.exe spec TaxLibrary.csproj
   ```

   This will create a Manifest file for your package that we can then manually edit.

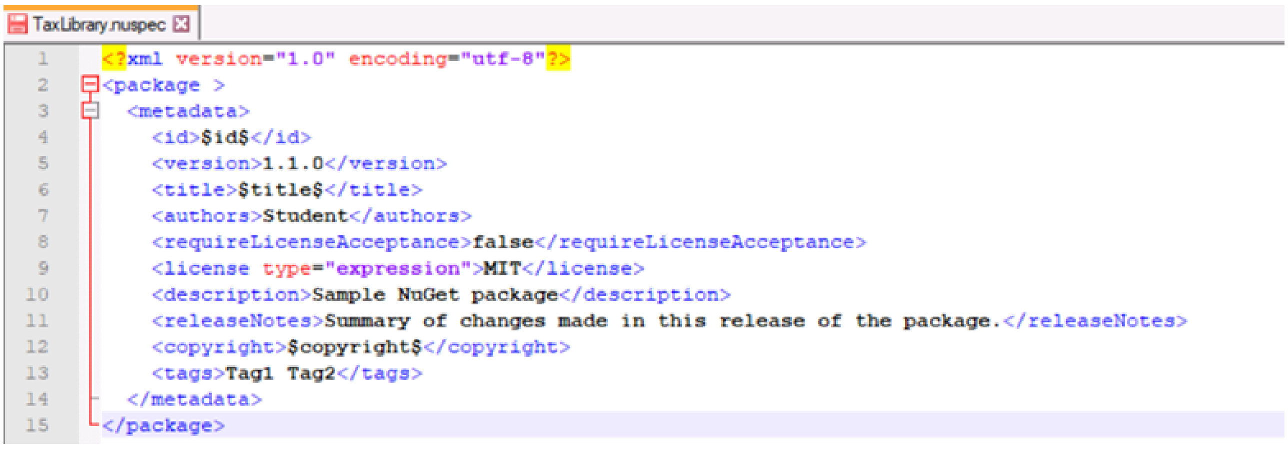
9. Open the File Explorer window containing your .csproj file and select the newly created **TaxLibrary.nuspec** manifest file to open in a **Notepad/text editor**.

   
   Name
   bin
   obj
   Properties
   nuget.exe
   TaxLibrary.1.0.0.nupkg
   TaxLibrary.1.1.0.nupkg
   TaxLibrary.csproj
   TaxLibrary.nuspec
   TaxService.cs

10. In the Manifest file, update the following:

    - $version$ to **1.1.0**
    - $author$ to **Student**
    - Remove the **projectURL** and **icon** lines completely
    - $description$ to **Sample NuGet package**

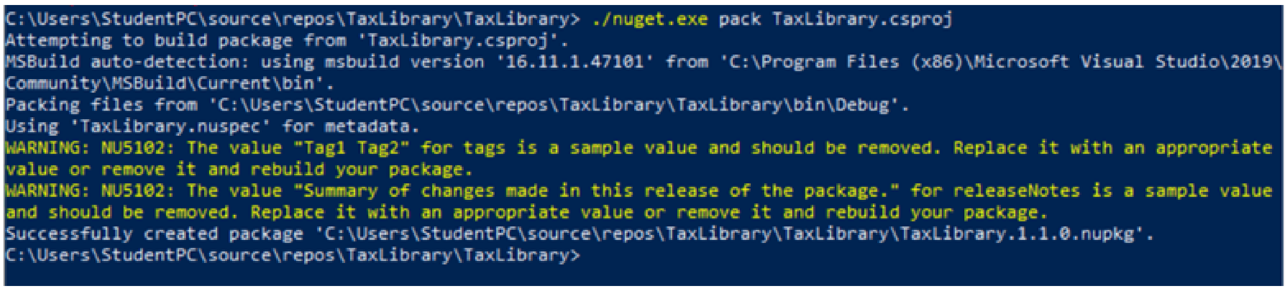
11. When finished editing, your Manifest file should look similar to the following:


TaxLibrary.nuspec

```
<?xml version="1.0" encoding="utf-8"?>
<package >
  <metadata>
    <id>$id$</id>
    <version>1.1.0</version>
    <title>$title$</title>
    <authors>Student</authors>
    <requireLicenseAcceptance>false</requireLicenseAcceptance>
    <license type="expression">MIT</license>
    <description>Sample NuGet package</description>
    <releaseNotes>Summary of changes made in this release of the package.</releaseNotes>
    <copyright>$copyright$</copyright>
    <tags>Tag1 Tag2</tags>
  </metadata>
</package>
```

12. **Save** and **close** this file and return to **PowerShell** window. Re-run the following command again:

```
./nuget.exe pack TaxLibrary.csproj
```



```
C:\Users\StudentPC\source\repos\TaxLibrary\TaxLibrary> ./nuget.exe pack TaxLibrary.csproj
Attempting to build package from 'TaxLibrary.csproj'.
MSBuild auto-detection: using msbuild version '16.11.1.47101' from 'C:\Program Files (x86)\Microsoft Visual Studio\2019\
Community\MSBuild\Current\bin'.
Packing files from 'C:\Users\StudentPC\source\repos\TaxLibrary\TaxLibrary\bin\Debug'.
Using 'TaxLibrary.nuspec' for metadata.
WARNING: NU5102: The value "Tag1 Tag2" for tags is a sample value and should be removed. Replace it with an appropriate
value or remove it and rebuild your package.
WARNING: NU5102: The value "Summary of changes made in this release of the package." for releaseNotes is a sample value
and should be removed. Replace it with an appropriate value or remove it and rebuild your package.
Successfully created package 'C:\Users\StudentPC\source\repos\TaxLibrary\TaxLibrary\TaxLibrary.1.1.0.nupkg'.
C:\Users\StudentPC\source\repos\TaxLibrary\TaxLibrary>
```

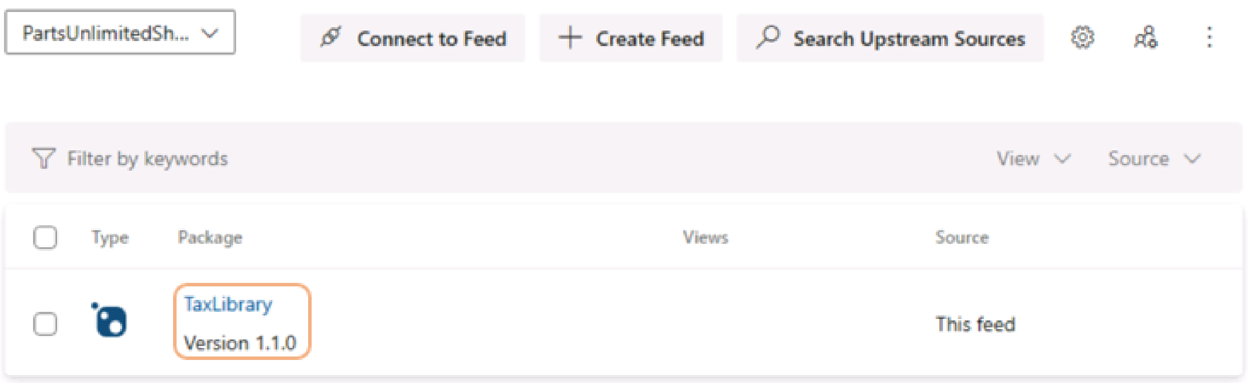
Your Manifest file will override the AssemblyVersion and tell NuGet that this is version 1.1.0 of your NuGet package.

13. Execute the following line to publish the updated package. Note that the version number has changed to reflect the new package. If prompted, enter your username and password again.

```
./nuget.exe push -source "PartsUnlimitedShared" -ApiKey AzureDevOps
TaxLibrary.1.1.0.nupkg
```

You should see **Your package was pushed** message again!

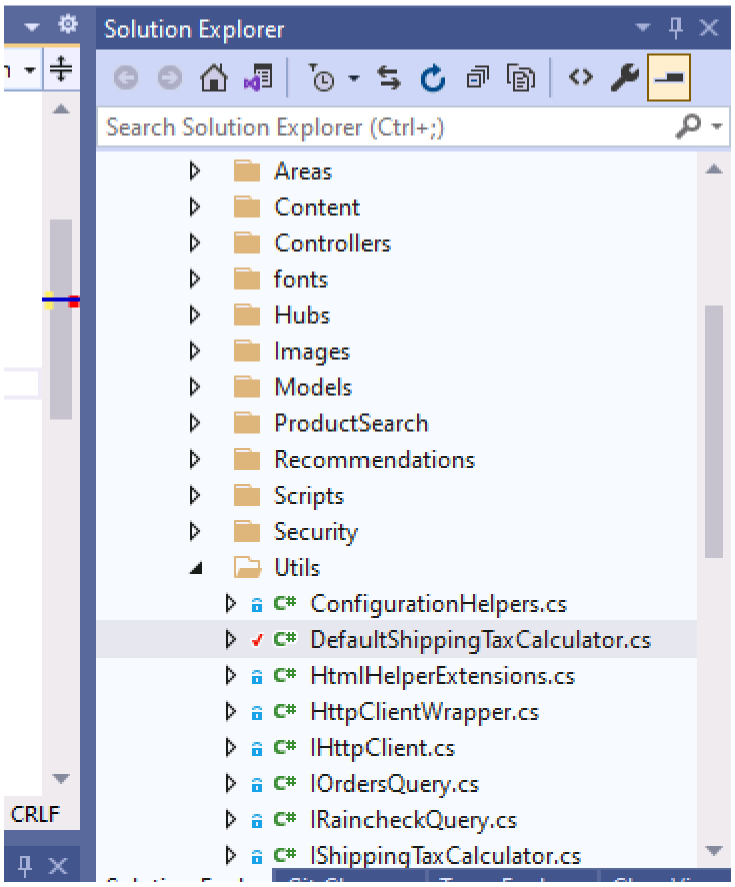
14. Return to the browser window open to Azure DevOps Services and click on **Artifacts**. You can now see **Version 1.1.0** for TaxLibrary package.



Type	Package	Views	Source
	TaxLibrary Version 1.1.0		This feed

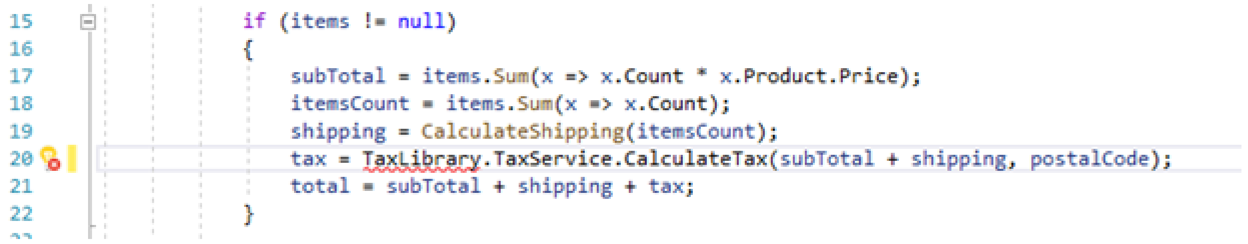
15. Switch back to the instance of **Visual Studio** with the **PartsUnlimited** project (with PartsUnlimitedWebsite).

16. From the **Solution Explorer**, open **PartsUnlimitedWebsite\Utils\DefaultShippingTaxCalculator.cs**.



17. Locate the call to CalculateTax around line 20 and add the qualifier **TaxLibrary.TaxService.** at the beginning.

> The original code called a method internal to this class, so the code we're adding to the beginning of the line is redirecting it to code from our NuGet assembly. However, since this project hasn't updated the NuGet package yet, it's still referencing 1.0.0.0 and doesn't have these new changes available, so the code will not build.

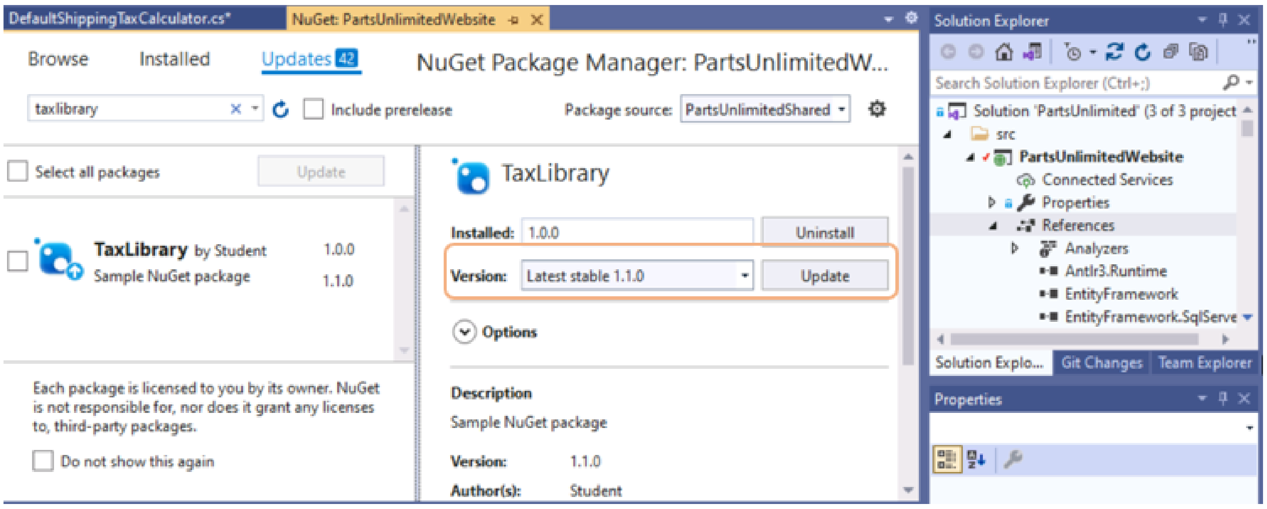


```
if (items != null)
{
    subTotal = items.Sum(x => x.Count * x.Product.Price);
    itemsCount = items.Sum(x => x.Count);
    shipping = CalculateShipping(itemsCount);
    tax = TaxLibrary.TaxService.CalculateTax(subTotal + shipping, postalCode);
    total = subTotal + shipping + tax;
}
```

18. In the **Solution Explorer**, right-click the **References** node and select **Manage NuGet Packages**.

19. NuGet is aware of our update, so click the **Updates** tab to view the details. Click **Update** to bring down the new version.

> [!note] If the **Updates** tab hasn't yet updated, you can still update the package from the **Browse** tab. Note that there may be many NuGet updates available, but you should only need to update TaxLibrary. Note that it may take a little while for the package to become completely available for updating. If you get an error, wait a moment and try again. You can also try clearing the NuGet cache in Visual Studio by going to Tools -> NuGet Package Manager -> Package Manager Settings -> General -> Clear All NuGet Cache(s)



NuGet Package Manager: PartsUnlimitedW...

taxlibrary

Package source: PartsUnlimitedShared

TaxLibrary by Student 1.0.0
Sample NuGet package 1.1.0

Installed: 1.0.0

Version: Latest stable 1.1.0

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

Description
Sample NuGet package

Version: 1.1.0
Author(s): Student

20. If asked, click OK to approve the update.

21. Press **F5** to build and run the site. It should work as expected.