UiPath Custom Activity to Edit a PDF

UIPath.PDF.Activities pack allows users to perform various actions on PDF, like reading pdf data, splitting pdf pages, etc. however, what if we want to edit a PDF? Even if we are filling data into a simple document such as a form, does Acrobat DC or any other pdf editing tool license always need to be purchased? Absolutely not.

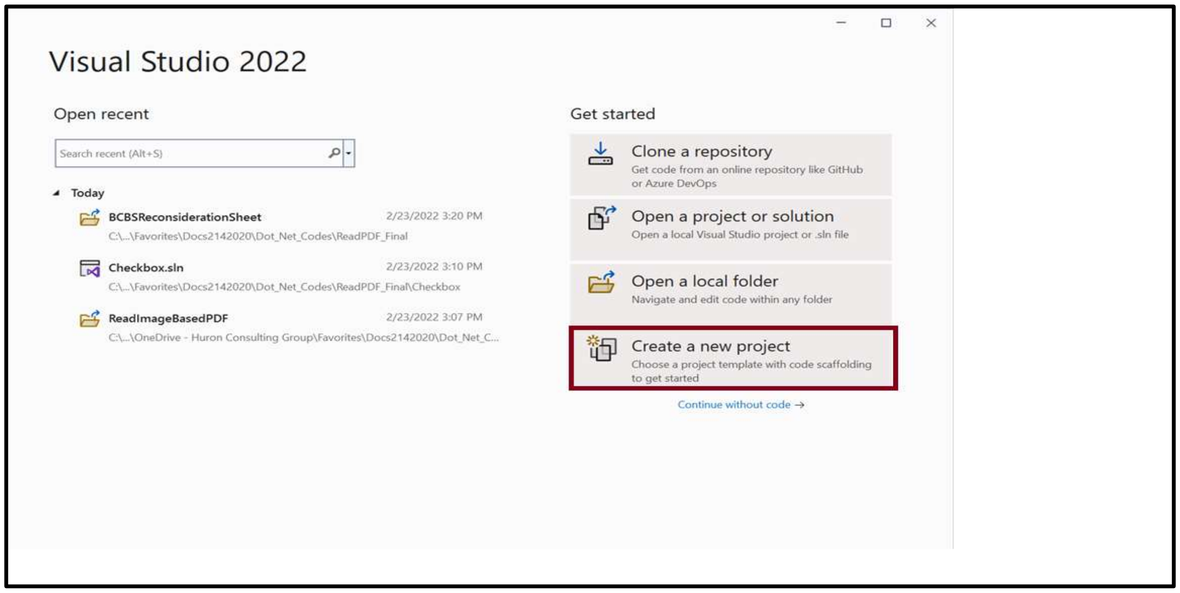
I am writing this post with the intention of showing you how you can create a custom activity to edit a pdf or fill in data in a form that is in the form of a pdf.

Listed below are the major steps involved in creating and adding a custom activity to UiPath –

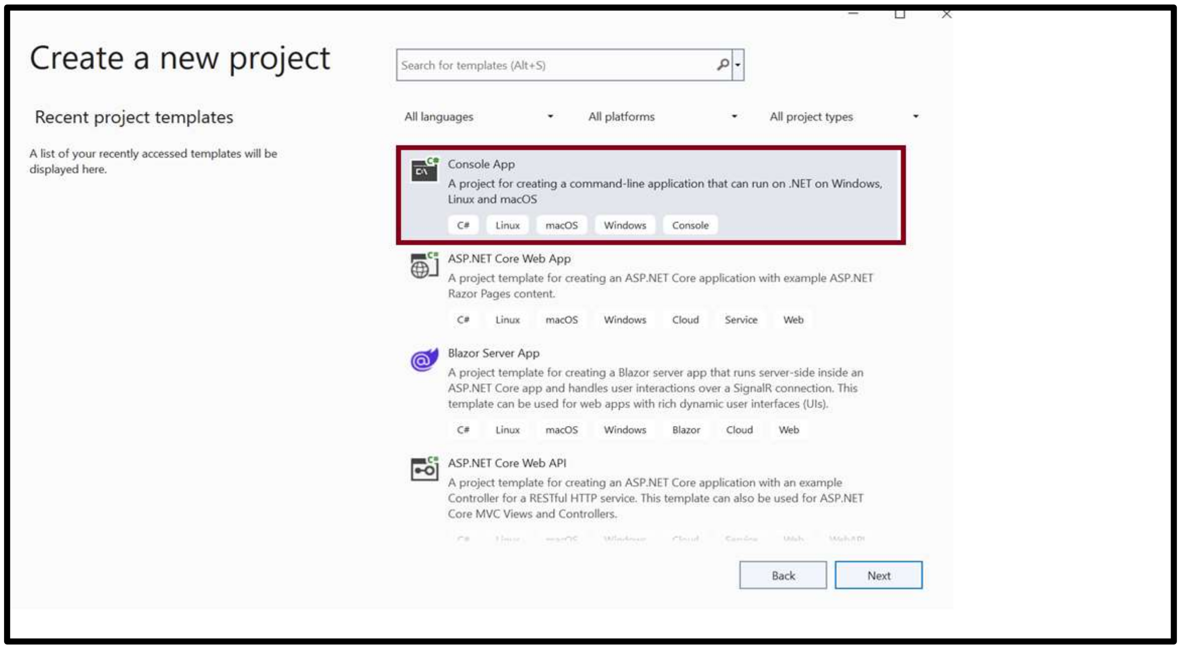
1. Develop a console application in C# to read the field names from pdf.
2. Create the custom activity code to target the fields to edit the pdf file.
3. Convert the DLL generated from step 2 to NuGet package.
4. Configure UiPath Studio to use the NuGet package generated from step 2.

Developing a console application in C# to read the field names from pdf

1. Open Visual Studio and click on Create a new project.


Visual Studio 2022
Open recent
Search recent (Alt+S)
Today
BCBSReconsiderationSheet 2/23/2022 3:20 PM
C:\...\Favorites\Docs2142020\Dot_Net_Codes\ReadPDF_Final
Checkbox.sln 2/23/2022 3:10 PM
C:\...\Favorites\Docs2142020\Dot_Net_Codes\ReadPDF_Final\Checkbox
ReadImageBasedPDF 2/23/2022 3:07 PM
C:\...\OneDrive - Huron Consulting Group\Favorites\Docs2142020\Dot_Net_C...
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 →

2. Select Console App from the templates and click on Next.



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

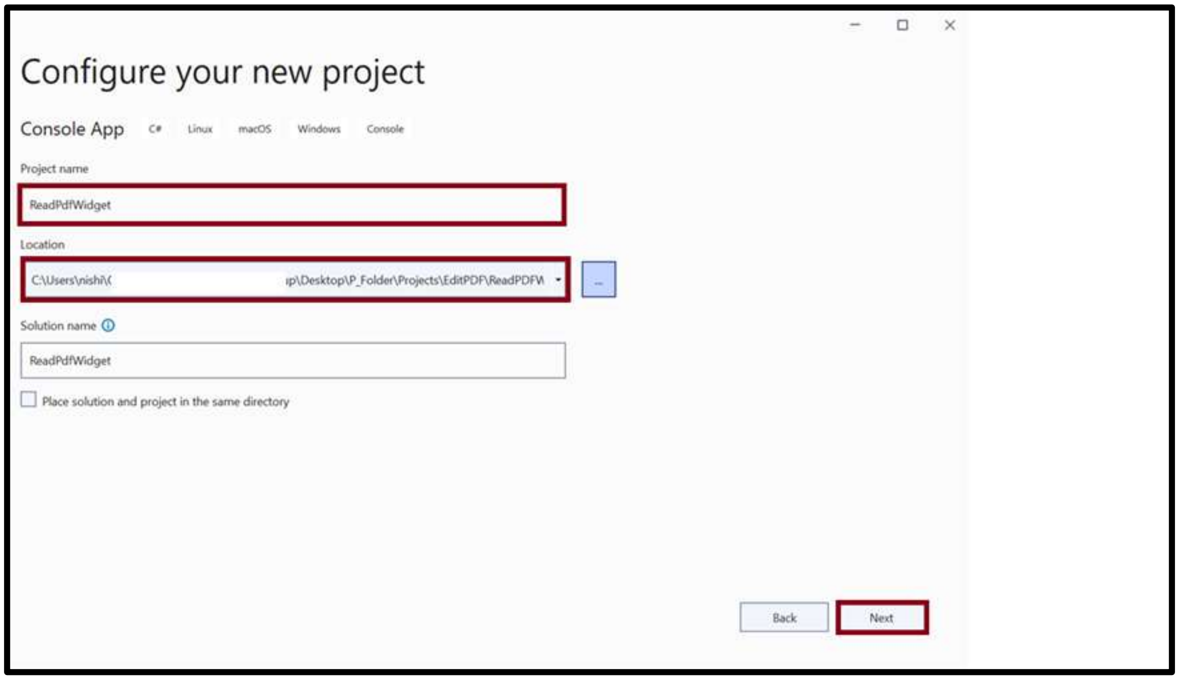
Console App
A project for creating a command-line application that can run on .NET on Windows, Linux and macOS
C# Linux macOS Windows Console

ASP.NET Core Web App
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.
C# Linux macOS Windows Cloud Service Web

Blazor Server App
A project template for creating a Blazor server app that runs server-side inside an ASP.NET Core app and handles user interactions over a SignalR connection. This template can be used for web apps with rich dynamic user interfaces (UIs).
C# Linux macOS Windows Blazor Cloud Web

ASP.NET Core Web API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

3. Type in your Project name and choose the location where you want to save your project and click on Next.


Configure your new project
Console App C# Linux macOS Windows Console
Back
Next

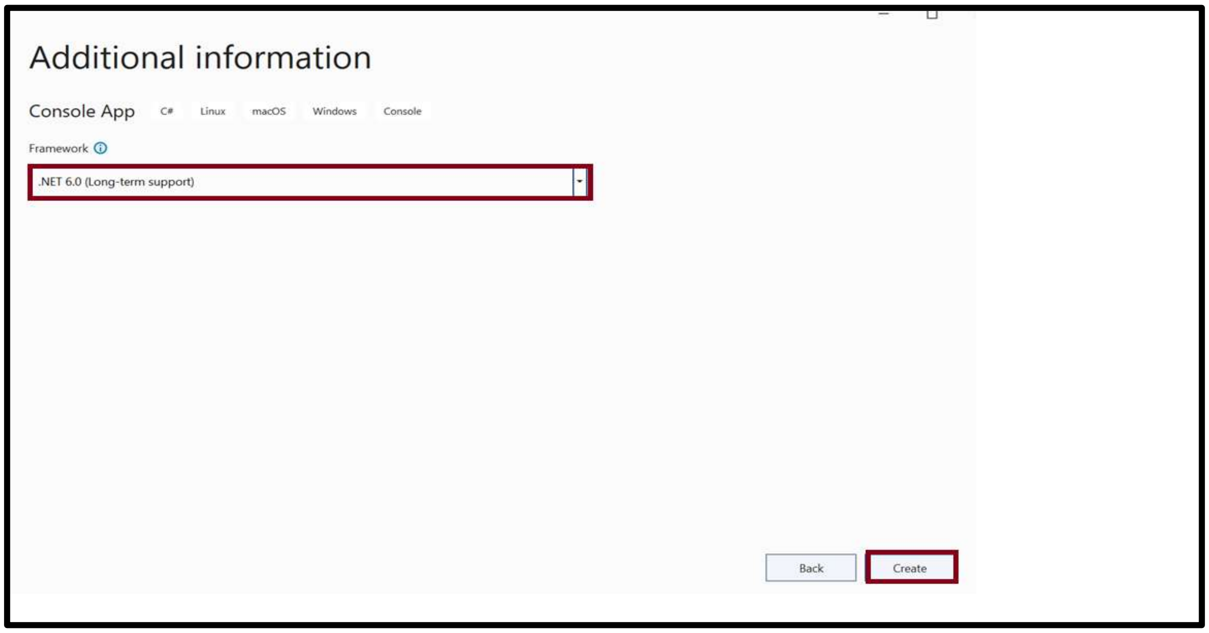
Project name
ReadPdfWidget

Location
C:\Users\nishi\(ıp\Desktop\P_Folder\Projects\EditPDF\ReadPDFW

Solution name
ReadPdfWidget

☐ Place solution and project in the same directory

4. Choose the target framework in which you want to build your project and click on Create. (I have used .Net 6.0.)

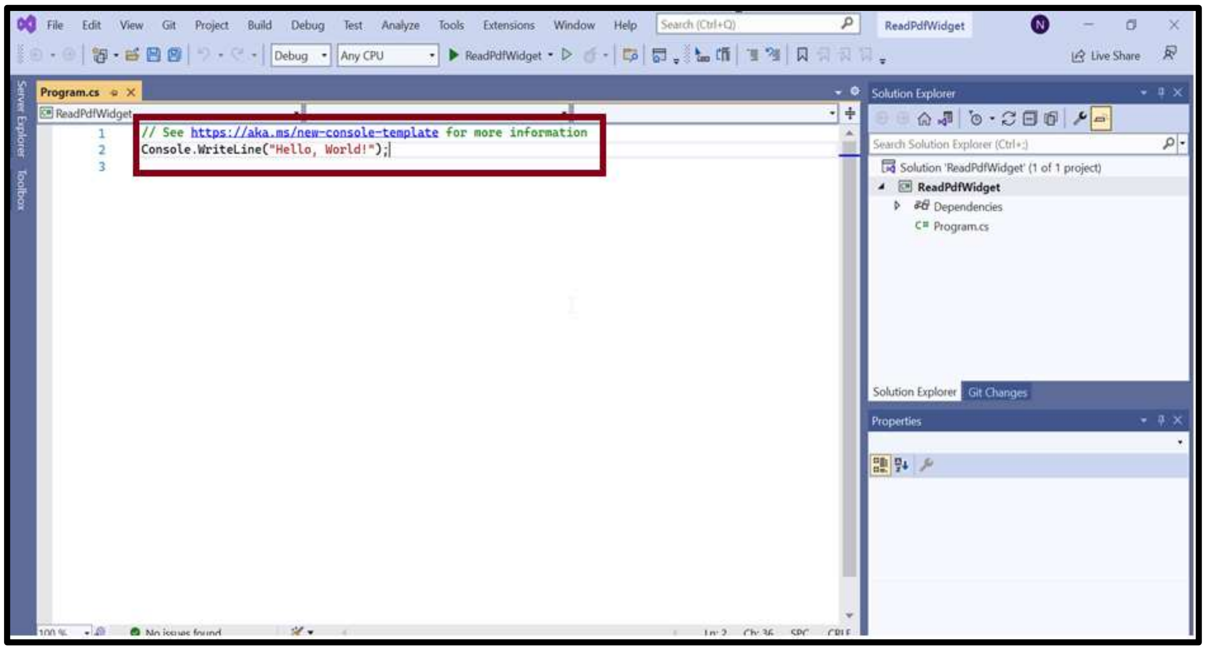


Additional information

Console App C# Linux macOS Windows Console

Framework: .NET 6.0 (Long-term support)

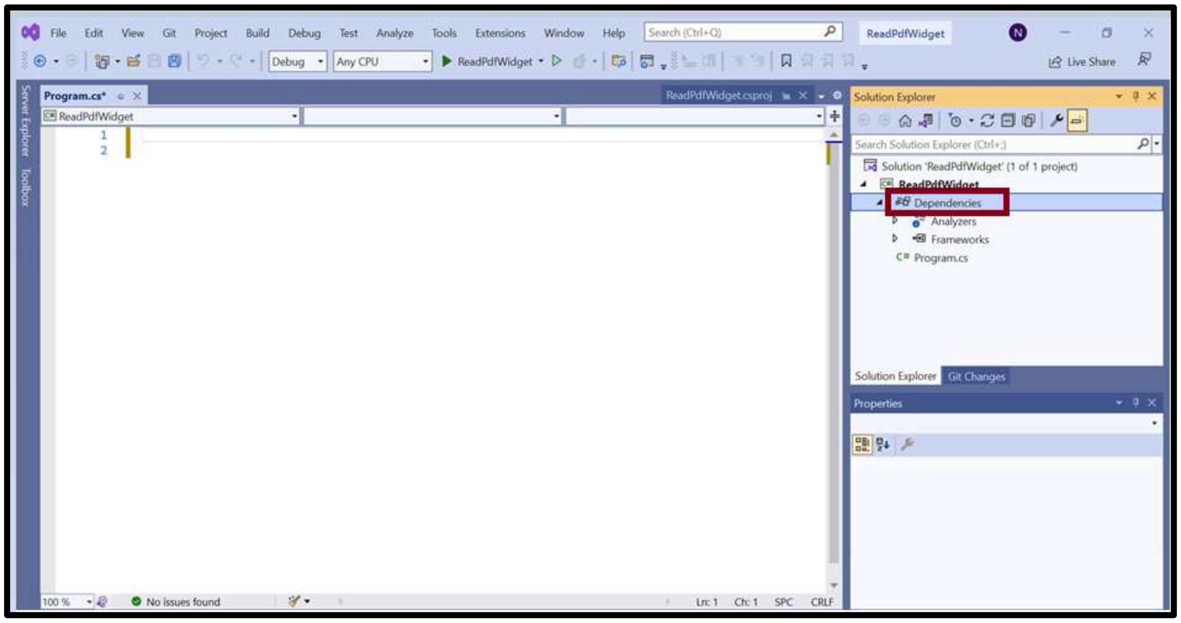
5. Remove all the existing code from the Program.cs file.



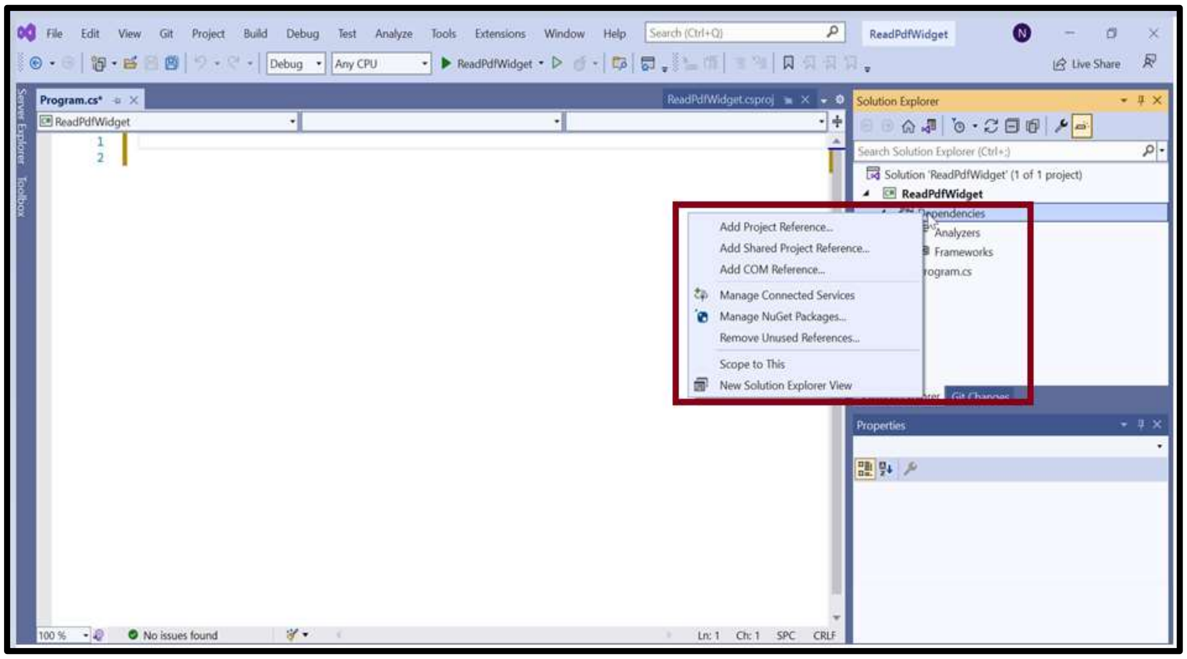
```
// See https://aka.ms/new-console-template for more information
Console.WriteLine("Hello, World!");
```

The next step is to add the required dependencies to get the field names from the pdf.

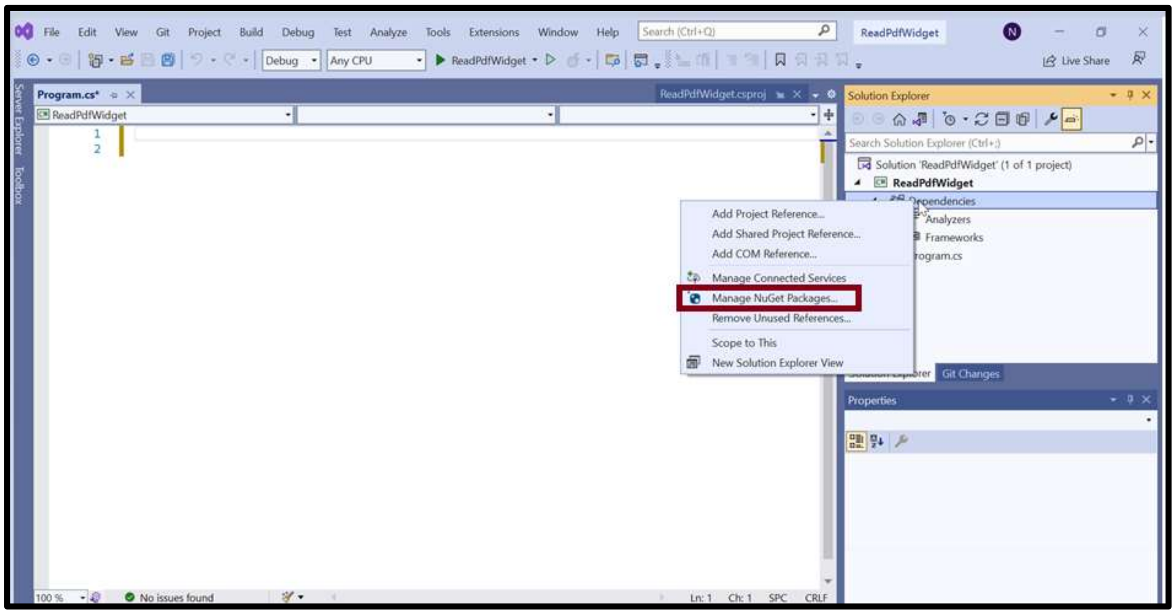
6. Select Dependencies on the right pane.



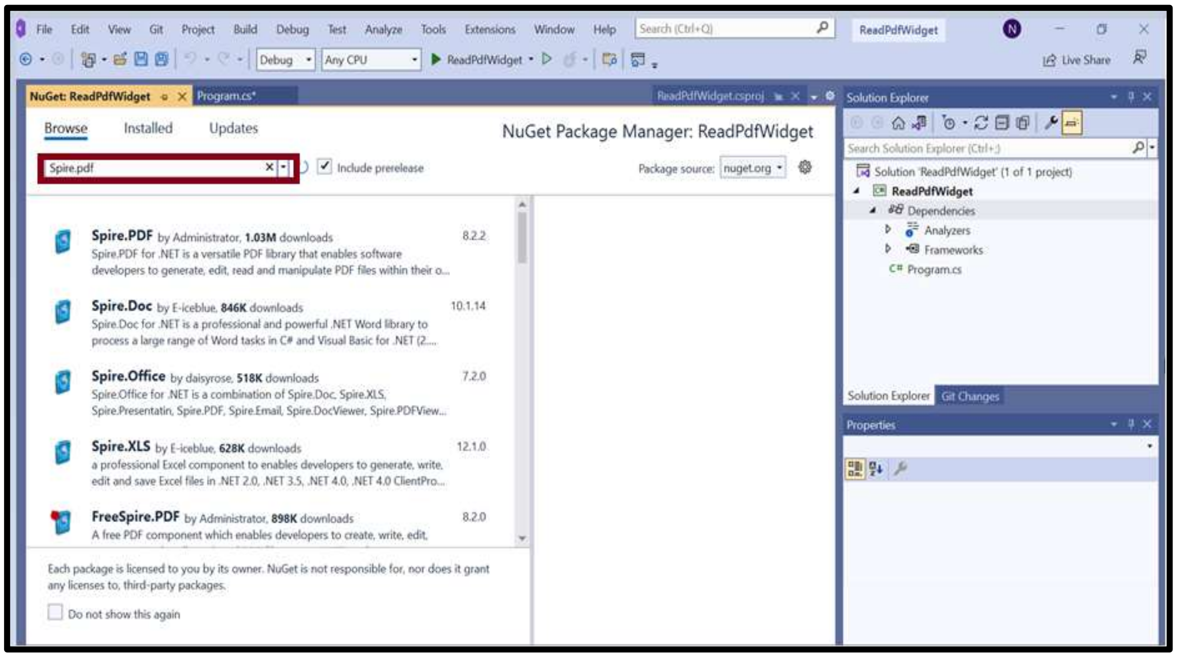
7. Right click on Dependencies.


Add Project Reference...
Add Shared Project Reference...
Add COM Reference...
Manage Connected Services
Manage NuGet Packages...
Remove Unused References...
Scope to This
New Solution Explorer View

8. Select Manage Nuget Packages.



9. Type in Spire.pdf



Spire.pdf

Include prerelease [✓]

Package source: nuget.org

Spire.PDF by Administrator, **1.03M** downloads 8.2.2
Spire.PDF for .NET is a versatile PDF library that enables software developers to generate, edit, read and manipulate PDF files within their o...

Spire.Doc by E-iceblue, **846K** downloads 10.1.14
Spire.Doc for .NET is a professional and powerful .NET Word library to process a large range of Word tasks in C# and Visual Basic for .NET (2....

Spire.Office by daisyrose, **518K** downloads 7.2.0
Spire.Office for .NET is a combination of Spire.Doc, Spire.XLS, Spire.Presentatin, Spire.PDF, Spire.Email, Spire.DocViewer, Spire.PDFView...

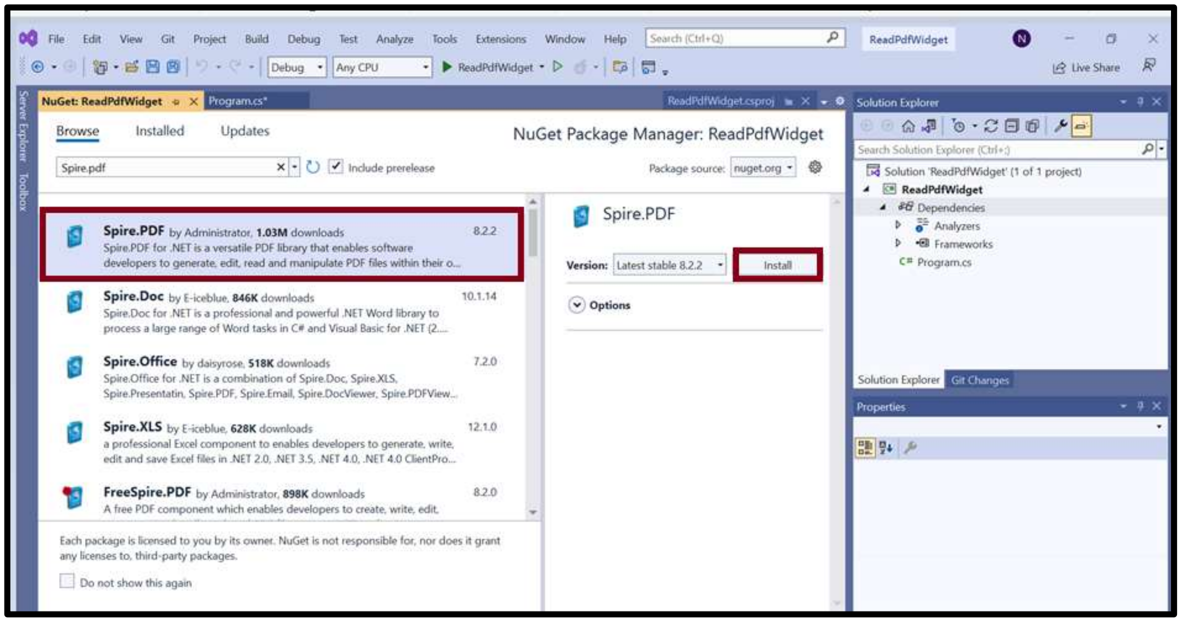
Spire.XLS by E-iceblue, **628K** downloads 12.1.0
a professional Excel component to enables developers to generate, write, edit and save Excel files in .NET 2.0, .NET 3.5, .NET 4.0, .NET 4.0 ClientPro...

FreeSpire.PDF by Administrator, **898K** downloads 8.2.0
A free PDF component which enables developers to create, write, edit,

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

Do not show this again

10. Select Spire.PDF and click on Install



Spire.PDF by Administrator, **1.03M** downloads 8.2.2
Spire.PDF for .NET is a versatile PDF library that enables software developers to generate, edit, read and manipulate PDF files within their o...

Spire.Doc by E-iceblue, **846K** downloads 10.1.14
Spire.Doc for .NET is a professional and powerful .NET Word library to process a large range of Word tasks in C# and Visual Basic for .NET (2....

Spire.Office by daisyrose, **518K** downloads 7.2.0
Spire.Office for .NET is a combination of Spire.Doc, Spire.XLS, Spire.Presentatin, Spire.PDF, Spire.Email, Spire.DocViewer, Spire.PDFView...

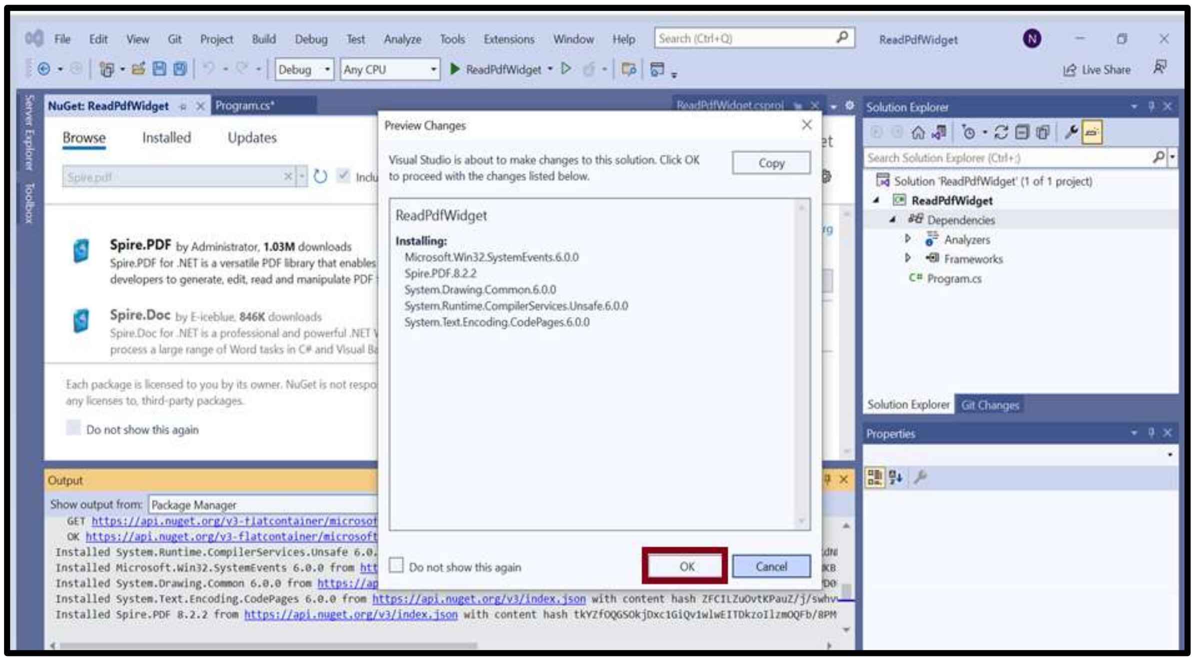
Spire.XLS by E-iceblue, **628K** downloads 12.1.0
a professional Excel component to enables developers to generate, write, edit and save Excel files in .NET 2.0, .NET 3.5, .NET 4.0, .NET 4.0 ClientPro...

FreeSpire.PDF by Administrator, **898K** downloads 8.2.0
A free PDF component which enables developers to create, write, edit,

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

Do not show this again

11. Click on OK.



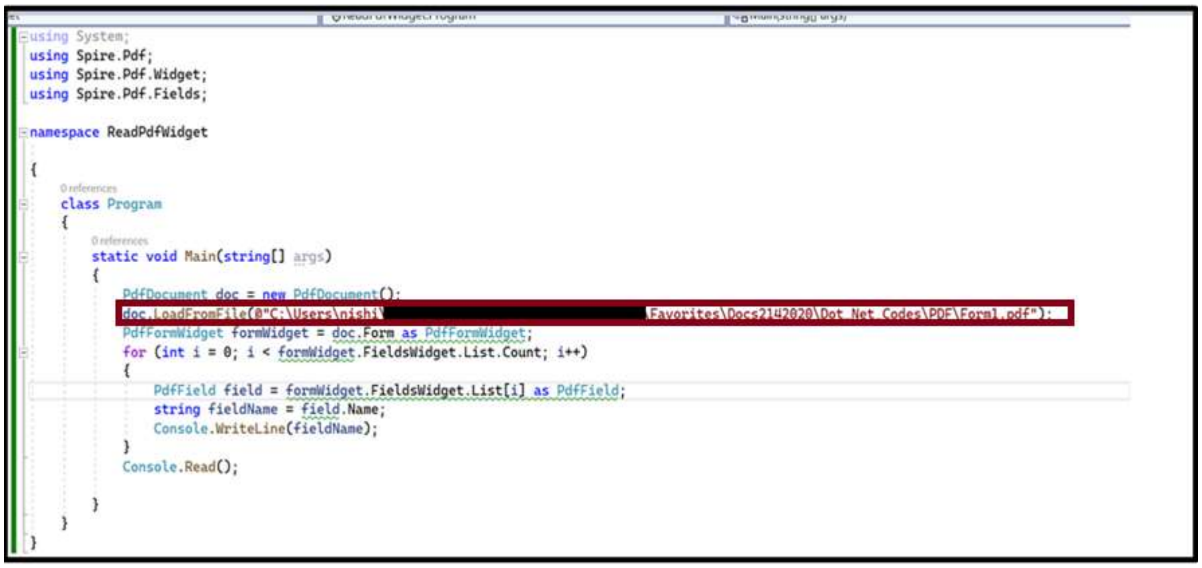
Visual Studio is about to make changes to this solution. Click OK to proceed with the changes listed below.

ReadPdfWidget
Installing:
Microsoft.Win32.SystemEvents.6.0.0
Spire.PDF.8.2.2
System.Drawing.Common.6.0.0
System.Runtime.CompilerServices.Unsafe.6.0.0
System.Text.Encoding.CodePages.6.0.0

Do not show this again

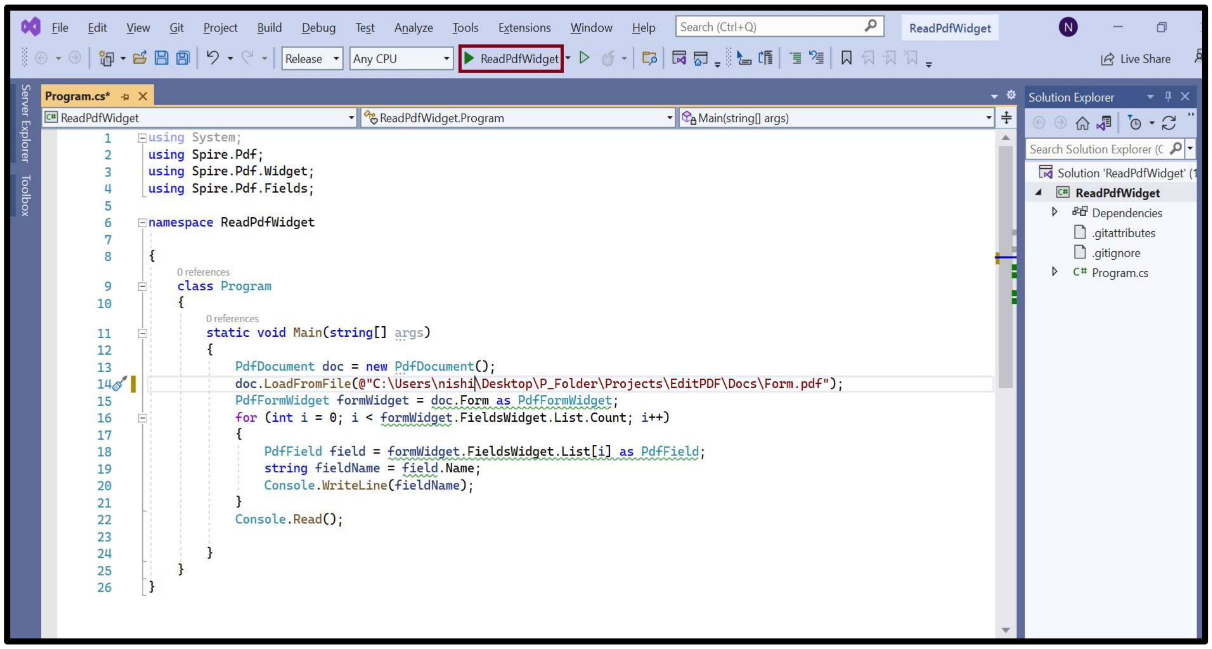
```
  GET https://api.nuget.org/v3-flatcontainer/microsof
  OK https://api.nuget.org/v3-flatcontainer/microsoft
Installed System.Runtime.CompilerServices.Unsafe 6.0.
Installed Microsoft.Win32.SystemEvents 6.0.0 from htt
Installed System.Drawing.Common 6.0.0 from https://ap
Installed System.Text.Encoding.CodePages 6.0.0 from https://api.nuget.org/v3/index.json with content hash ZFCILZuOvtKPauZ/j/swhv
Installed Spire.PDF 8.2.2 from https://api.nuget.org/v3/index.json with content hash tkYZfOQGSOkjDxc1GiQv1wlwEITDkzoIlzmOQFb/8PM
```

12. Copy the code from the repository – https://github.com/Nishi80100/ReadPdfWidget. Specify the pdf file path which you want to edit in the highlighted region.



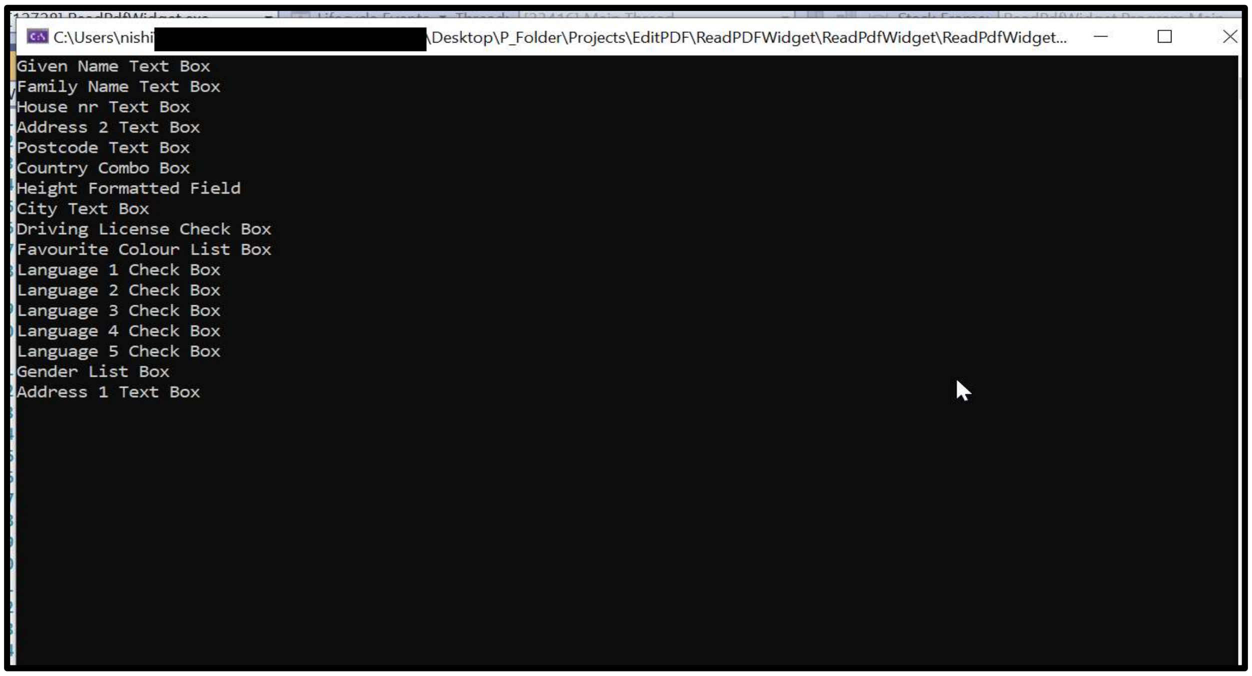
```
using System;
using Spire.Pdf;
using Spire.Pdf.Widget;
using Spire.Pdf.Fields;

namespace ReadPdfWidget

{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            PdfDocument doc = new PdfDocument();
            doc.LoadFromFile(@"C:\Users\nishi\[redacted]Favorites\Docs2142020\Dot Net Codes\PDF\Form1.pdf");
            PdfFormWidget formWidget = doc.Form as PdfFormWidget;
            for (int i = 0; i < formWidget.FieldsWidget.List.Count; i++)
            {
                PdfField field = formWidget.FieldsWidget.List[i] as PdfField;
                string fieldName = field.Name;
                Console.WriteLine(fieldName);
            }
            Console.Read();

        }
    }
}
```

13. Build and run your solution.



```
using System;
using Spire.Pdf;
using Spire.Pdf.Widget;
using Spire.Pdf.Fields;

namespace ReadPdfWidget

{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            PdfDocument doc = new PdfDocument();
            doc.LoadFromFile(@"C:\Users\nishi\Desktop\P_Folder\Projects\EditPDF\Docs\Form.pdf");
            PdfFormWidget formWidget = doc.Form as PdfFormWidget;
            for (int i = 0; i < formWidget.FieldsWidget.List.Count; i++)
            {
                PdfField field = formWidget.FieldsWidget.List[i] as PdfField;
                string fieldName = field.Name;
                Console.WriteLine(fieldName);
            }
            Console.Read();

        }
    }
}
```

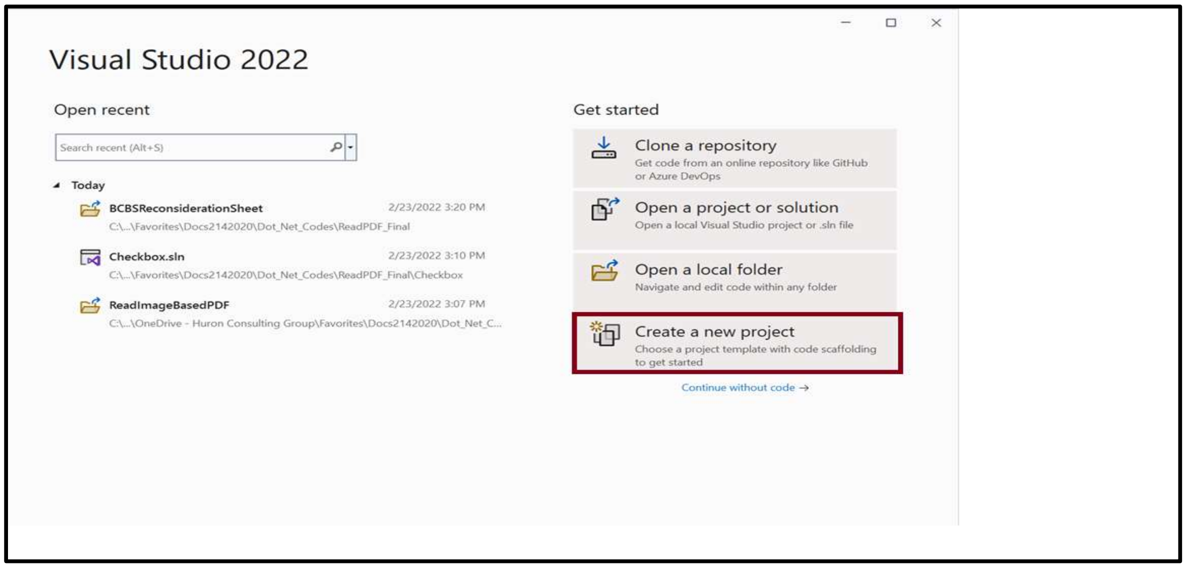
14. The output will list all the text fields associated with the pdf. Make a note of all the field names.



```
C:\Users\nishi[redacted]\Desktop\P_Folder\Projects\EditPDF\ReadPDFWidget\ReadPdfWidget\ReadPdfWidget...
Given Name Text Box
Family Name Text Box
House nr Text Box
Address 2 Text Box
Postcode Text Box
Country Combo Box
Height Formatted Field
City Text Box
Driving License Check Box
Favourite Colour List Box
Language 1 Check Box
Language 2 Check Box
Language 3 Check Box
Language 4 Check Box
Language 5 Check Box
Gender List Box
Address 1 Text Box
```

Creating the custom activity code to edit PDF files by targeting the appropriate fields

1. Open Visual Studio and click on Create a new project.



Visual Studio 2022

Open recent

Search recent (Alt+S)

Today

BCBSReconsiderationSheet 2/23/2022 3:20 PM
C:\...\Favorites\Docs2142020\Dot_Net_Codes\ReadPDF_Final

Checkbox.sln 2/23/2022 3:10 PM
C:\...\Favorites\Docs2142020\Dot_Net_Codes\ReadPDF_Final\Checkbox

ReadImageBasedPDF 2/23/2022 3:07 PM
C:\...\OneDrive - Huron Consulting Group\Favorites\Docs2142020\Dot_Net_C...

Get started

Clone a repository
Get code from an online repository like GitHub or Azure DevOps

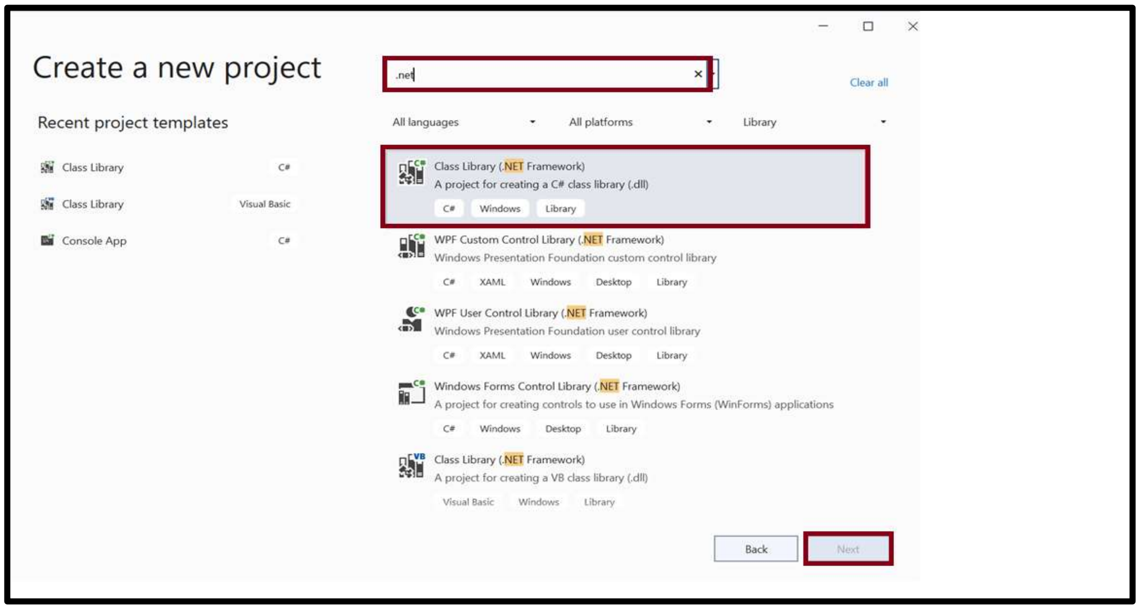
Open a project or solution
Open a local Visual Studio project or .sln file

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 →

2. In the search bar, type .net and then select Class Library (.NET Framework) and click Next.



Create a new project

.net

Clear all

Recent project templates

Class Library	C#
Class Library	Visual Basic
Console App	C#

All languages · All platforms · Library

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

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

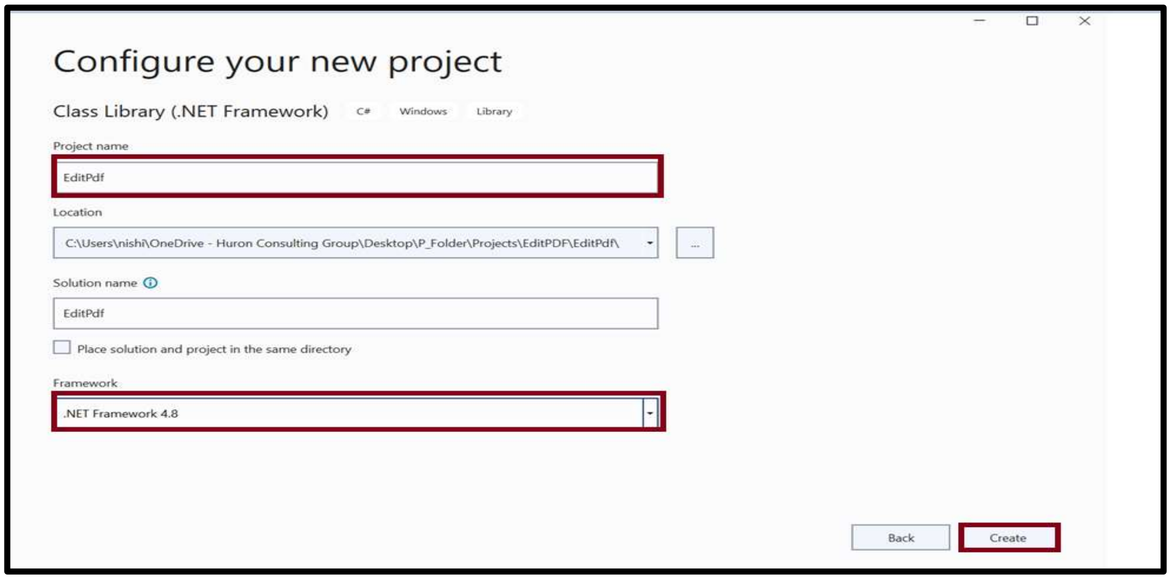
WPF User Control Library (.NET Framework)
Windows Presentation Foundation user control library
C# · XAML · Windows · Desktop · Library

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

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

Back · Next

3. Choose the target framework in which you want to build your custom activity and click on Create. (I have used .Net Framework 4.8)



Configure your new project

Class Library (.NET Framework) · C# · Windows · Library

Project name: EditPdf

Location: C:\Users\nishi\OneDrive - Huron Consulting Group\Desktop\P_Folder\Projects\EditPDF\EditPdf\

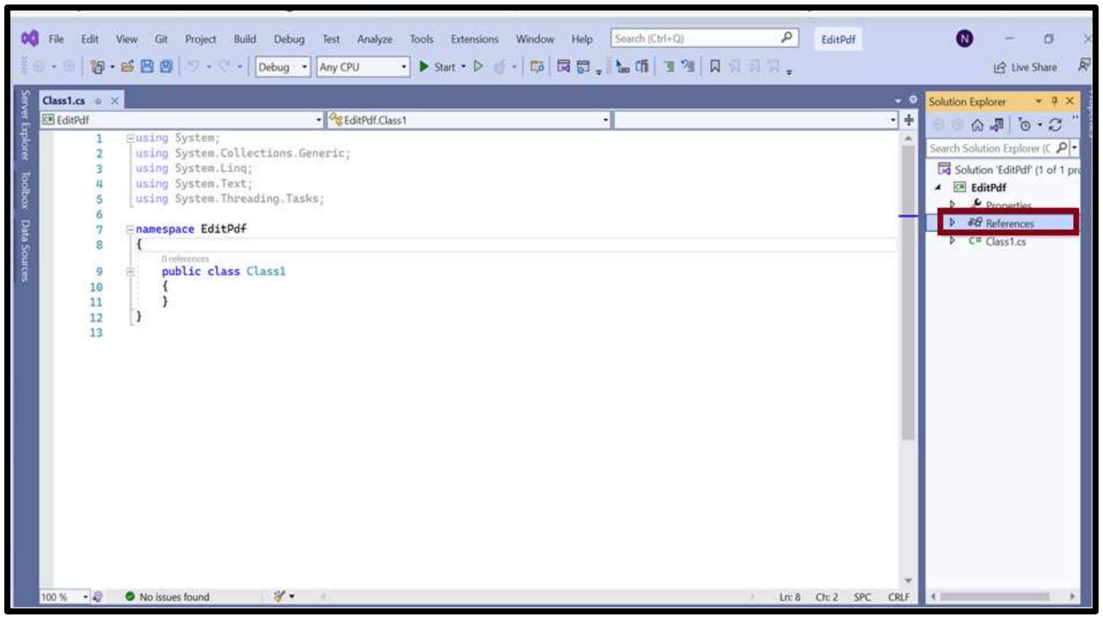
Solution name: EditPdf

☐ Place solution and project in the same directory

Framework: .NET Framework 4.8

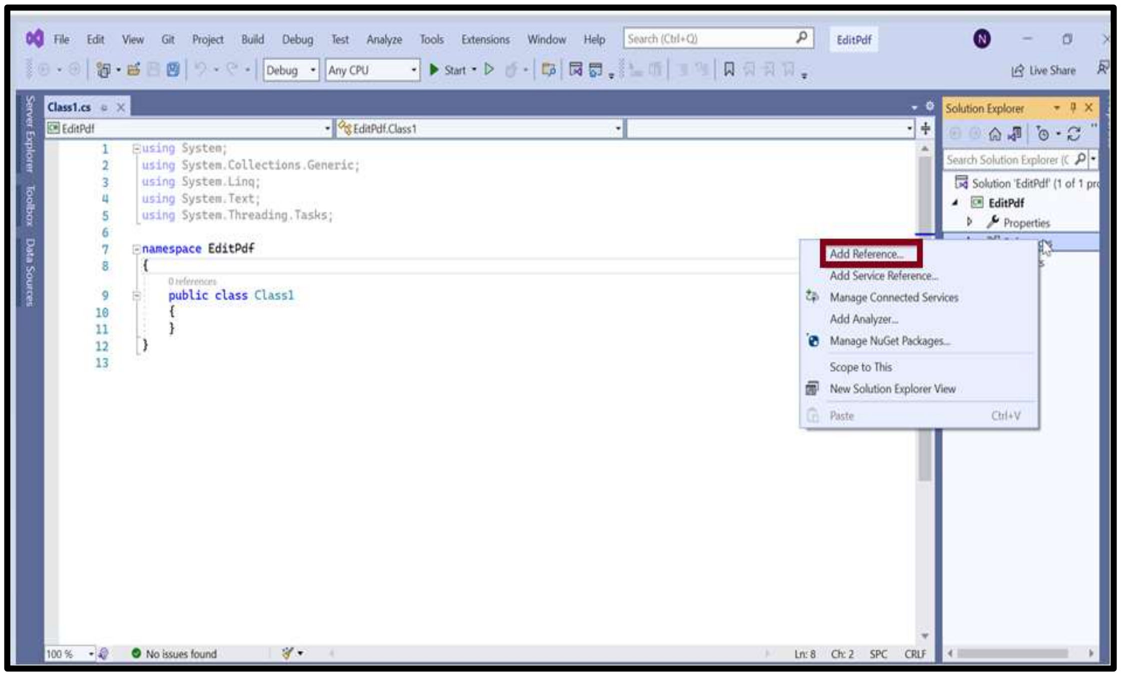
Back · Create

4. Select References from the top in the right pane.



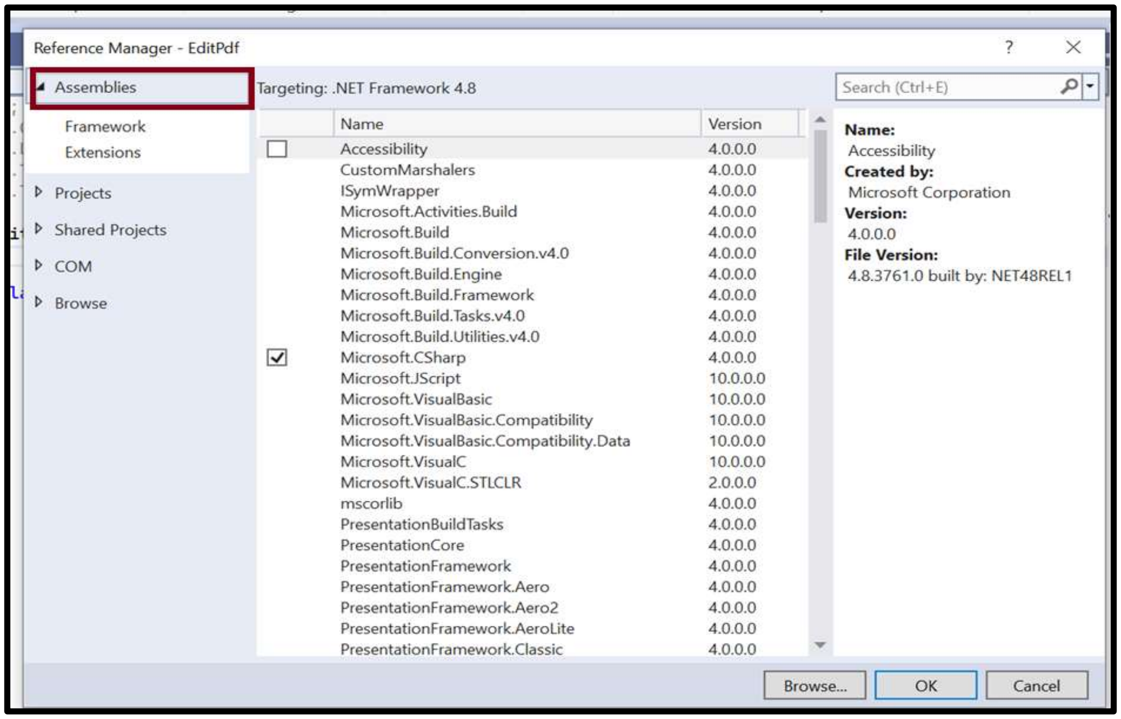
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace EditPdf
{
    0 references
    public class Class1
    {
    }
}
```

5. Right click on References and select Add Reference.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace EditPdf
{
    0 references
    public class Class1
    {
    }
}
```

6. Select Assemblies from the top in the left pane.



Reference Manager - EditPdf

- Assemblies
  - Framework
  - Extensions
- Projects
- Shared Projects
- COM
- Browse

Targeting: .NET Framework 4.8

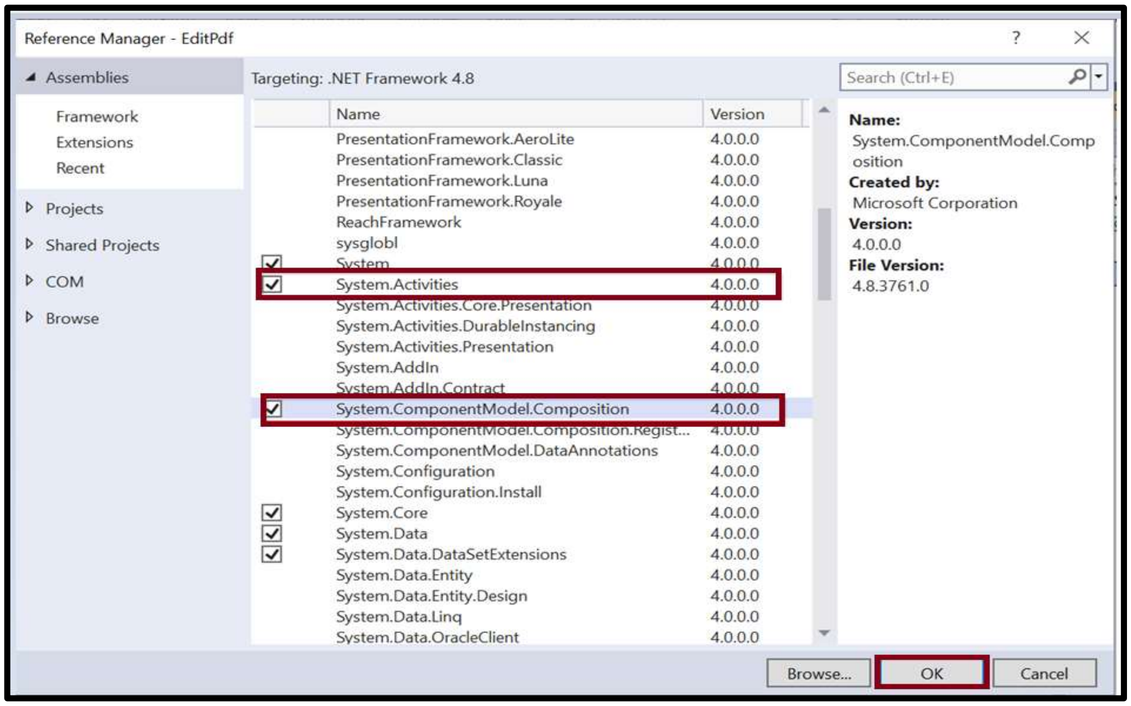
Search (Ctrl+E)

	Name	Version
☐	Accessibility	4.0.0.0
	CustomMarshalers	4.0.0.0
	ISymWrapper	4.0.0.0
	Microsoft.Activities.Build	4.0.0.0
	Microsoft.Build	4.0.0.0
	Microsoft.Build.Conversion.v4.0	4.0.0.0
	Microsoft.Build.Engine	4.0.0.0
	Microsoft.Build.Framework	4.0.0.0
	Microsoft.Build.Tasks.v4.0	4.0.0.0
	Microsoft.Build.Utilities.v4.0	4.0.0.0
✓	Microsoft.CSharp	4.0.0.0
	Microsoft.JScript	10.0.0.0
	Microsoft.VisualBasic	10.0.0.0
	Microsoft.VisualBasic.Compatibility	10.0.0.0
	Microsoft.VisualBasic.Compatibility.Data	10.0.0.0
	Microsoft.VisualC	10.0.0.0
	Microsoft.VisualC.STLCLR	2.0.0.0
	mscorlib	4.0.0.0
	PresentationBuildTasks	4.0.0.0
	PresentationCore	4.0.0.0
	PresentationFramework	4.0.0.0
	PresentationFramework.Aero	4.0.0.0
	PresentationFramework.Aero2	4.0.0.0
	PresentationFramework.AeroLite	4.0.0.0
	PresentationFramework.Classic	4.0.0.0

Name:
Accessibility
Created by:
Microsoft Corporation
Version:
4.0.0.0
File Version:
4.8.3761.0 built by: NET48REL1

Browse... OK Cancel

7. Select the highlighted assemblies and click on OK.



Reference Manager - EditPdf

- Assemblies
  - Framework
  - Extensions
  - Recent
- Projects
- Shared Projects
- COM
- Browse

Targeting: .NET Framework 4.8

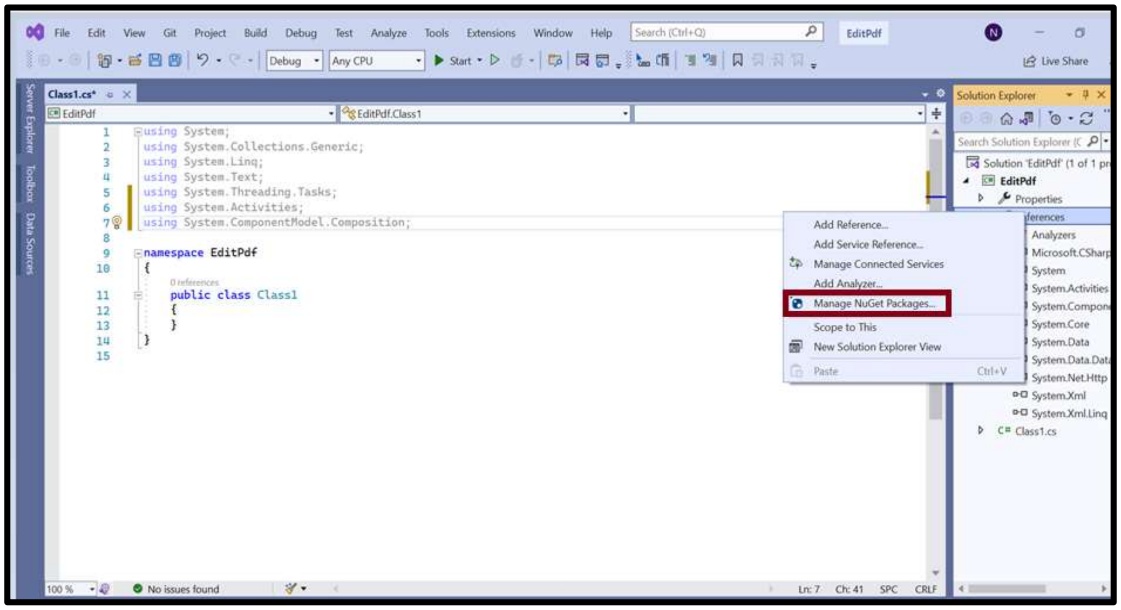
Search (Ctrl+E)

	Name	Version
	PresentationFramework.AeroLite	4.0.0.0
	PresentationFramework.Classic	4.0.0.0
	PresentationFramework.Luna	4.0.0.0
	PresentationFramework.Royale	4.0.0.0
	ReachFramework	4.0.0.0
	sysglobl	4.0.0.0
✓	System	4.0.0.0
✓	System.Activities	4.0.0.0
	System.Activities.Core.Presentation	4.0.0.0
	System.Activities.DurableInstancing	4.0.0.0
	System.Activities.Presentation	4.0.0.0
	System.AddIn	4.0.0.0
	System.AddIn.Contract	4.0.0.0
✓	System.ComponentModel.Composition	4.0.0.0
	System.ComponentModel.Composition.Regist...	4.0.0.0
	System.ComponentModel.DataAnnotations	4.0.0.0
	System.Configuration	4.0.0.0
	System.Configuration.Install	4.0.0.0
✓	System.Core	4.0.0.0
✓	System.Data	4.0.0.0
✓	System.Data.DataSetExtensions	4.0.0.0
	System.Data.Entity	4.0.0.0
	System.Data.Entity.Design	4.0.0.0
	System.Data.Linq	4.0.0.0
	System.Data.OracleClient	4.0.0.0

Name:
System.ComponentModel.Comp
osition
Created by:
Microsoft Corporation
Version:
4.0.0.0
File Version:
4.8.3761.0

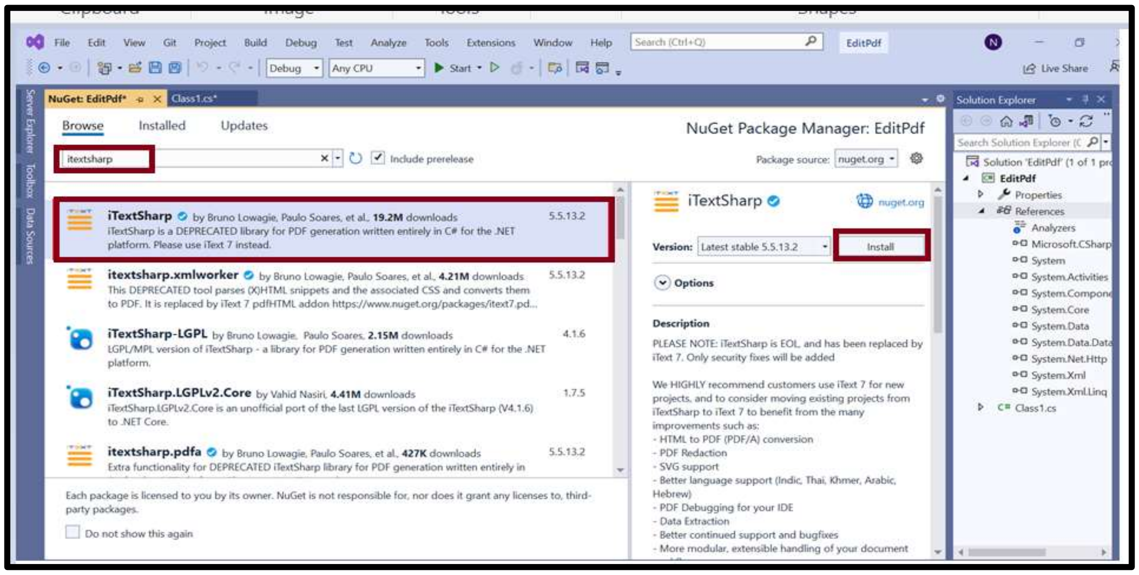
Browse... OK Cancel

8. Right-click on References in the right pane and select Manage NuGet Packages.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Activities;
using System.ComponentModel.Composition;

namespace EditPdf
{
    public class Class1
    {
    }
}
```

9. Type in itextsharp in the search bar. Select iTextSharp NuGet Package and click on Install.



iTextSharp by Bruno Lowagie, Paulo Soares, et al., **19.2M** downloads 5.5.13.2
iTextSharp is a DEPRECATED library for PDF generation written entirely in C# for the .NET platform. Please use iText 7 instead.

itextsharp.xmlworker by Bruno Lowagie, Paulo Soares, et al., **4.21M** downloads 5.5.13.2
This DEPRECATED tool parses (X)HTML snippets and the associated CSS and converts them to PDF. It is replaced by iText 7 pdfHTML addon https://www.nuget.org/packages/itext7.pd...

iTextSharp-LGPL by Bruno Lowagie, Paulo Soares, **2.15M** downloads 4.1.6
LGPL/MPL version of iTextSharp - a library for PDF generation written entirely in C# for the .NET platform.

iTextSharp.LGPLv2.Core by Vahid Nasiri, **4.41M** downloads 1.7.5
iTextSharp.LGPLv2.Core is an unofficial port of the last LGPL version of the iTextSharp (V4.1.6) to .NET Core.

itextsharp.pdfa by Bruno Lowagie, Paulo Soares, et al., **427K** downloads 5.5.13.2
Extra functionality for DEPRECATED iTextSharp library for PDF generation written entirely in

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

iTextSharp

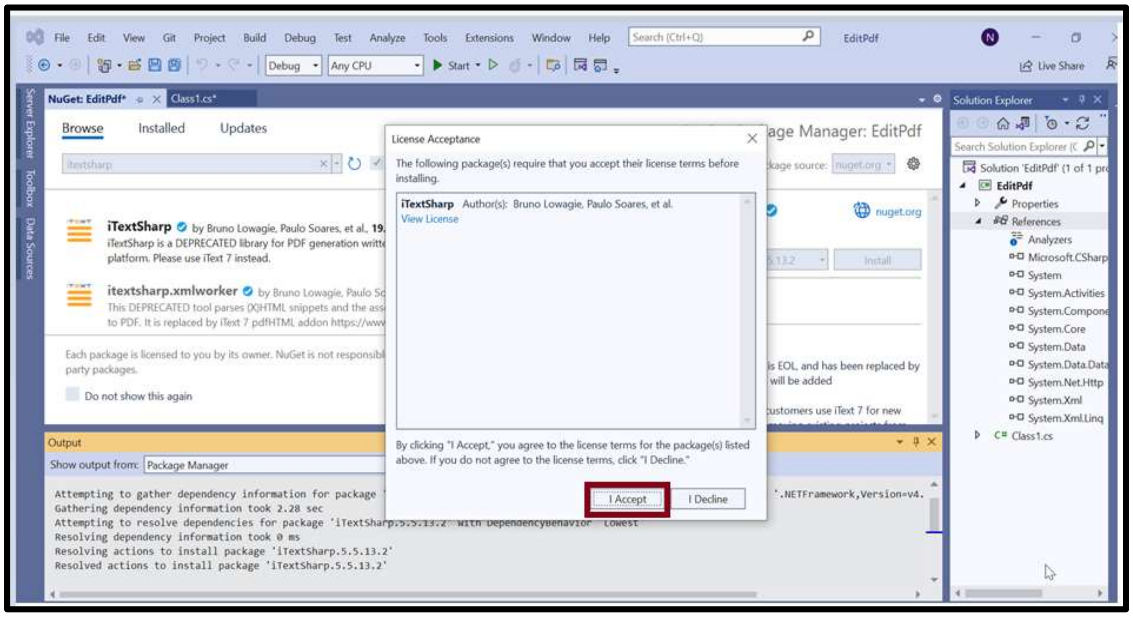
Version: Latest stable 5.5.13.2

Description

PLEASE NOTE: iTextSharp is EOL, and has been replaced by iText 7. Only security fixes will be added

We HIGHLY recommend customers use iText 7 for new projects, and to consider moving existing projects from iTextSharp to iText 7 to benefit from the many improvements such as:
- HTML to PDF (PDF/A) conversion
- PDF Redaction
- SVG support
- Better language support (Indic, Thai, Khmer, Arabic, Hebrew)
- PDF Debugging for your IDE
- Data Extraction
- Better continued support and bugfixes
- More modular, extensible handling of your document

10.Click on I Accept if you agree with the license terms for the package.



The following package(s) require that you accept their license terms before installing.

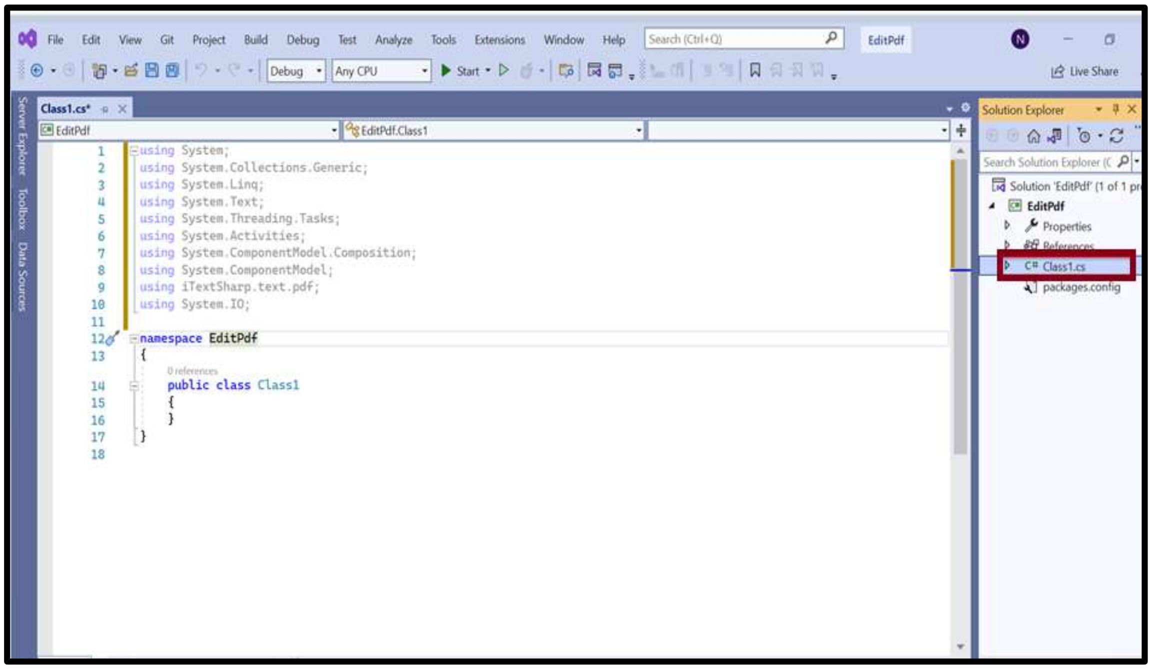
iTextSharp Author(s): Bruno Lowagie, Paulo Soares, et al.

By clicking "I Accept," you agree to the license terms for the package(s) listed above. If you do not agree to the license terms, click "I Decline."

Show output from: Package Manager

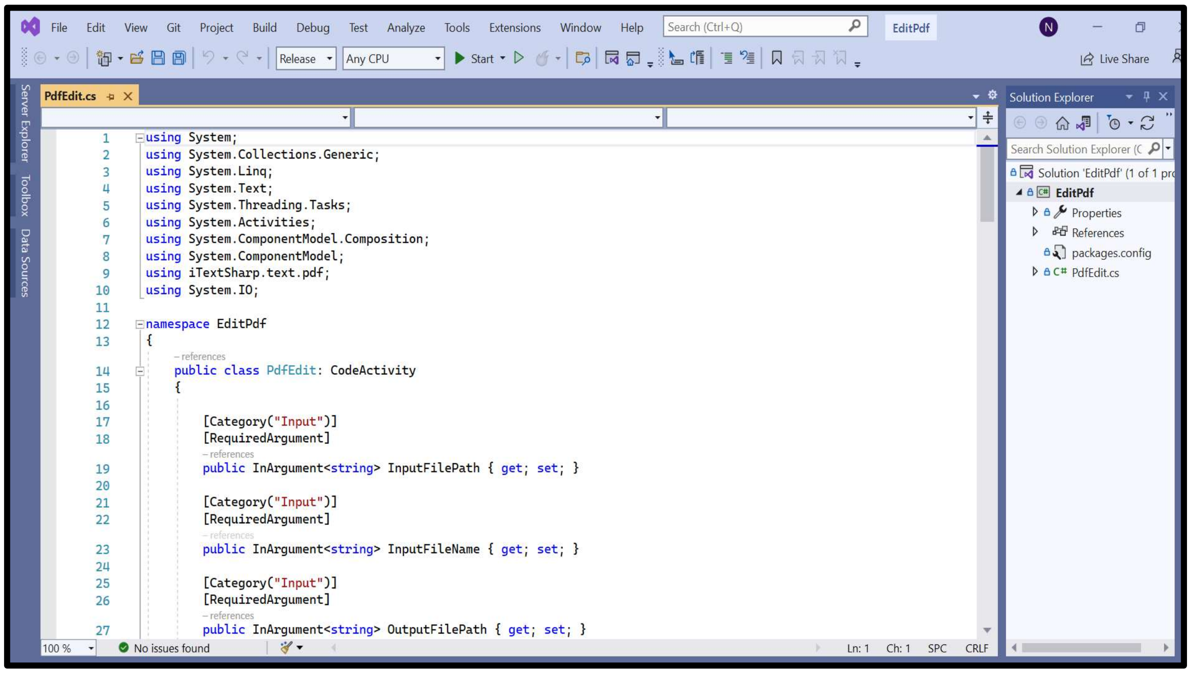
```
Attempting to gather dependency information for package '                    '.NETFramework,Version=v4.
Gathering dependency information took 2.28 sec
Attempting to resolve dependencies for package 'iTextSharp.5.5.13.2' with DependencyBehavior 'Lowest'
Resolving dependency information took 0 ms
Resolving actions to install package 'iTextSharp.5.5.13.2'
Resolved actions to install package 'iTextSharp.5.5.13.2'
```

11.Right-click the class1.cs file and select Rename.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Activities;
using System.ComponentModel.Composition;
using System.ComponentModel;
using iTextSharp.text.pdf;
using System.IO;

namespace EditPdf
{
    0 references
    public class Class1
    {
    }
}
```

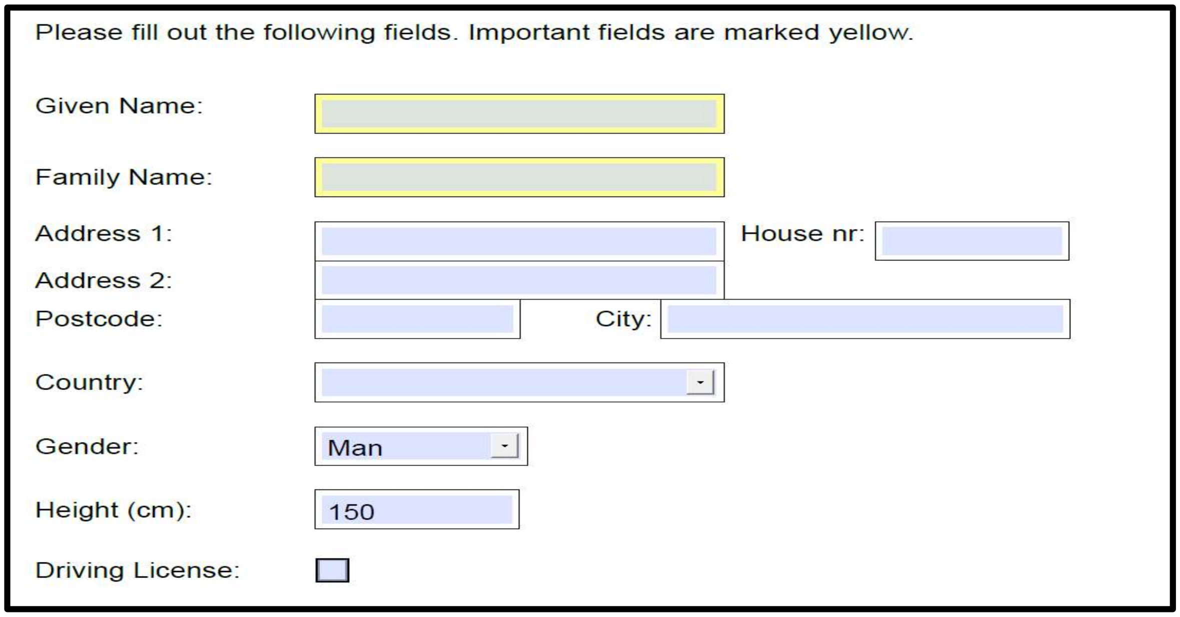
12. Copy the code from the repository - https://github.com/Nishi80100/EditPdf



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Activities;
using System.ComponentModel.Composition;
using System.ComponentModel;
using iTextSharp.text.pdf;
using System.IO;

namespace EditPdf
{
    - references
    public class PdfEdit: CodeActivity
    {

        [Category("Input")]
        [RequiredArgument]
        - references
        public InArgument<string> InputFilePath { get; set; }

        [Category("Input")]
        [RequiredArgument]
        - references
        public InArgument<string> InputFileName { get; set; }

        [Category("Input")]
        [RequiredArgument]
        - references
        public InArgument<string> OutputFilePath { get; set; }
```

13. Understanding the necessary modifications based on your pdf file.

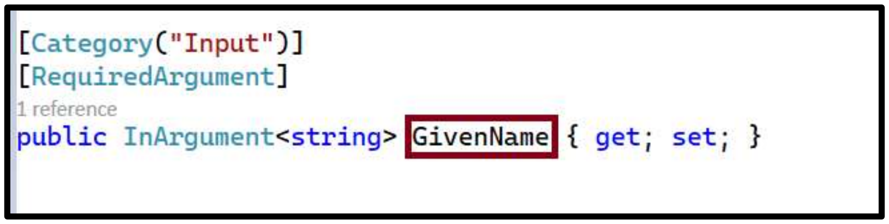
Input pdf file:



Please fill out the following fields. Important fields are marked yellow.

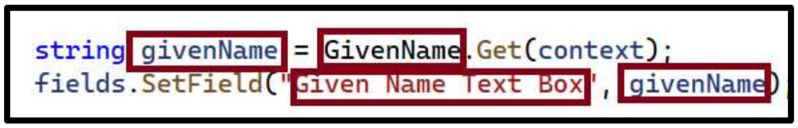
Given Name:			
Family Name:			
Address 1:		House nr:	
Address 2:			
Postcode:		City:	
Country:			
Gender:	Man		
Height (cm):	150		
Driving License:	☐		

Code Modification:



```
[Category("Input")]
[RequiredArgument]
1 reference
public InArgument<string> GivenName { get; set; }
```

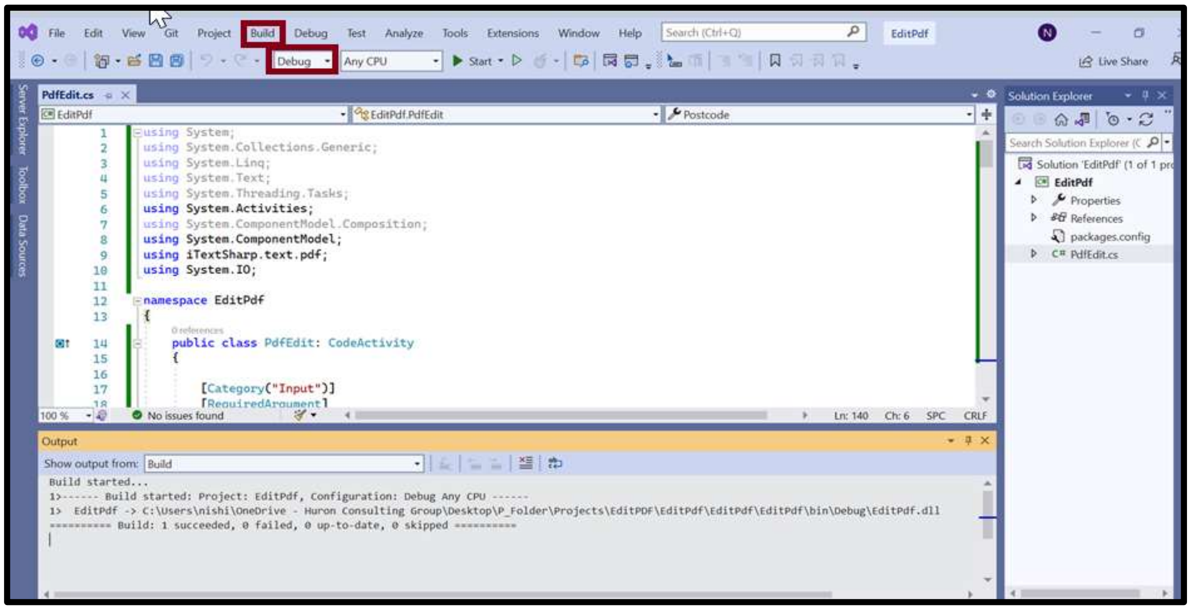
So, as you can see, I need to fill in the data against GivenName in the Input pdf file so I've specified the field Given Name in my code. The same applies to your pdf file.



```
string givenName = GivenName.Get(context);
fields.SetField("Given Name Text Box", givenName);
```

In this case, **givenName** represents the string that will be used to store the user input. Replace it with the name that fits your pdf file best. In the previous step, we declared the field name **GivenName**. **Given Name Text Box** represents the field names that we have received as final output after following the steps mentioned under ‘Developing a console application in C# to read the field names from pdf’.

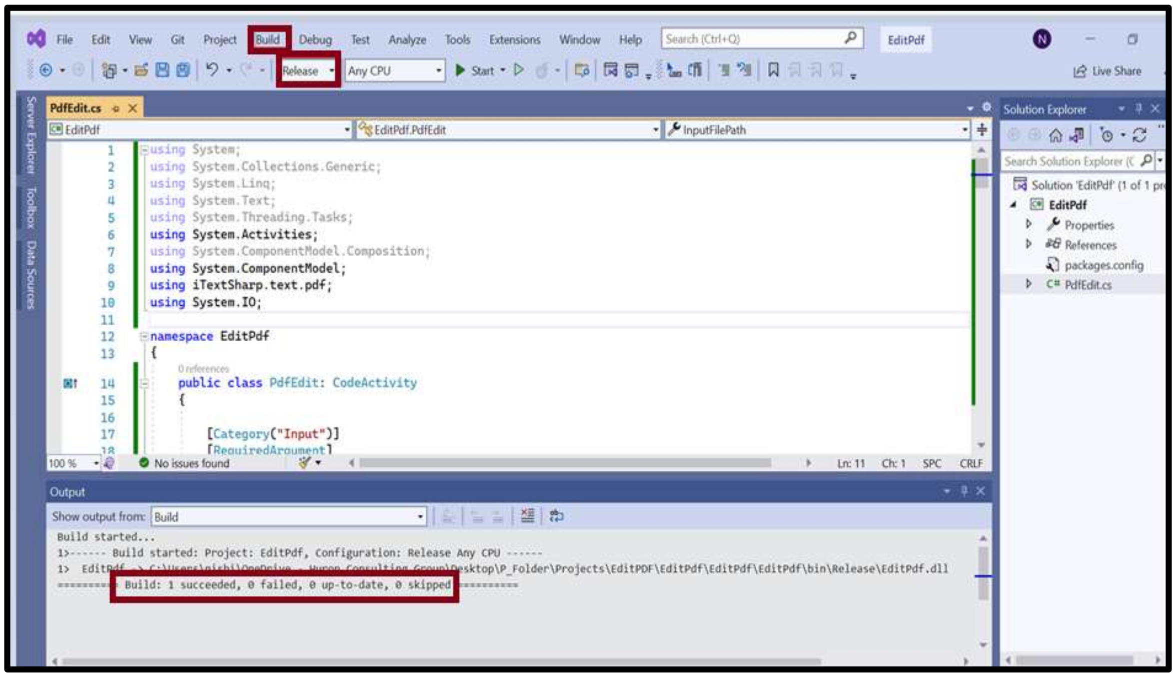
14.Select the execution mode as Debug, then build, and finally click on Build Solution.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Activities;
using System.ComponentModel.Composition;
using System.ComponentModel;
using iTextSharp.text.pdf;
using System.IO;

namespace EditPdf
{
    0 references
    public class PdfEdit: CodeActivity
    {

        [Category("Input")]
        [RequiredArgument]
```

```
Build started...
1>------ Build started: Project: EditPdf, Configuration: Debug Any CPU ------
1>  EditPdf -> C:\Users\nishi\OneDrive - Huron Consulting Group\Desktop\P_Folder\Projects\EditPDF\EditPdf\EditPdf\EditPdf\bin\Debug\EditPdf.dll
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

15. Select the execution mode as Release, then build, and finally click on Build solution. Ensure that your build has succeeded.

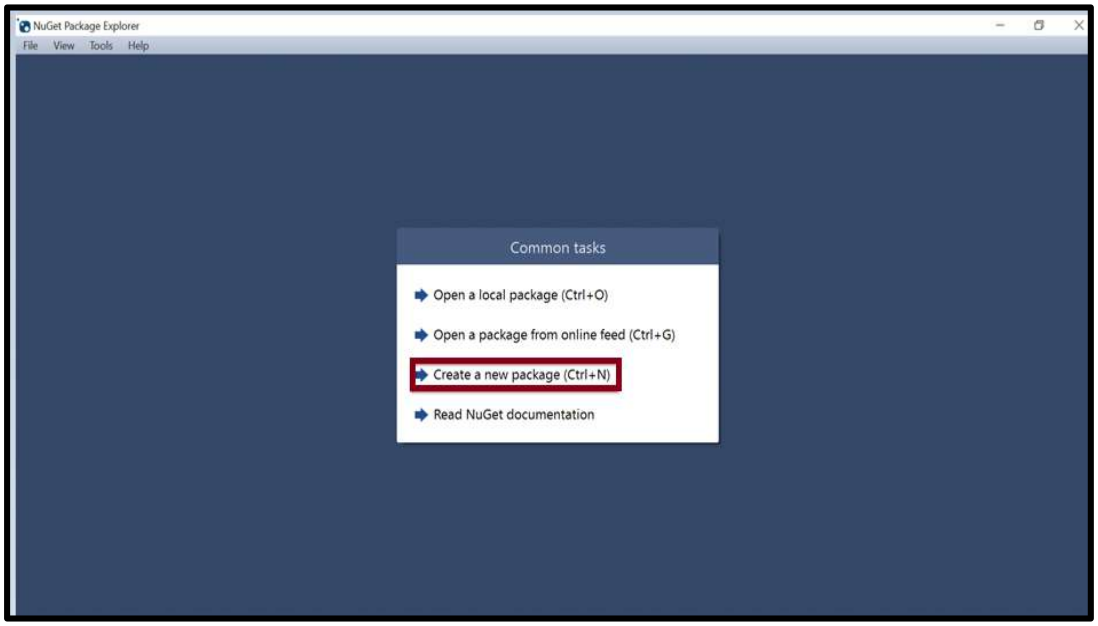


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Activities;
using System.ComponentModel.Composition;
using System.ComponentModel;
using iTextSharp.text.pdf;
using System.IO;

namespace EditPdf
{
    0 references
    public class PdfEdit: CodeActivity
    {

        [Category("Input")]
        [RequiredArgument]
```

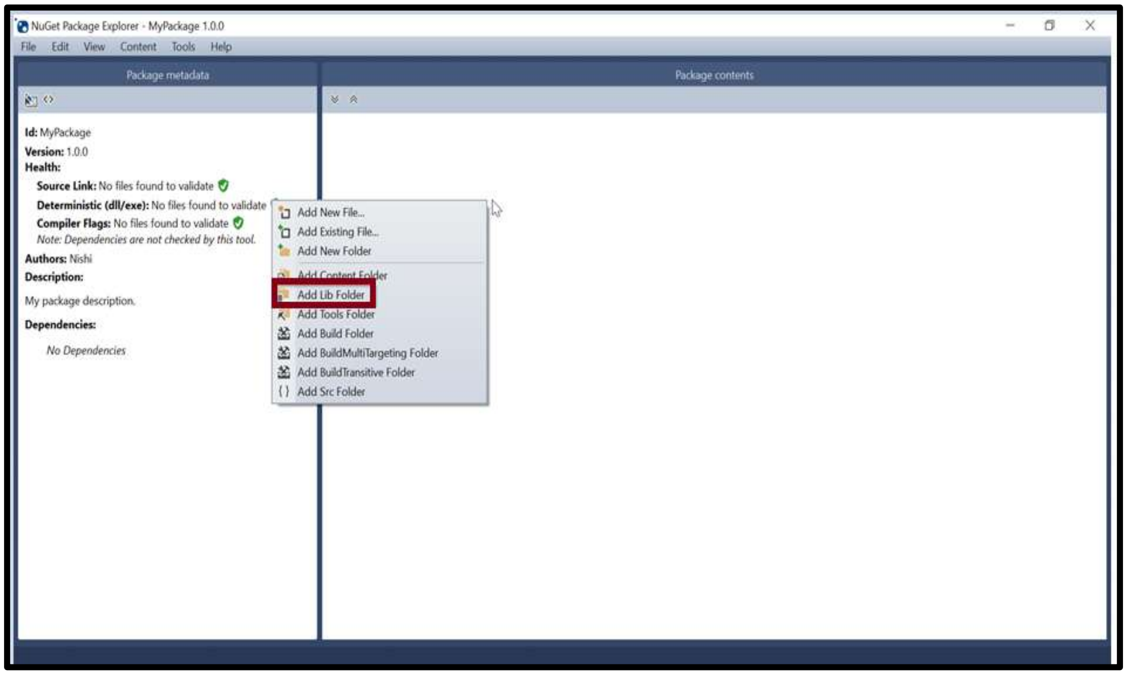
```
Build started...
1>------ Build started: Project: EditPdf, Configuration: Release Any CPU ------
1>  EditPdf -> C:\Users\nishi\OneDrive - Huron Consulting Group\Desktop\P_Folder\Projects\EditPDF\EditPdf\EditPdf\EditPdf\bin\Release\EditPdf.dll
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

Converting DLL generated from above step to NuGet package

1. Open Nuget package explorer and select create a new package.

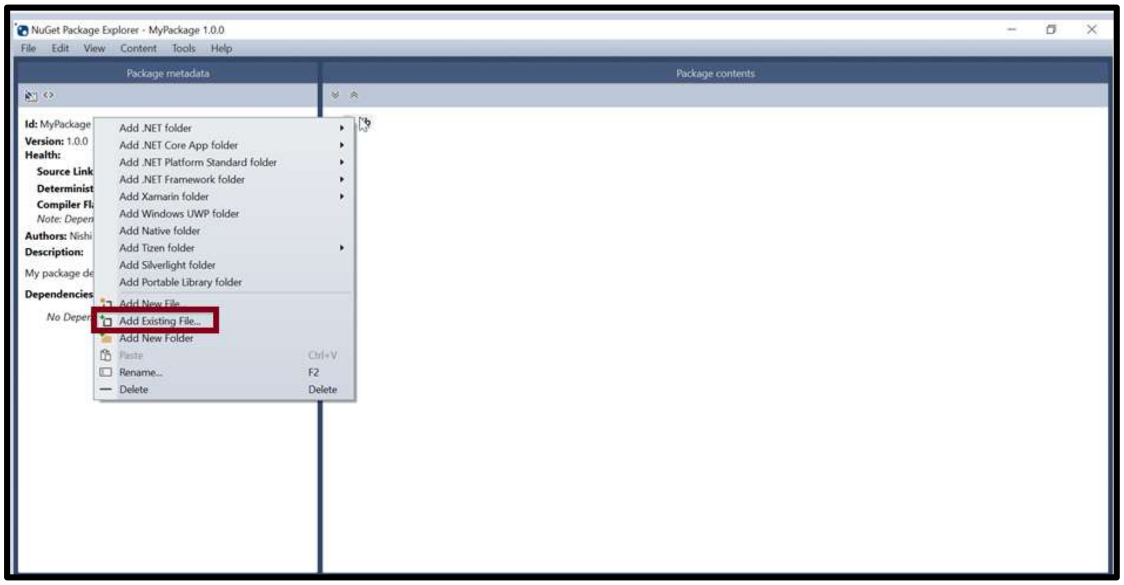

NuGet Package Explorer
File
View
Tools
Help
Common tasks
Open a local package (Ctrl+O)
Open a package from online feed (Ctrl+G)
Create a new package (Ctrl+N)
Read NuGet documentation

2. Right click on Package content section and select Add Lib Folder.

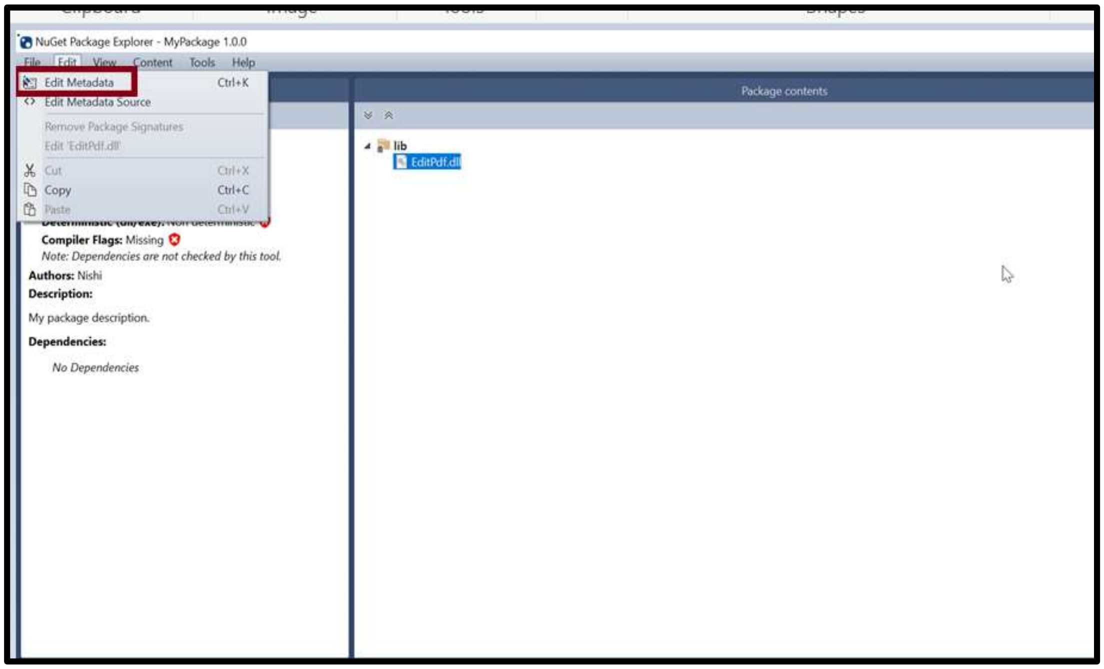


Id: MyPackage
Version: 1.0.0
Health:
Source Link: No files found to validate
Deterministic (dll/exe): No files found to validate
Compiler Flags: No files found to validate
Note: Dependencies are not checked by this tool.
Authors: Nishi
Description:
My package description.
Dependencies:
No Dependencies

3. Right click on Lib and select Add Existing File. You can find the dll file inside the folder where you have created your Custom activity code. Let's say you have created the code inside ReadPdfWidget folder then you can find the dll inside ReadPdfWidget\ReadPdfWidget\bin\Release\net6.0

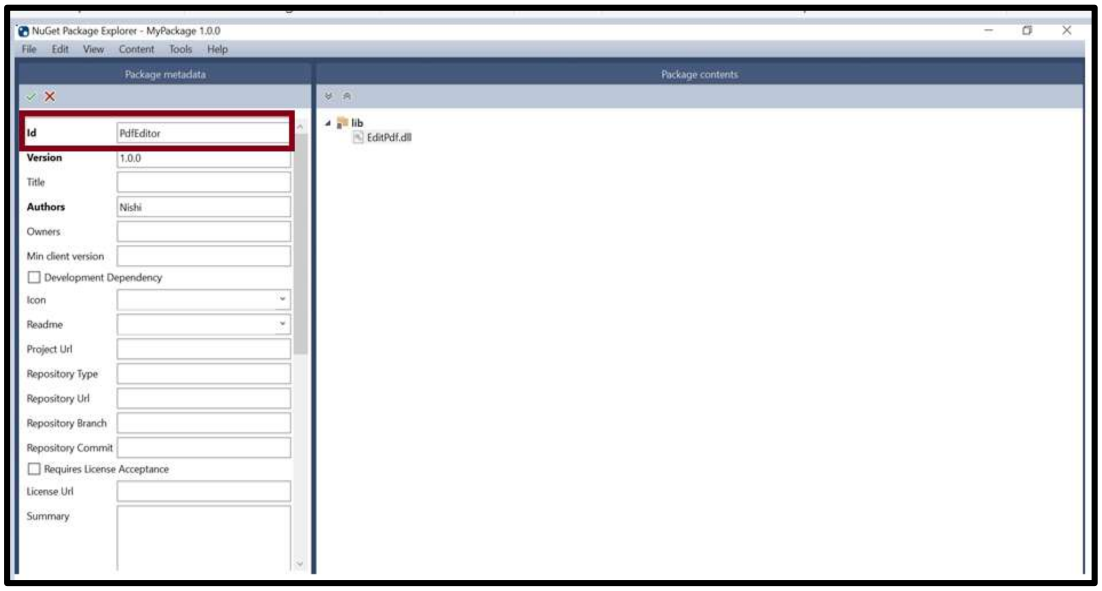


4. Select Edit tab and then click on Edit Metadata.



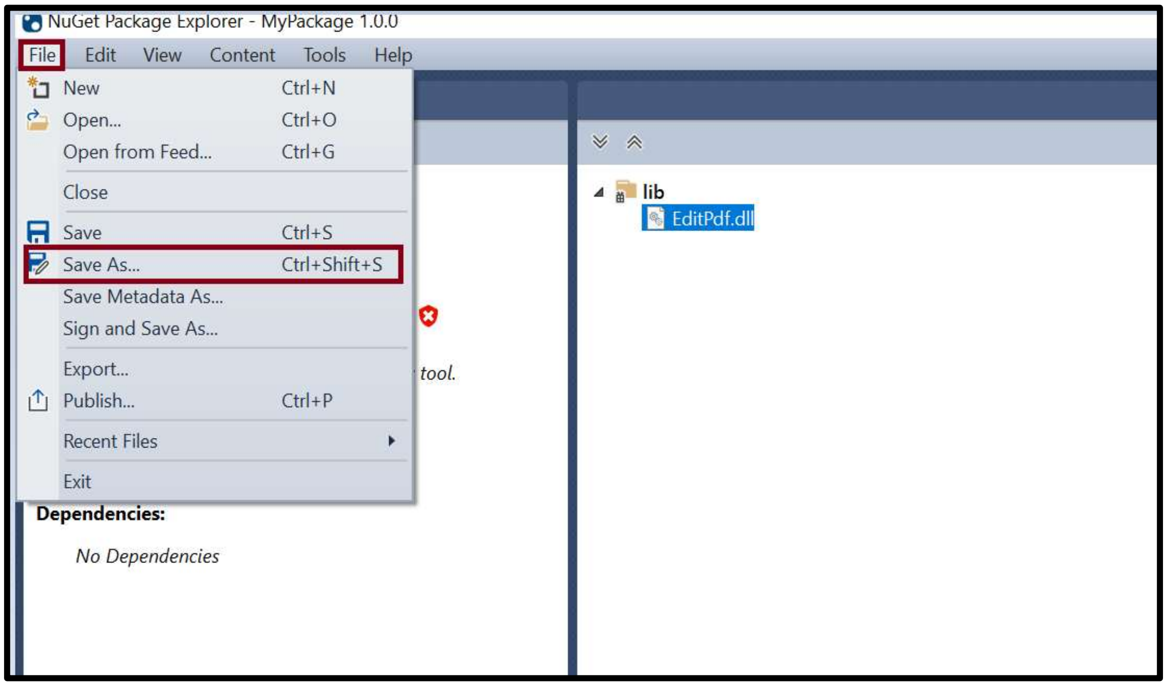
Compiler Flags: Missing
Note: Dependencies are not checked by this tool.
Authors: Nishi
Description:
My package description.
Dependencies:
No Dependencies

5. Fill in the name you want to give to your Nuget Package in the Id field.

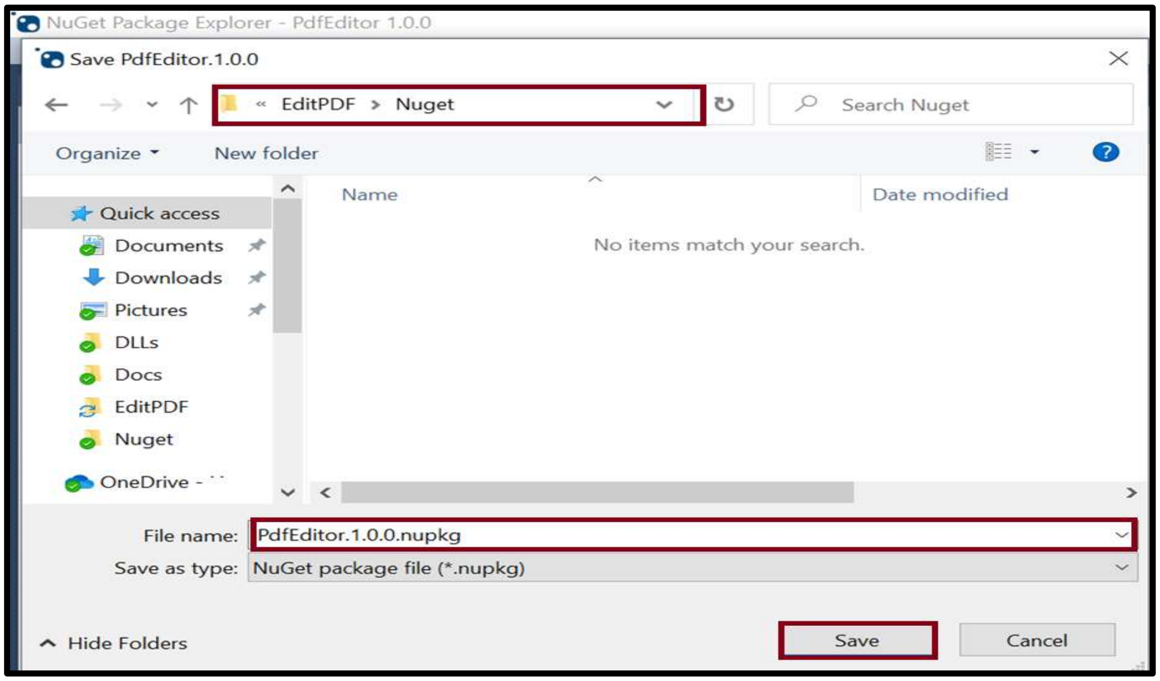


Id	PdfEditor
Version	1.0.0
Title	
Authors	Nishi
Owners	
Min client version	
☐ Development Dependency	
Icon	
Readme	
Project Url	
Repository Type	
Repository Url	
Repository Branch	
Repository Commit	
☐ Requires License Acceptance	
License Url	
Summary	

6. Select File tab, then Save As and specify the location where you want to save your Nuget Package.

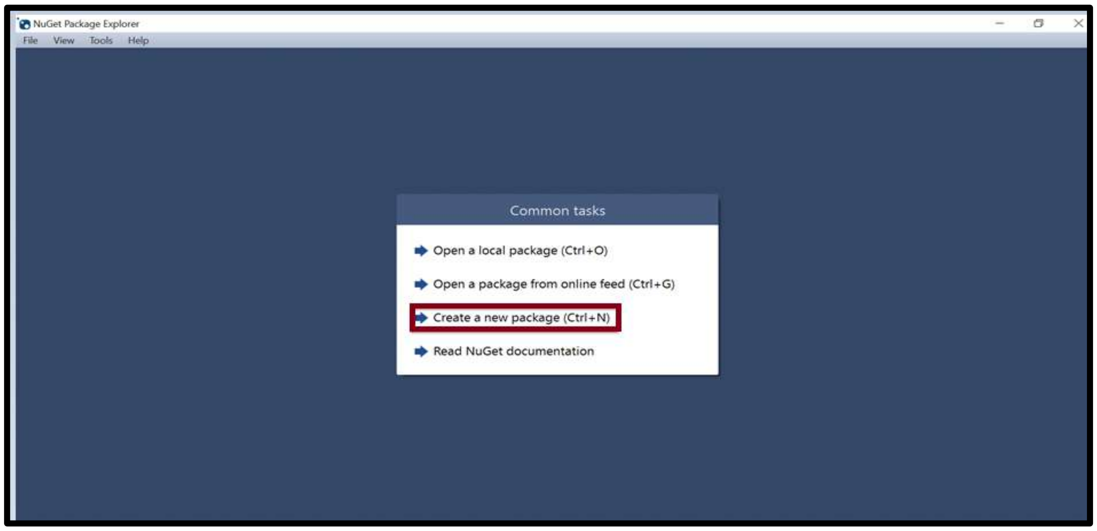


7. Type in the name you want to give to your Nuget Package and click on Save.

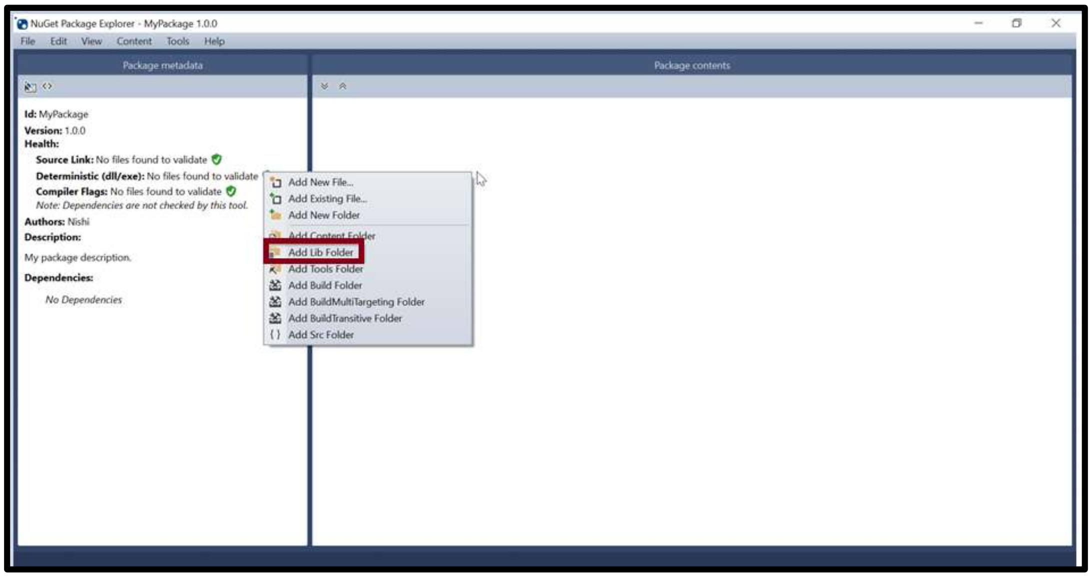


Now the next step is to create NuGet package for the iTextsharp dll file.

1. Open Nuget package explorer and select Create a new package.

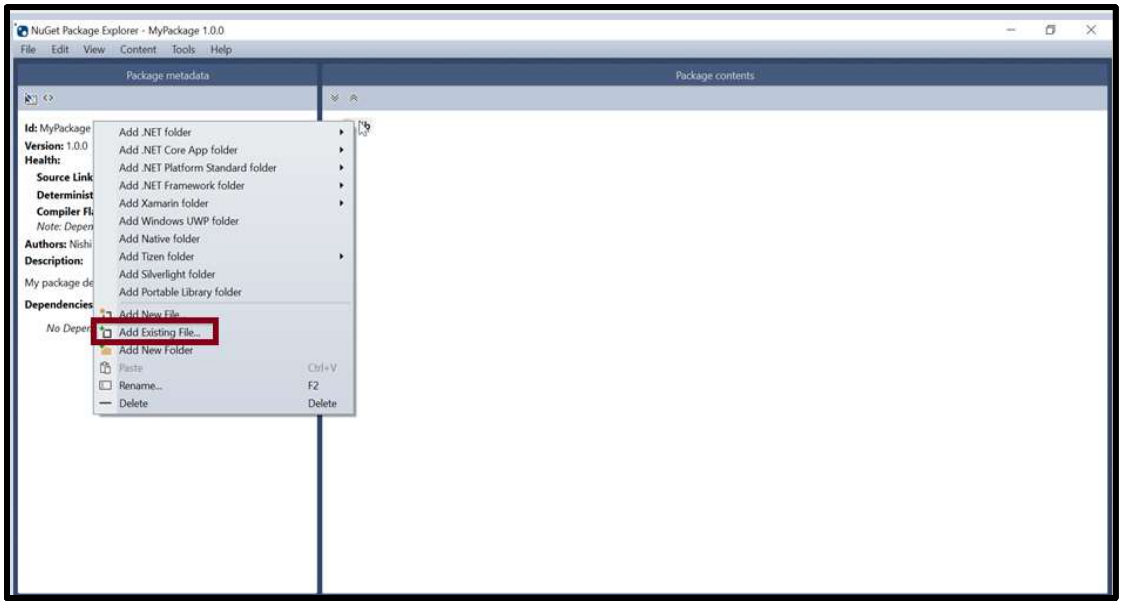

NuGet Package Explorer
File View Tools Help
Common tasks
Open a local package (Ctrl+O)
Open a package from online feed (Ctrl+G)
Create a new package (Ctrl+N)
Read NuGet documentation

2. Right click on Package Content section and select Add Lib Folder.

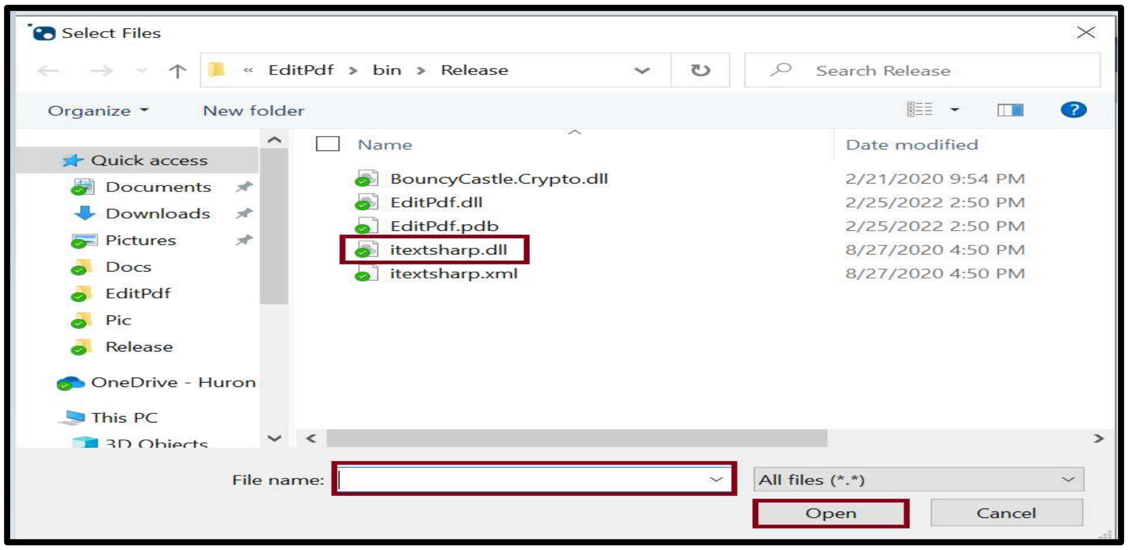


Id: MyPackage
Version: 1.0.0
Health:
Source Link: No files found to validate
Deterministic (dll/exe): No files found to validate
Compiler Flags: No files found to validate
Note: Dependencies are not checked by this tool.
Authors: Nishi
Description:
My package description.
Dependencies:
No Dependencies

3. Right click on Lib and select Add Existing File.

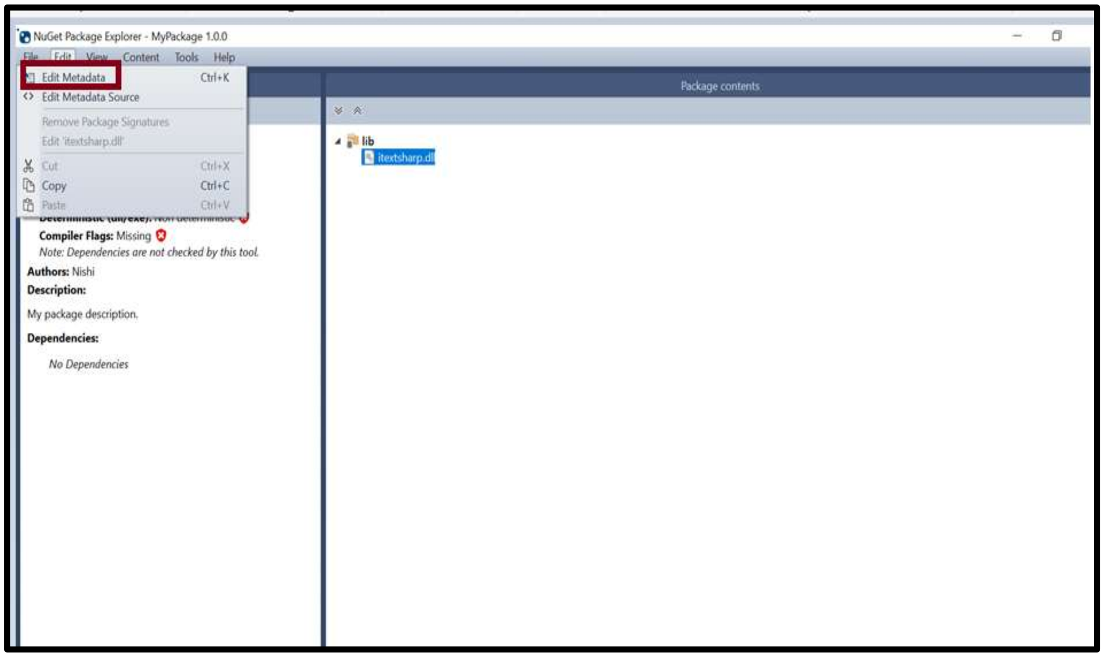


4. Specify the location of your itextsharp dll file. You can find the dll file inside the folder where you have created your Custom activity code. Let's say you have created the code inside ReadPdfWidget folder then you can find the dll inside ReadPdfWidget\ReadPdfWidget\bin\Debug\net6.0



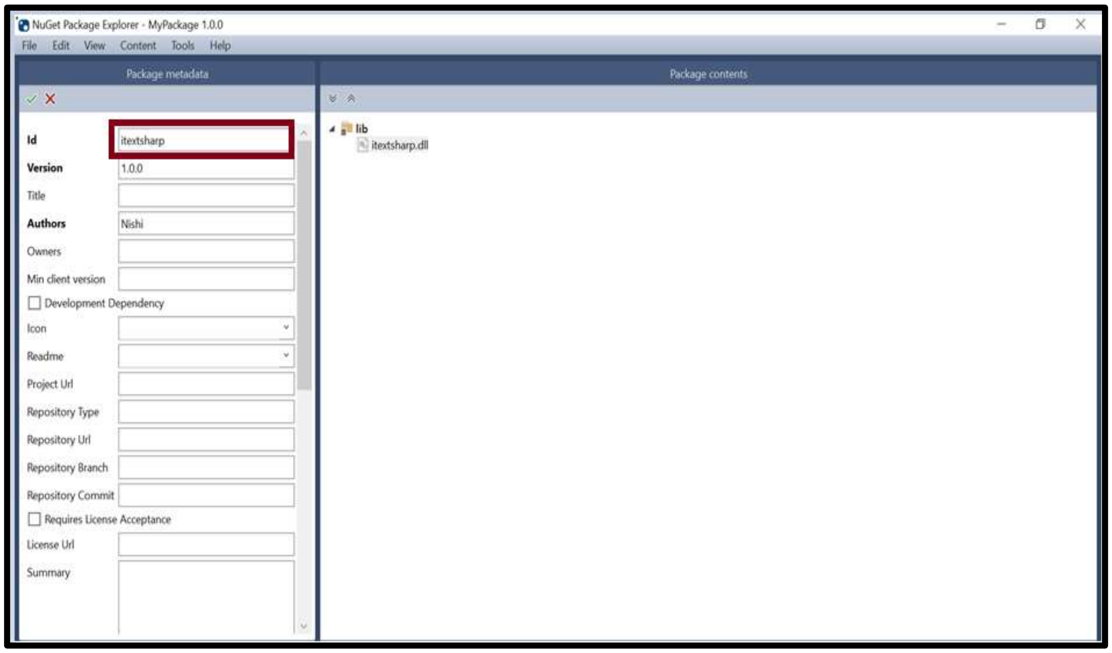
Name	Date modified
BouncyCastle.Crypto.dll	2/21/2020 9:54 PM
EditPdf.dll	2/25/2022 2:50 PM
EditPdf.pdb	2/25/2022 2:50 PM
itextsharp.dll	8/27/2020 4:50 PM
itextsharp.xml	8/27/2020 4:50 PM

5. Select Edit Tab and then Edit Metadata.



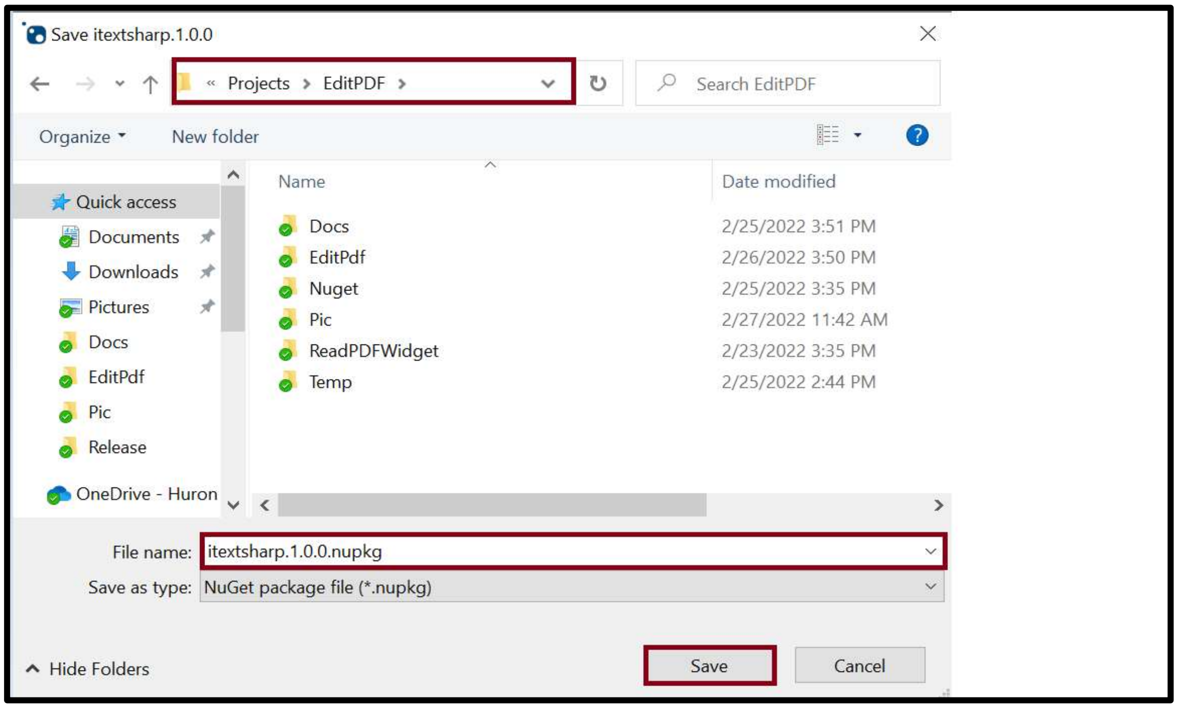
Compiler Flags: Missing
Note: Dependencies are not checked by this tool.
Authors: Nishi
Description:
My package description.
Dependencies:
No Dependencies

6. Fill in the name you want to give to your NuGet package in the Id field.


NuGet Package Explorer - MyPackage 1.0.0
File Edit View Content Tools Help
Package metadata
Package contents
lib
itextsharp.dll

Id	itextsharp
Version	1.0.0
Title	
Authors	Nishi
Owners	
Min client version	
☐ Development Dependency	
Icon	
Readme	
Project Url	
Repository Type	
Repository Url	
Repository Branch	
Repository Commit	
☐ Requires License Acceptance	
License Url	
Summary	

7. Select File tab, then Save As and specify the location where you want to save your NuGet Package.



Save itextsharp.1.0.0

« Projects › EditPDF ›

Search EditPDF

Organize ▾ New folder

Name	Date modified
Docs	2/25/2022 3:51 PM
EditPdf	2/26/2022 3:50 PM
Nuget	2/25/2022 3:35 PM
Pic	2/27/2022 11:42 AM
ReadPDFWidget	2/23/2022 3:35 PM
Temp	2/25/2022 2:44 PM

Quick access: Documents, Downloads, Pictures, Docs, EditPdf, Pic, Release; OneDrive - Huron

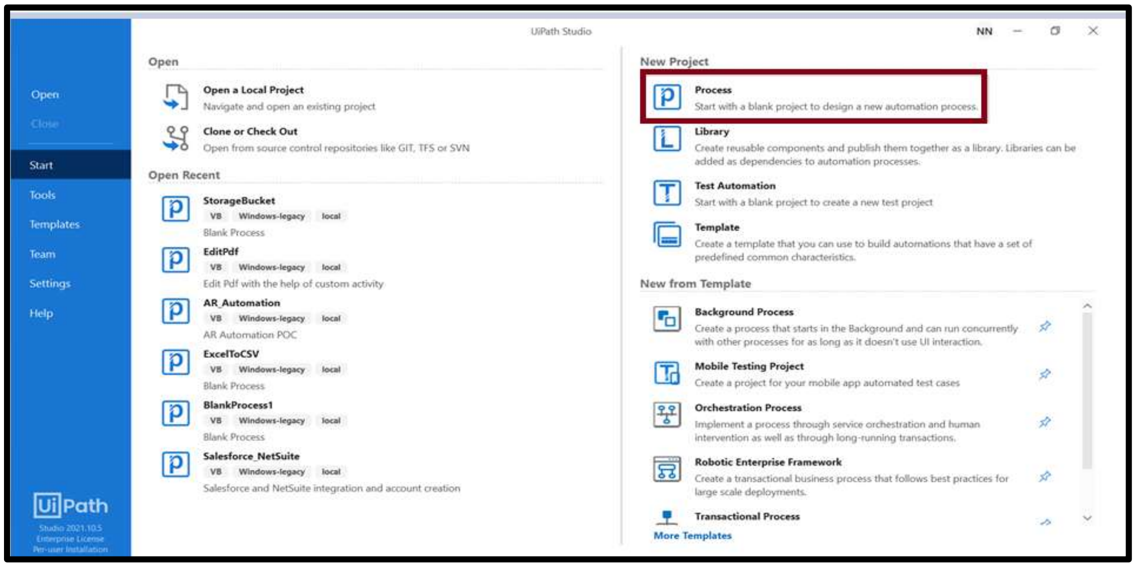
File name: itextsharp.1.0.0.nupkg

Save as type: NuGet package file (*.nupkg)

Hide Folders — Save — Cancel

Configuring NuGet package in UiPath Studio

1. Open UiPath Studio and select Process.



UiPath Studio

Open

Open a Local Project
Navigate and open an existing project

Clone or Check Out
Open from source control repositories like GIT, TFS or SVN

Open Recent

StorageBucket — VB Windows-legacy local
Blank Process

EditPdf — VB Windows-legacy local
Edit Pdf with the help of custom activity

AR_Automation — VB Windows-legacy local
AR Automation POC

ExcelToCSV — VB Windows-legacy local
Blank Process

BlankProcess1 — VB Windows-legacy local
Blank Process

Salesforce_NetSuite — VB Windows-legacy local
Salesforce and NetSuite integration and account creation

New Project

Process
Start with a blank project to design a new automation process.

Library
Create reusable components and publish them together as a library. Libraries can be added as dependencies to automation processes.

Test Automation
Start with a blank project to create a new test project

Template
Create a template that you can use to build automations that have a set of predefined common characteristics.

New from Template

Background Process
Create a process that starts in the Background and can run concurrently with other processes for as long as it doesn't use UI interaction.

Mobile Testing Project
Create a project for your mobile app automated test cases

Orchestration Process
Implement a process through service orchestration and human intervention as well as through long-running transactions.

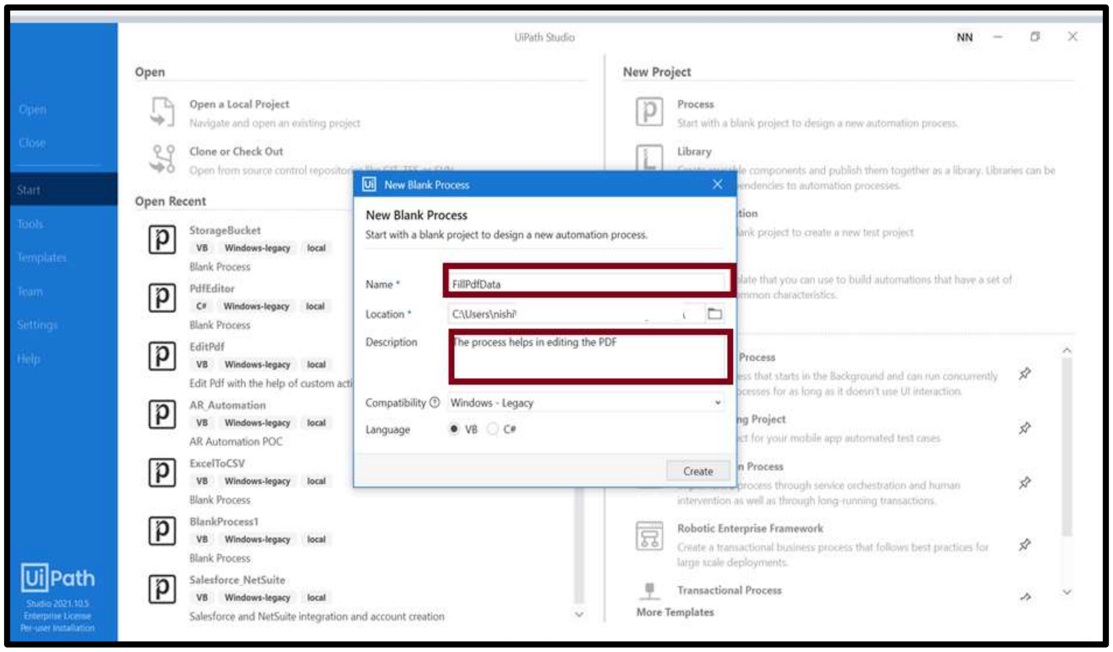
Robotic Enterprise Framework
Create a transactional business process that follows best practices for large scale deployments.

Transactional Process

More Templates

2. Fill in the Process name and Description of your choice.

Note: Process name and NuGet Package names should not be the same otherwise it will cause a dependency cycle, which will prevent NuGet Package installation.



New Blank Process

Start with a blank project to design a new automation process.

Name * FillPdfData

Location * C:\Users\nishi\

Description The process helps in editing the PDF

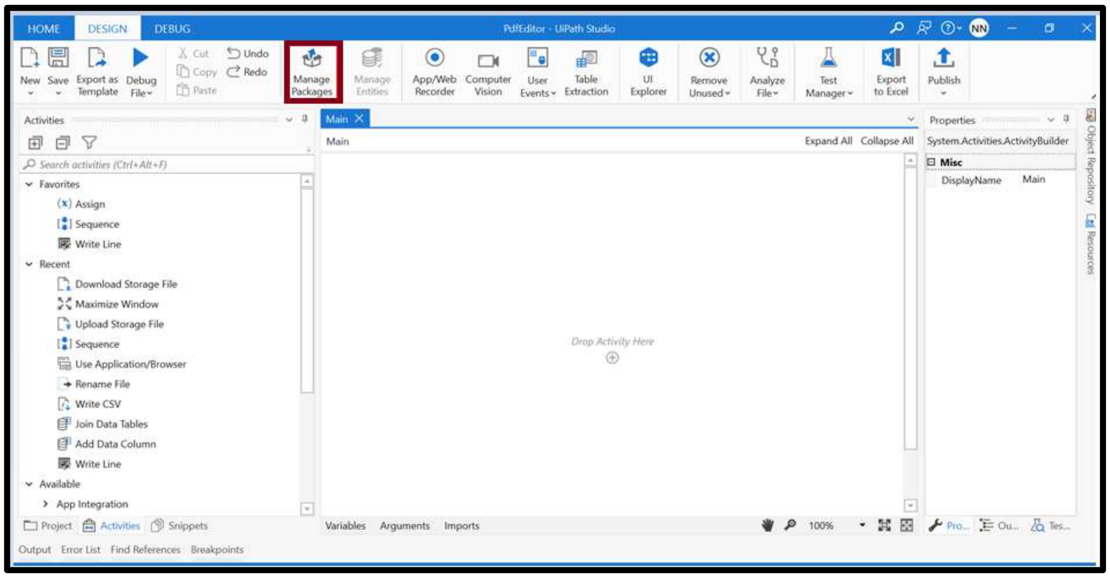
Compatibility Windows - Legacy

Language [✓ VB] C#

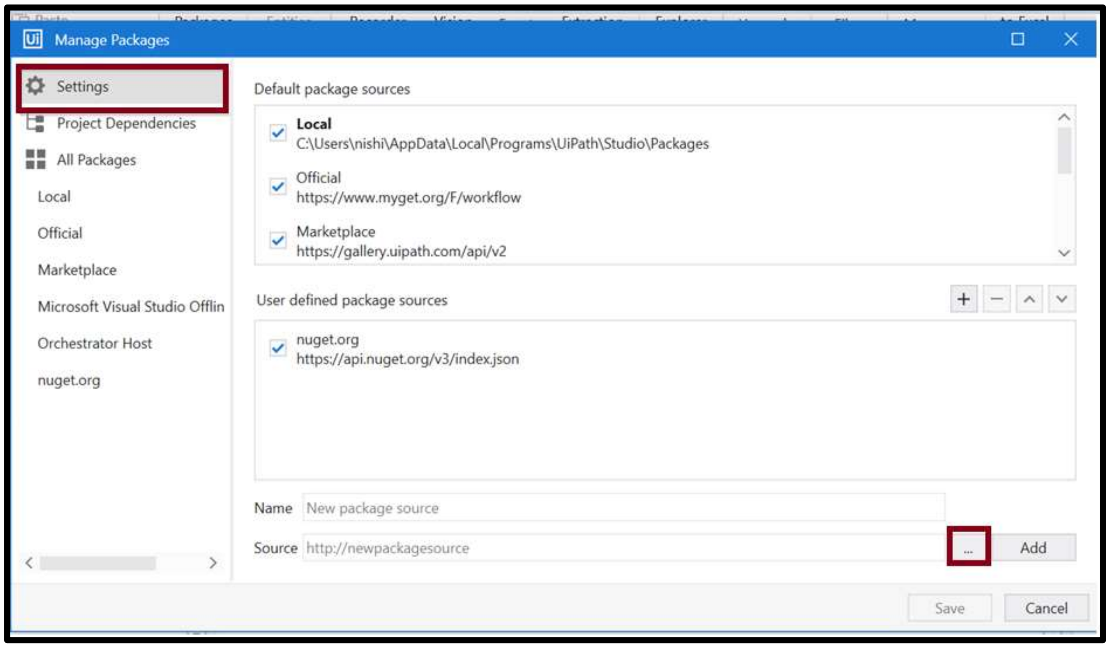
Create

The next step is to install the NuGet package as a dependency for your Project.

3. Click on Manage Packages.



4. Select Settings tab and click on ellipsis icon next to Source.



Default package sources

Local
C:\Users\nishi\AppData\Local\Programs\UiPath\Studio\Packages

Official
https://www.myget.org/F/workflow

Marketplace
https://gallery.uipath.com/api/v2

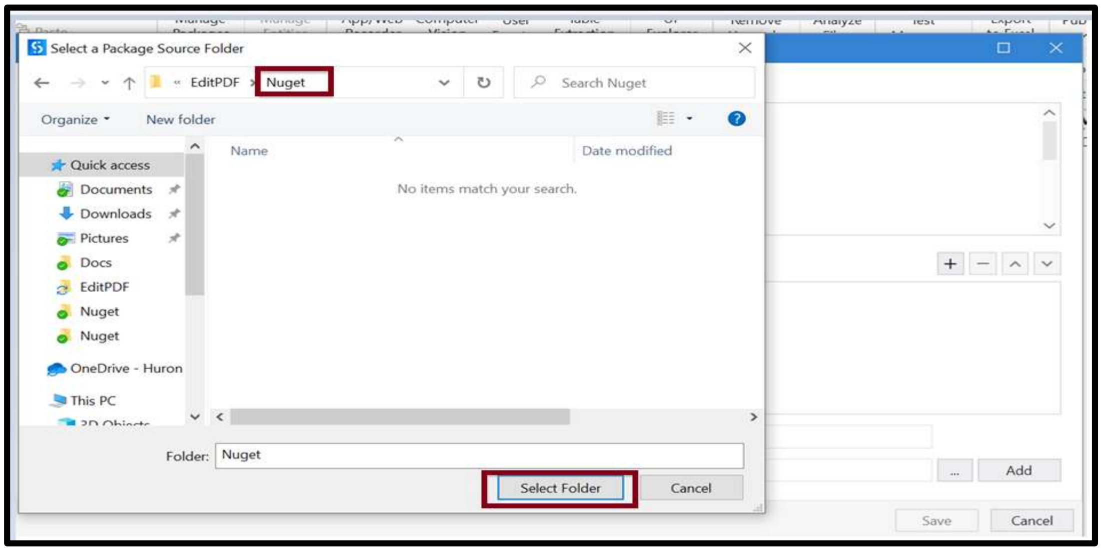
User defined package sources

nuget.org
https://api.nuget.org/v3/index.json

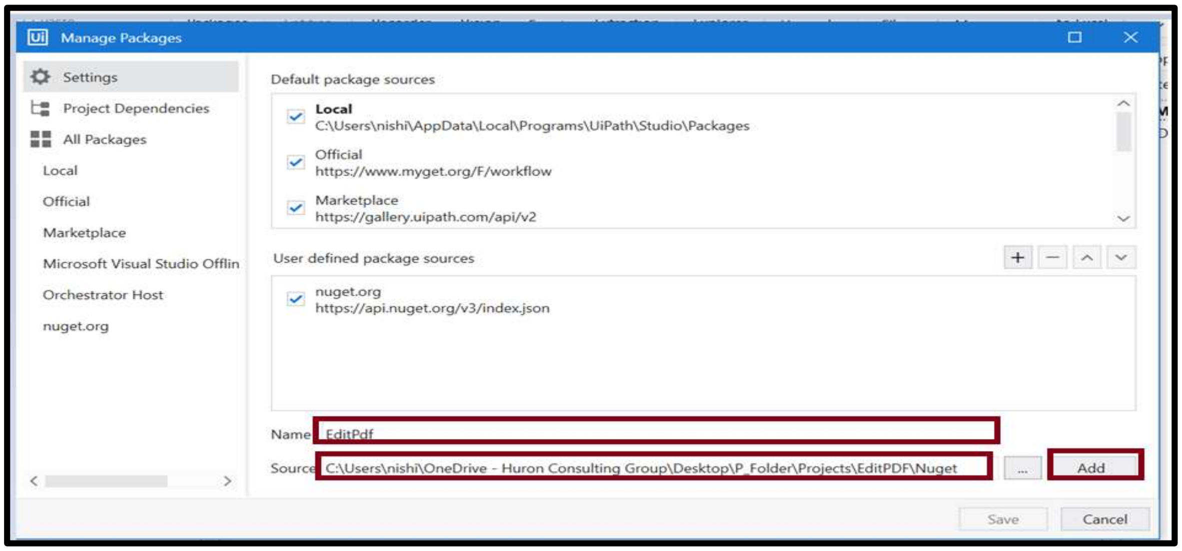
Name: New package source

Source: http://newpackagesource

5. Select NuGet Package folder. (The folder where you have saved your itextsharp and PdfEditor NuGet Package)
   Note: You just need to indicate the folder where you have saved your NuGet Packages. When you select the folder, you won't see the .nupkg files.



6. Specify the name and click on Add. Source will be the folder location that you have selected in the above step.



Default package sources

- ✓ Local C:\Users\nishi\AppData\Local\Programs\UiPath\Studio\Packages
- ✓ Official https://www.myget.org/F/workflow
- ✓ Marketplace https://gallery.uipath.com/api/v2

User defined package sources

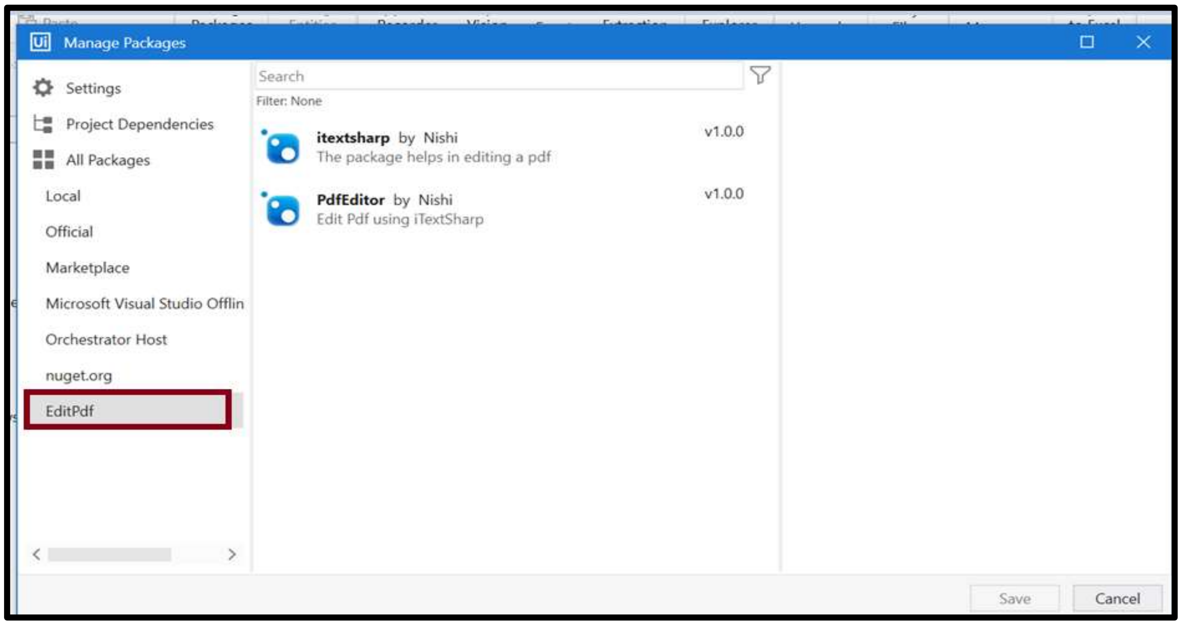
- ✓ nuget.org https://api.nuget.org/v3/index.json

Name EditPdf

Source C:\Users\nishi\OneDrive - Huron Consulting Group\Desktop\P_Folder\Projects\EditPDF\Nuget

Add

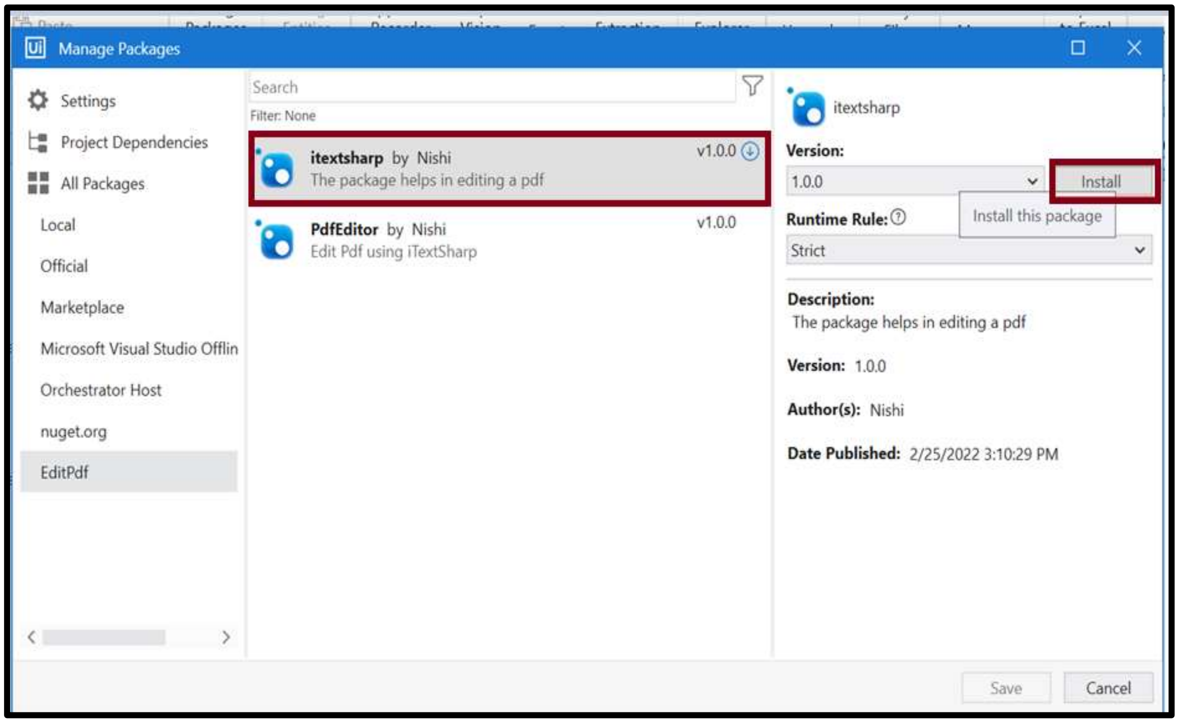
7. Select the newly added NuGet Package folder.



itextsharp by Nishi v1.0.0
The package helps in editing a pdf

PdfEditor by Nishi v1.0.0
Edit Pdf using iTextSharp

8. Select itextsharp NuGet package and click on Install.



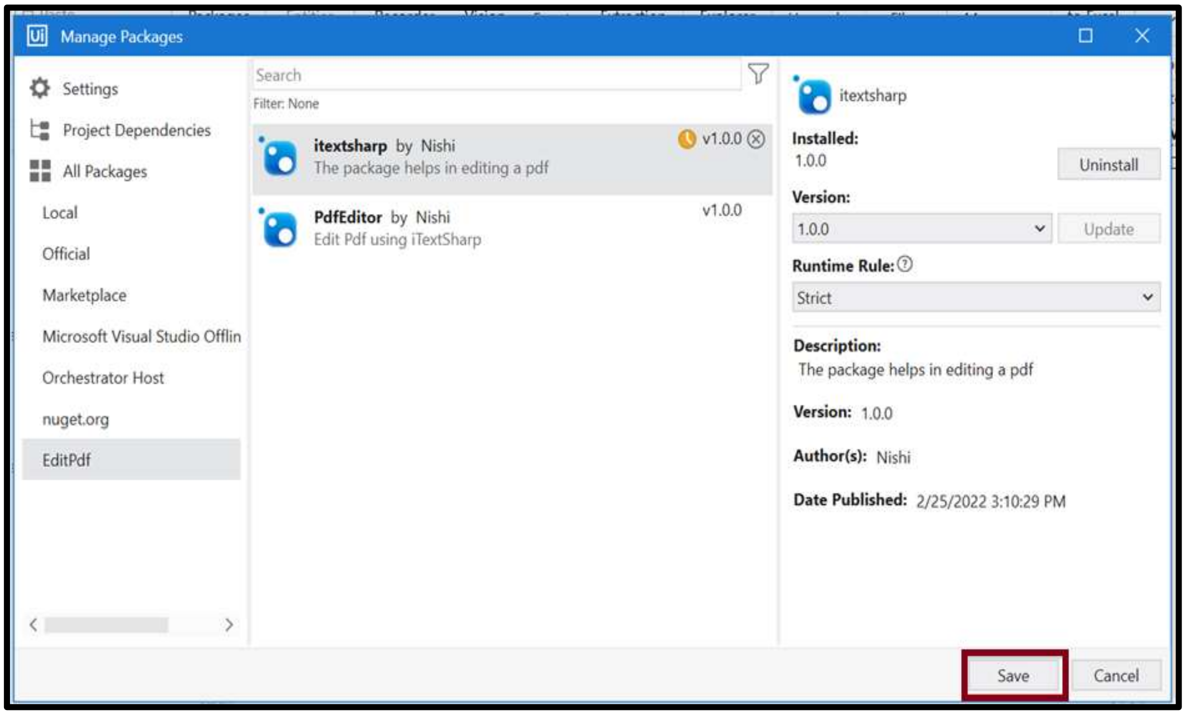
Description: The package helps in editing a pdf

Version: 1.0.0

Author(s): Nishi

Date Published: 2/25/2022 3:10:29 PM

9. Click on Save.



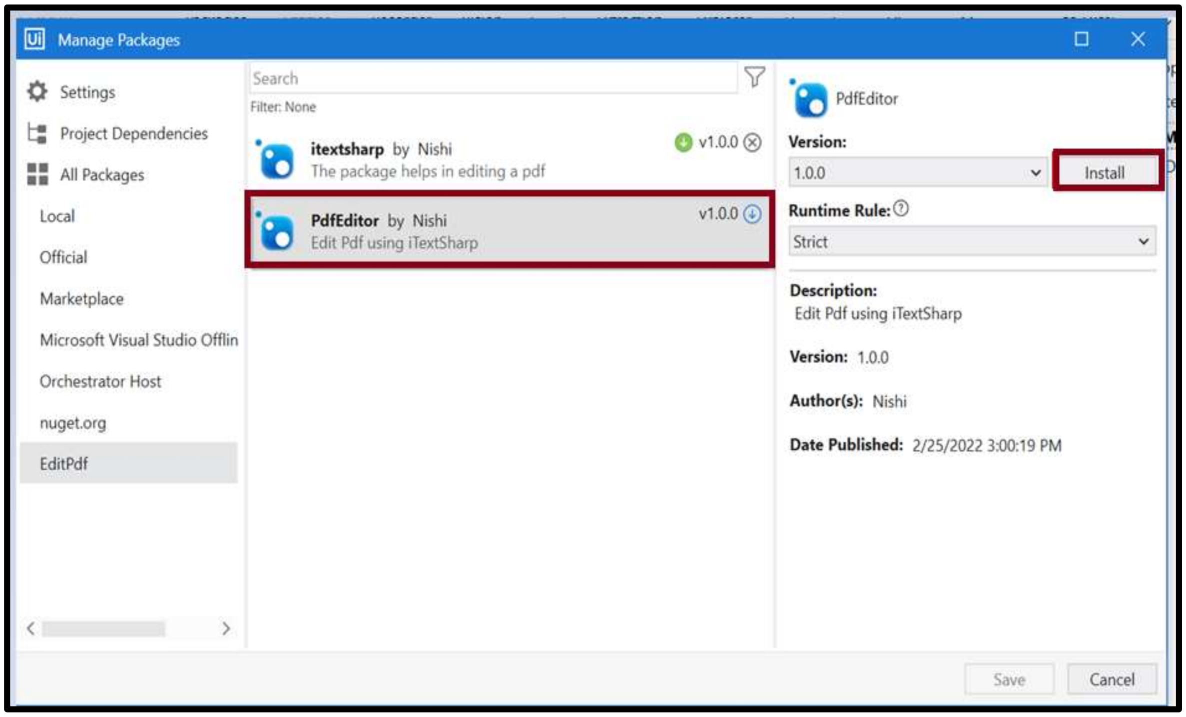
Description: The package helps in editing a pdf

Version: 1.0.0

Author(s): Nishi

Date Published: 2/25/2022 3:10:29 PM

10.Select PdfEditor NuGet Package and click on Install.



itextsharp by Nishi v1.0.0
The package helps in editing a pdf

PdfEditor by Nishi v1.0.0
Edit Pdf using iTextSharp

PdfEditor

Version: 1.0.0

Runtime Rule: Strict

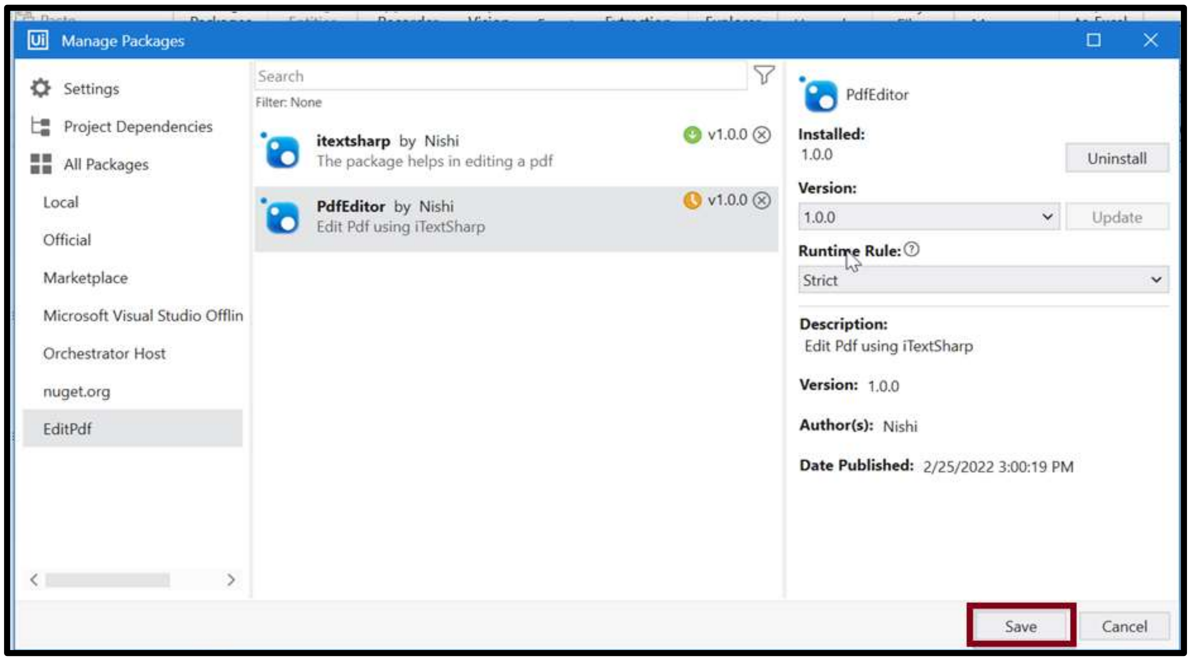
Description:
Edit Pdf using iTextSharp

Version: 1.0.0

Author(s): Nishi

Date Published: 2/25/2022 3:00:19 PM

11.Click on Save.



itextsharp by Nishi v1.0.0
The package helps in editing a pdf

PdfEditor by Nishi v1.0.0
Edit Pdf using iTextSharp

PdfEditor

Installed: 1.0.0

Version: 1.0.0

Runtime Rule: Strict

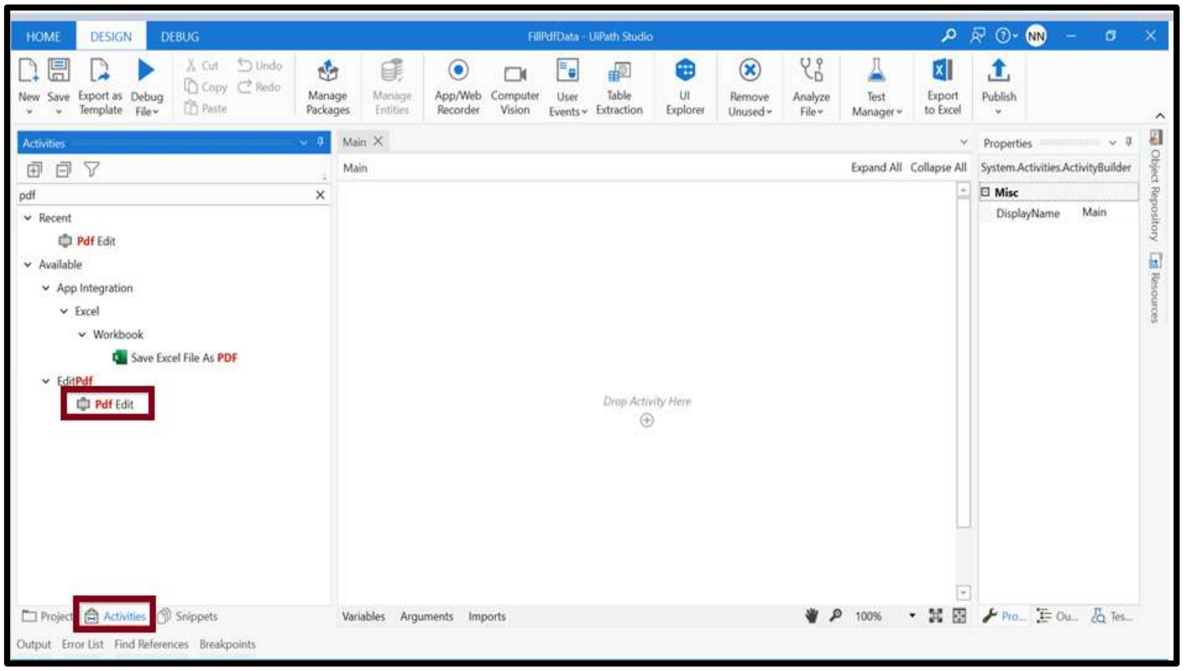
Description:
Edit Pdf using iTextSharp

Version: 1.0.0

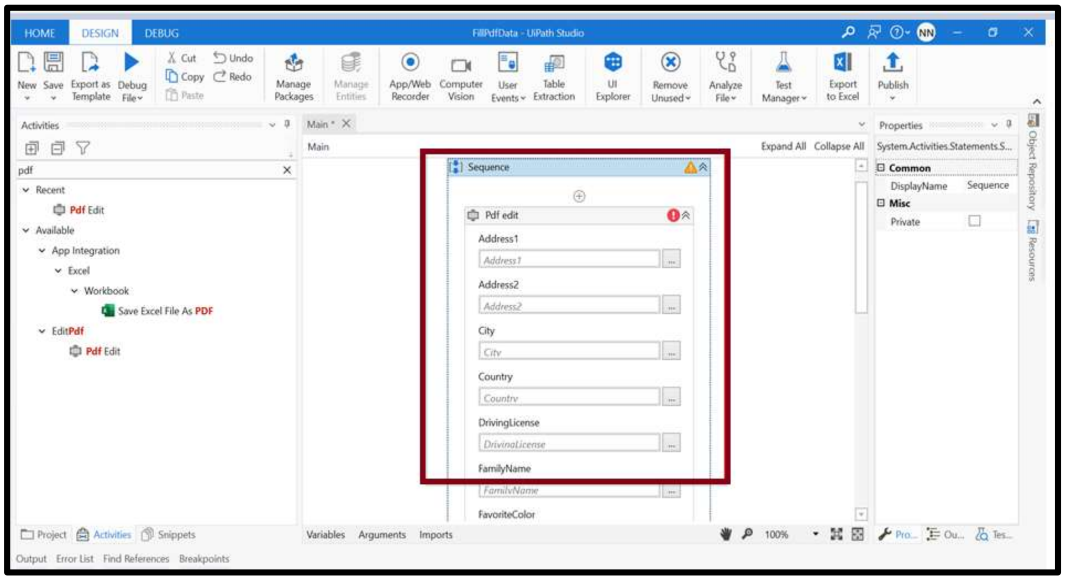
Author(s): Nishi

Date Published: 2/25/2022 3:00:19 PM

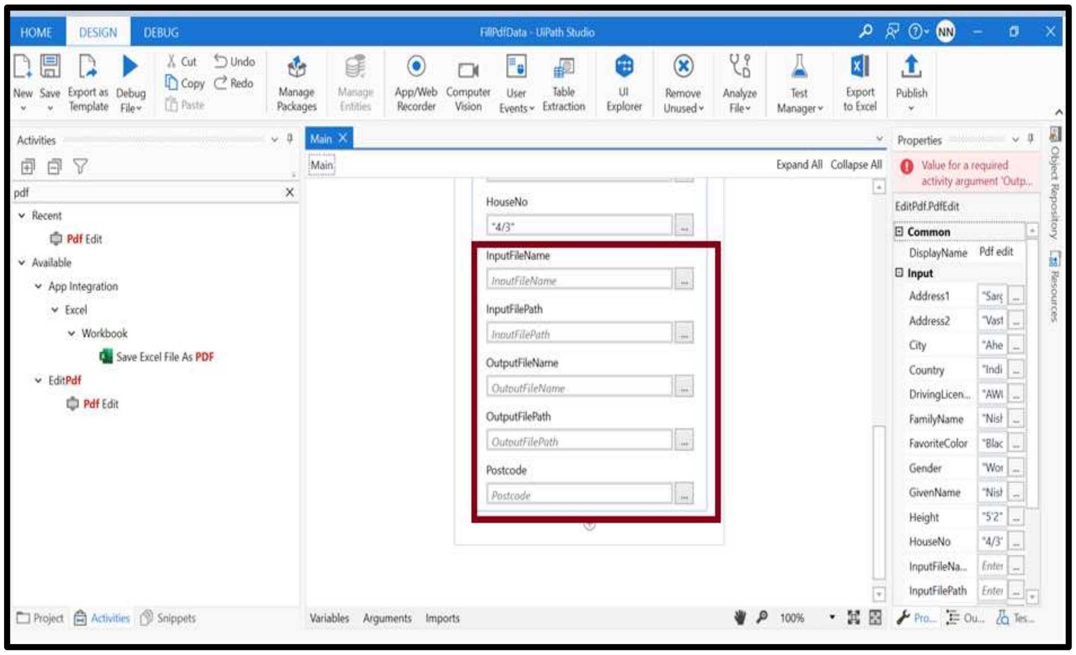
12.Click on the Activities tab in the bottom left corner.



13.Drag and Drop the PdfEdit activity and fill in the values for all the fields.

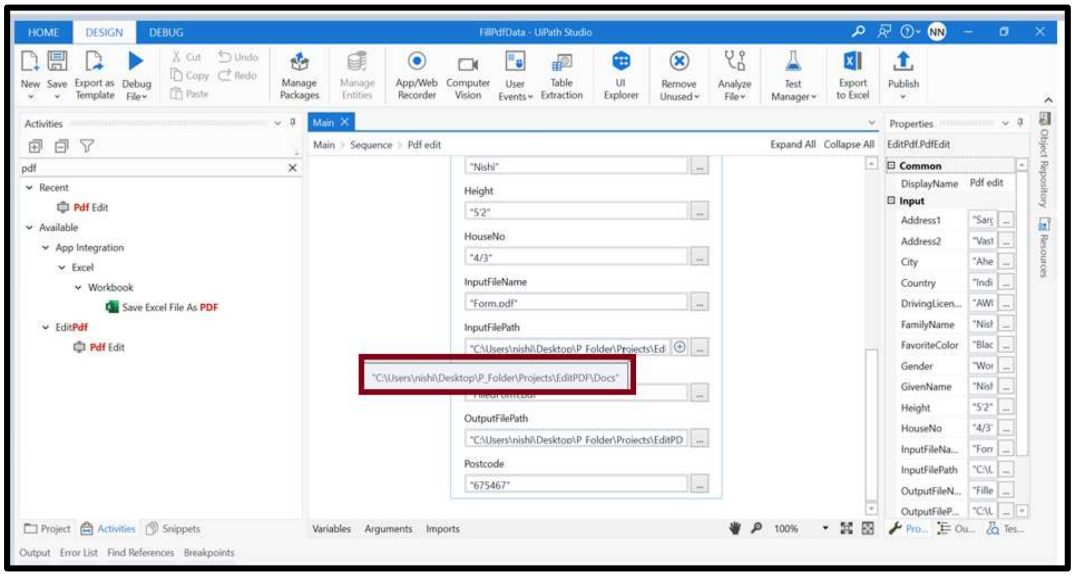


14.Specify input file name, input file path, output file name and output file path. InputFileName is the name of the pdf file which you want to edit. InputFilePath is the folder location where you have saved your pdf file. OutputFileName is the name of the edited pdf file which you will receive as an output after execution of the workflow. OutputFilePath is the location where you want to save your edited pdf.



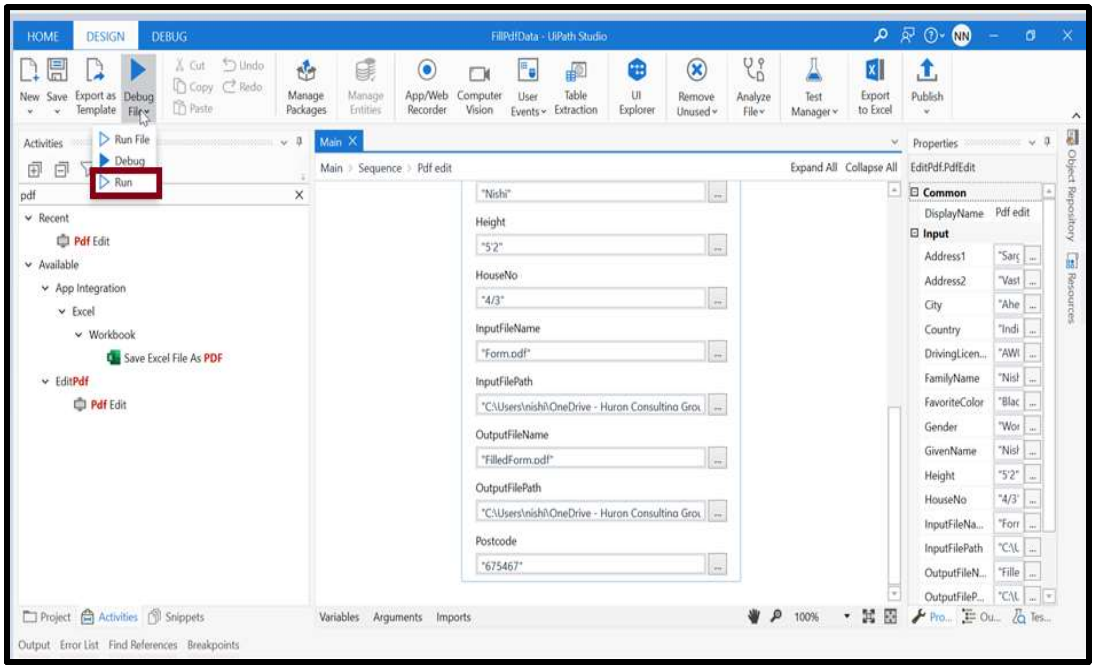
HouseNo
"4/3"
InputFileName
InputFileName
InputFilePath
InputFilePath
OutputFileName
OutputFileName
OutputFilePath
OutputFilePath
Postcode
Postcode

15.Specify File Paths. An example of how you need to specify the file path.



"Nishi"
Height
"5'2"
HouseNo
"4/3"
InputFileName
"Form.pdf"
InputFilePath
"C:\Users\nishi\Desktop\P Folder\Projects\Ed
"C:\Users\nishi\Desktop\P_Folder\Projects\EditPDF\Docs"
OutputFilePath
"C:\Users\nishi\Desktop\P Folder\Projects\EditPD
Postcode
"675467"

16.Click on Debug File and then on Run.



pdf

"Nishi"

Height
"5'2"

HouseNo
"4/3"

InputFileName
"Form.pdf"

InputFilePath
"C:\Users\nishi\OneDrive - Huron Consulting Grou

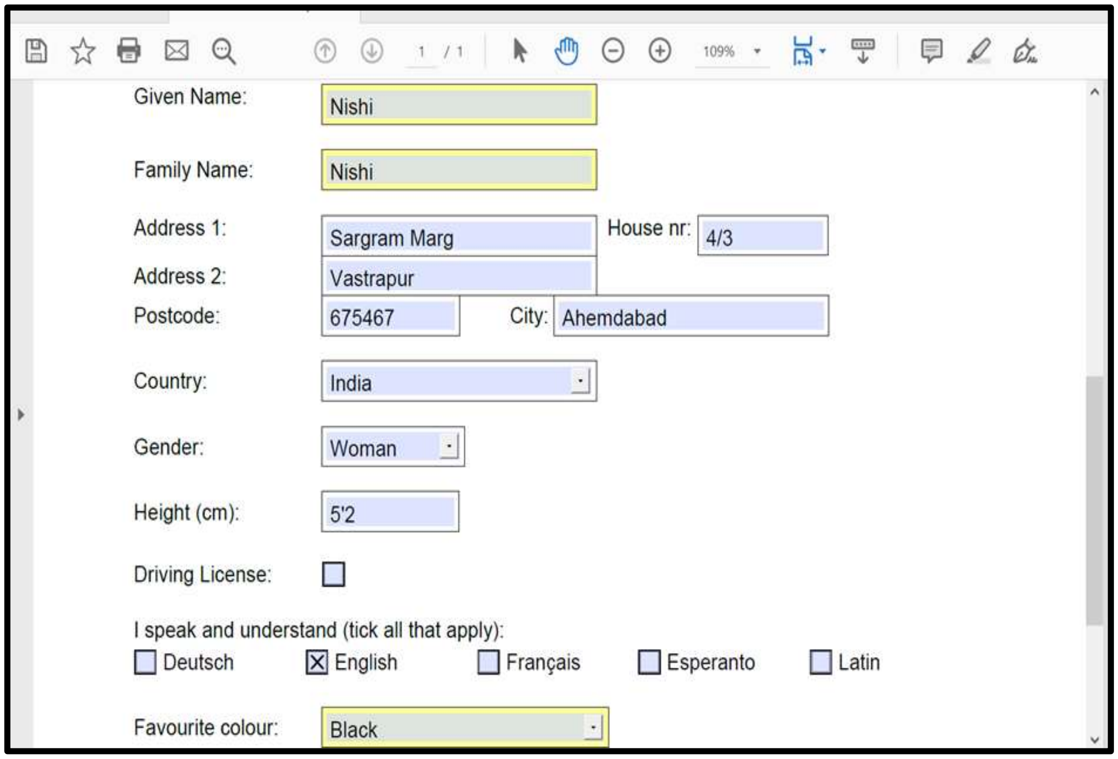
OutputFileName
"FilledForm.pdf"

OutputFilePath
"C:\Users\nishi\OneDrive - Huron Consulting Grou

Postcode
"675467"

DisplayName	Pdf edit
Address1	"Sarç
Address2	"Vast
City	"Ahe
Country	"Indi
DrivingLicen...	"AW
FamilyName	"Nish
FavoriteColor	"Blac
Gender	"Wor
GivenName	"Nish
Height	"5'2"
HouseNo	"4/3"
InputFileNa...	"Forr
InputFilePath	"C:\U
OutputFileN...	"File
OutputFileP...	"C:\U

The final outcome - In the pdf file, the bot has filled in the data against it's label.



Given Name: Nishi

Family Name: Nishi

Address 1: Sargram Marg House nr: 4/3

Address 2: Vastrapur

Postcode: 675467 City: Ahemdabad

Country: India

Gender: Woman

Height (cm): 5'2

Driving License: ☐

I speak and understand (tick all that apply):
☐ Deutsch ☒ English ☐ Français ☐ Esperanto ☐ Latin

Favourite colour: Black