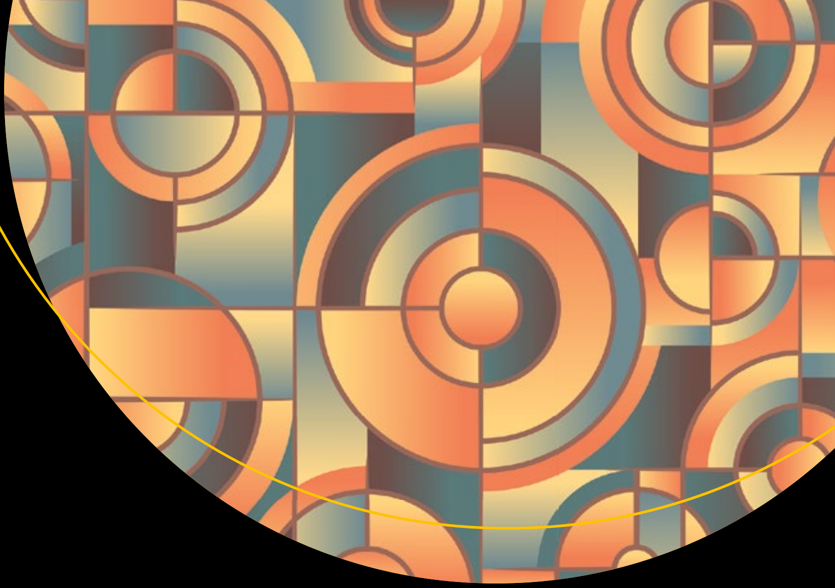


Getting Started with Visual Studio 2019

Learning and Implementing New Features

Dirk Strauss

Apress®

Getting Started with Visual Studio 2019

Learning and Implementing New Features

Dirk Strauss

Apress®

Getting Started with Visual Studio 2019: Learning and Implementing New Features

Dirk Strauss
Uitenhage, South Africa

ISBN-13 (pbk): 978-1-4842-5448-6 ISBN-13 (electronic): 978-1-4842-5449-3
https://doi.org/10.1007/978-1-4842-5449-3

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Smriti Srivastava
Development Editor: Laura Berendson
Coordinating Editor: Shrikant Vishwakarma

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit http://www.apress.com/rights-permissions.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-5448-6. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

To Adele, Tristan, and Irénéé.
My everything for you, always!

Table of Contents

About the Author

Dirk Strauss is a software developer from South Africa with over 13 years of programming experience. He has extensive experience in SYSPRO Customization, with C# and web development being his main focus. He studied at the Nelson Mandela University where he wrote software part-time to gain a better understanding of the technology. He remains passionate about writing code and imparting what he learns with others.

About the Technical Reviewer

James McCaffrey works for Microsoft Research in Redmond, Wash. James has a PhD in cognitive psychology and computational statistics from the University of Southern California, a BA in psychology, a BA in applied mathematics, and an MS in computer science. James worked on several key products including Azure and Bing. James is also the Senior Technical Editor for Microsoft *MSDN Magazine*, the most widely read technical journal in the world.

Acknowledgments

First off, I would like to thank my wife and children for their support while writing this book. I would not have been able to do it without you on my side.

I would also like to thank the team at Apress for their support of this book and for turning my vision into reality. It is a topic that I have wanted to write about for a very long time.

I want to thank James McCaffrey for his help and dedication during the review of this book. Technically reviewing a book such as this is not easy, and his feedback and suggestions are always appreciated and highly valued.

Last, but not least, I want to thank you for reading this book. Your passion to know more is what drives me to learn more, and impart what I learn. It's a symbiotic relationship that benefits us as we both grow and become better at what we do.

Introduction

Visual Studio 2019 is the next version of the stellar development tool we love to use. This book is for folks ready to get to know the IDE a little bit better. It aims to get you started on the road to exploring Visual Studio 2019, beyond what you are already comfortable with.

The book starts off with installing Visual Studio and adding workloads. Then you explore the IDE a bit more before having a look at the existing (and some new) features in Visual Studio. After that, a few productivity tips are thrown in for good measure.

Being able to effectively work with different project types and knowing when to use which are explored in a bit more detail in Chapter 2. We will also have a look at using NuGet packages and how to manage them. We then see how to make use of project templates and then explore using and creating code snippets. This chapter covers many of the basics that are essential to working with Visual Studio and include using bookmarks, code shortcuts, the Server Explorer, and other Visual Studio Windows.

In Chapter 3, we will take a closer look at debugging techniques such as using breakpoints, setting conditional breakpoints, breakpoint actions, and labels. We will see how to effectively use data tips as well as the DebuggerDisplay attribute. We then take a closer look at diagnostic tools and the Immediate Window. Finally, to close off the debugging chapter, we see how to attach to a running process and how to use remote debugging.

The next chapter will introduce you to creating and running unit tests. We will also see how to create live unit tests, how to use IntelliTest to generate unit tests, and how to measure code coverage in Visual Studio.

Finally, we look at working with Git and GitHub. We see how to create a GitHub account and what creating and cloning a repository involves.

You will learn how to commit changes in code to the repository and how to create a branch of your code when you need to work on a new feature in isolation. Then we will look at creating a pull request and how these pull requests are handled. Lastly, we have a look at the benefit of working with stashes.

If you need a nice reference book that deals exclusively with (and only with) Visual Studio, then have a look at what this book has to offer you. If you spend any time using Visual Studio or want to learn how working with Visual Studio 2019 can increase your productivity, then this book will make a perfect reference book for your office.

CHAPTER 1

Getting to Know Visual Studio 2019

Visual Studio is an amazing bit of software. If you have been using Visual Studio for a number of years, you will certainly agree that the IDE offers developers a host of tools and features to make them more productive. You will also be aware that it has grown a lot during the past couple of years and is an absolute powerhouse when it comes to providing tools to develop world-class software.

Initially released as Visual Studio 97 in February 1997, this was the first attempt at using a single development environment for multiple languages. The evolution of Visual Studio is detailed in Table 1-1.

Table 1-1. *The evolution of Visual Studio*

Release	Version	.NET Framework	Release Date
Visual Studio 2019	16.0	3.5–4.8	April 2, 2019
Visual Studio 2017	15.0	3.5–4.7	March 7, 2017
Visual Studio 2015	14.0	2.0–4.6	July 20, 2015
Visual Studio 2013	12.0	2.0–4.5.2	October 17, 2013
Visual Studio 2012	11.0	2.0–4.5.2	September 12, 2012
Visual Studio 2010	10.0	2.0–4.0	April 12, 2010

(*continued*)

D. Strauss, *Getting Started with Visual Studio 2019*,
https://doi.org/10.1007/978-1-4842-5449-3_1

***Table 1-1.** (continued)*

Release	Version	.NET Framework	Release Date
Visual Studio 2008	9.0	2.0, 3.0, 3.5	November 19, 2007
Visual Studio 2005	8.0	2.0, 3.0	November 7, 2005
Visual Studio .NET 2003	7.1	1.1	April 24, 2003
Visual Studio .NET 2002	7.0	1.0	February 13, 2002
Visual Studio 6.0	6.0	N/A	June 1998
Visual Studio 97	5.0	N/A	February 1997

There is so much to see and learn when it comes to Visual Studio. Therefore, in this chapter, we will start by having a look at the following:

- Installing Visual Studio
- What workloads are
- Exploring the IDE (integrated development environment)
- Existing and new features available in Visual Studio 2019
- Productivity tips

If you are using a macOS or a Windows machine, Visual Studio will happily run on both. Let us see where to find the Visual Studio Installer and get going.

Installing Visual Studio

At the time of this writing, Visual Studio 2019 is available for Windows machines as well as for macOS machines. You can download Visual Studio 2019 for Windows from https://visualstudio.microsoft.com/vs/, and if you are on macOS, you will need to head on over to https://visualstudio.microsoft.com/vs/mac/ to download the installer.

Clicking the Download Visual Studio button, you will see a list drop-down with the options as displayed in Figure 1-1.

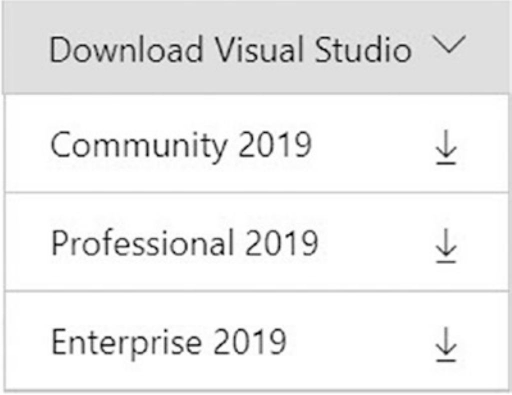


Figure 1-1. *Versions of Visual Studio available*

If you would like to compare the Visual Studio 2019 versions, you can have a look at `https://visualstudio.microsoft.com/vs/compare/` for a detailed comparison. The bottom line is that if you want Visual Studio 2019 for free, download Visual Studio Community 2019.

Visual Studio Community 2019 is aimed at students, open source, and individual developers. The paid tiers include Visual Studio Professional 2019 which is aimed toward small teams and Visual Studio Enterprise 2019 aimed at large development teams.

Microsoft specifies enterprise organizations as those having more than 250 PCs or more than $1 million US dollars in annual revenue.

Let us have a brief look at the recommended system requirements for installing Visual Studio on your machine. For a comprehensive list, browse to `https://docs.microsoft.com/en-us/visualstudio/releases/2019/system-requirements` and have a read through that.

Visual Studio 2019 System Requirements

The system requirements for installing Visual Studio 2019 might differ from those of previous versions of Visual Studio. Refer to the documentation on https://docs.microsoft.com to review the system requirements for previous versions of Visual Studio.

Visual Studio Enterprise 2019, Visual Studio Professional 2019, Visual Studio Community 2019, and Visual Studio Team Foundation Server Office Integration 2019 all support the following minimum system requirements.

Operating Systems

The following Windows operating systems (64-bit recommended) are supported:

- Windows 10 version 1703 or higher
- Windows Server 2019 – Standard and Datacenter
- Windows Server 2016 – Standard and Datacenter
- Windows 8.1 with update 2919355
- Windows Server 2012 R2 with update 2919355
- Windows 7 SP1 with latest Windows updates

Hardware

There is obviously a line here that developers generally don't like to cross when it comes to the minimum hardware specs. Many developers I know will geek out on system RAM and favor SSDs over HDDs. Nevertheless, here are the minimum recommended requirements:

- 1.8 GHz or faster processor (quad-core or better recommended).
- 2 GB of RAM (8 GB of RAM recommended).

- 2.5 GB of RAM minimum if running on a Virtual Machine.
- Between 800 MB and 210 GB of available hard disk space (depending on installed features, 20–50 GB of free space is typically required).
- For improved performance, install Windows and Visual Studio on an SSD.
- Minimum display resolution of 720p (1280x720) but works best at WXGA (1366x768) or higher.

Supported Languages

Visual Studio and the Visual Studio Installer is available in 14 languages as follows:

- English
- Chinese (Simplified)
- Chinese (Traditional)
- Czech
- French
- German
- Italian
- Japanese
- Korean
- Polish
- Portuguese (Brazil)
- Russian

- Spanish
- Turkish

Additional Notes

There are several additional requirements to take note of that I will briefly list here. There are however other requirements that might be of importance to your unique development environment. For a full list, refer to the system requirements at the following link: https://docs.microsoft.com/en-us/visualstudio/releases/2019/system-requirements

- Administrator rights are required to install Visual Studio.
- .NET Framework 4.5 is required to run the Visual Studio Installer and install Visual Studio.
- Visual Studio requires .NET Framework 4.7.2 and is installed during setup.

Using Workloads

After Visual Studio has been installed, you can customize the installation by selecting feature sets, also known as workloads. Think of workloads as a collection of individual features that belong together. This allows you to easily modify Visual Studio to include only what you need.

To launch the workloads screen, find the Visual Studio Installer as can be seen in Figure 1-2.

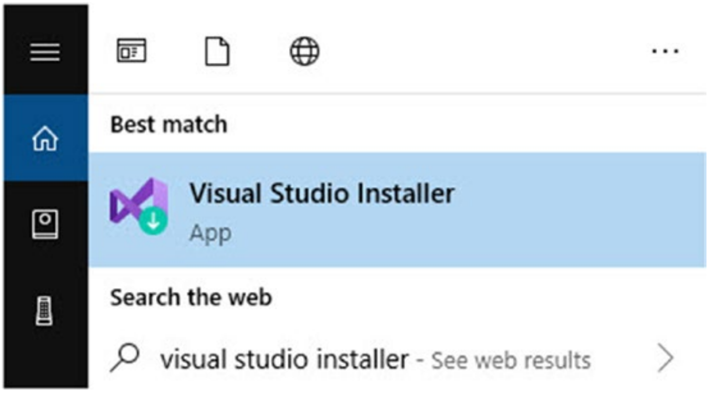
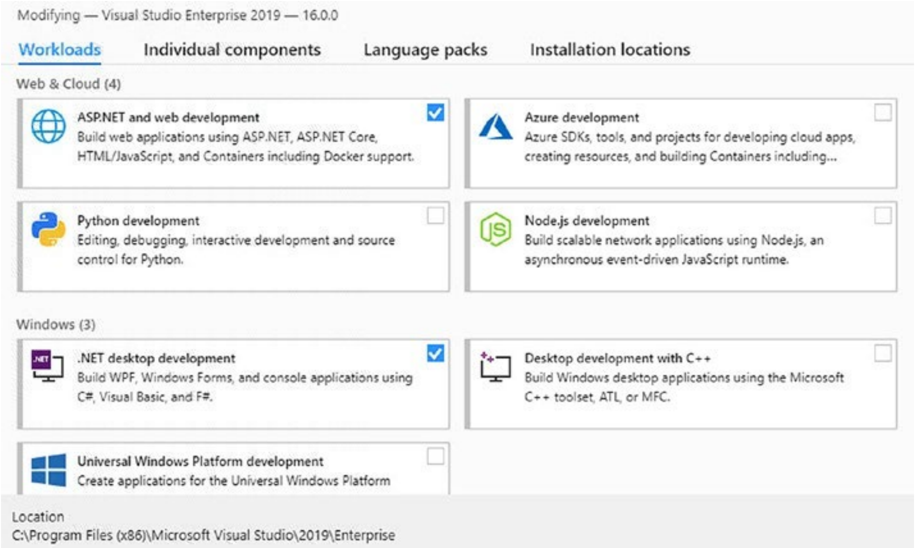


Figure 1-2. *Find the Visual Studio Installer*

Clicking the Visual Studio Installer will launch the installer from where you can modify your installation of Visual Studio as seen in Figure 1-3.


Modifying — Visual Studio Enterprise 2019 — 16.0.0

Workloads | Individual components | Language packs | Installation locations

Web & Cloud (4)

☑ ASP.NET and web development
Build web applications using ASP.NET, ASP.NET Core, HTML/JavaScript, and Containers including Docker support.

□ Azure development
Azure SDKs, tools, and projects for developing cloud apps, creating resources, and building Containers including...

□ Python development
Editing, debugging, interactive development and source control for Python.

□ Node.js development
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Windows (3)

☑ .NET desktop development
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F#.

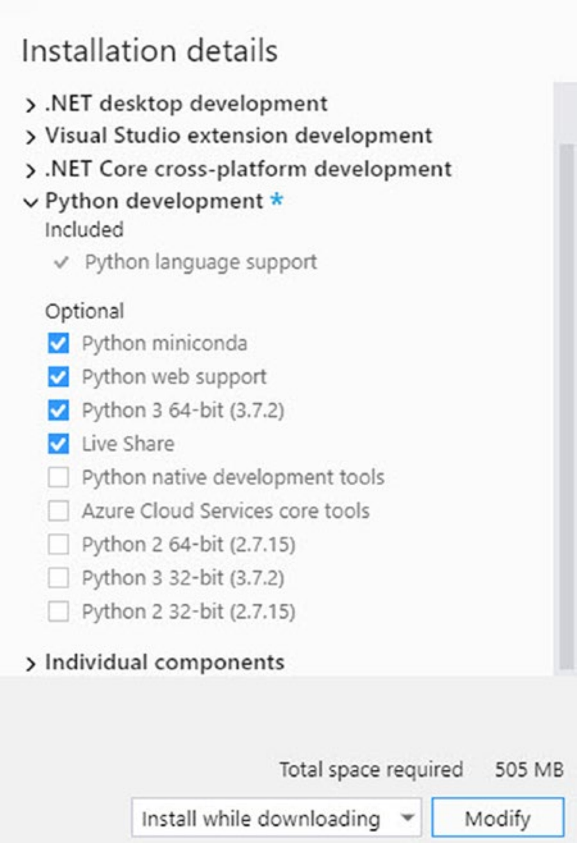
□ Desktop development with C++
Build Windows desktop applications using the Microsoft C++ toolset, ATL, or MFC.

□ Universal Windows Platform development
Create applications for the Universal Windows Platform

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Enterprise

Figure 1-3. *Installing additional workloads*

If you want to start doing Python development, you can simply check the *Python development* workload and install that. As can be seen in Figure 1-4, this will update the installation details section and show you exactly what is being installed and how much additional space you will need to install the selected workload.



Installation details

> .NET desktop development
> Visual Studio extension development
> .NET Core cross-platform development
˅ Python development *

Included

✓ Python language support

Optional

☑ Python miniconda
☑ Python web support
☑ Python 3 64-bit (3.7.2)
☑ Live Share
□ Python native development tools
□ Azure Cloud Services core tools
□ Python 2 64-bit (2.7.15)
□ Python 3 32-bit (3.7.2)
□ Python 2 32-bit (2.7.15)

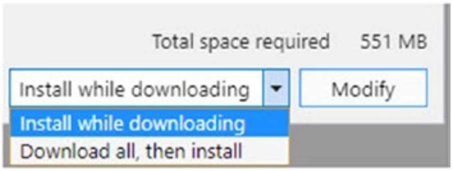
> Individual components

Total space required 505 MB

Figure 1-4. *Workload installation details*

The workloads also contain tabs, namely, Workloads, Individual components, Language packs, and Installation locations. If you needed to install an additional component such as Service Fabric Tools, you can do so by selecting the component on the Individual components tab.

When you have checked all the workloads and individual components you would like to install, you can choose to do the installation while downloading or to download everything before installing as can be seen in Figure 1-5.


Total space required 551 MB

Install while downloading
Modify
Install while downloading
Download all, then install

Figure 1-5. *Installation options*

This will modify your existing installation of Visual Studio 2019 and apply the changes you selected.

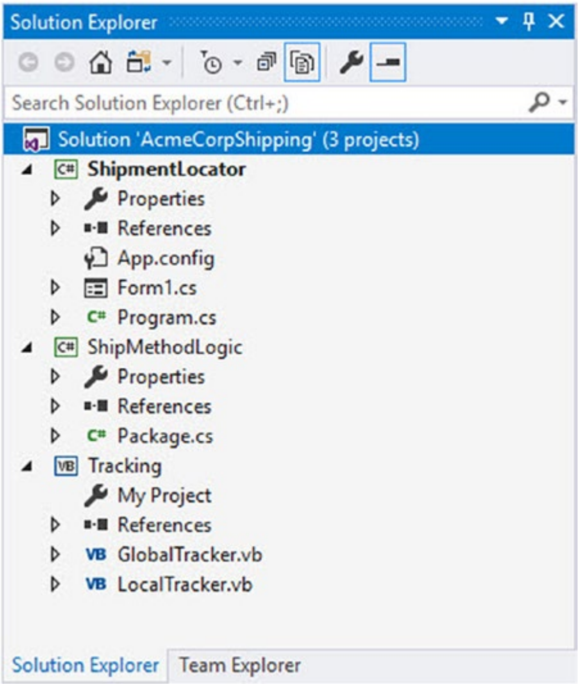
Exploring the IDE

The Visual Studio IDE is full of features and tools that help developers do what they need to do, efficiently and productively. Developers start off creating one or more projects that contain the logic for their code. These projects are contained in what we call a solution. Let's have a look at the Solution Explorer first.

The Solution Explorer

In Visual Studio, the notion of solutions and projects is used. A solution contains one or more projects. Each project contains code that runs the logic you need in order for your application to do what it does.

Consider the example of a Shipment Locator application as can be seen in Figure 1-6.



```
Solution 'AcmeCorpShipping' (3 projects)
  ShipmentLocator
    Properties
    References
    App.config
    Form1.cs
    Program.cs
  ShipMethodLogic
    Properties
    References
    Package.cs
  Tracking
    My Project
    References
    GlobalTracker.vb
    LocalTracker.vb
```

Figure 1-6. *The Shipment Locator Solution*

It is to this solution that you will add all the required projects in order to create your application. From the example in Figure 1-6, we can see that the solution contains three projects. The projects are as follows:

- ShipmentLocator – WinForms application – C#
- ShipMethodLogic – Class Library – C#
- Tracking – Class Library – VB.NET

Of particular interest, you will notice that you can have a solution that contains a mix of C# projects and VB.NET projects. You are therefore not limited by a particular language and can create applications containing a mix of .NET languages.

The reason that we can mix .NET languages in the same solution is due to something we call IL (Intermediate Language). IL is used by the .NET Framework to create machine-independent code from the source code used in your projects.

The WinForms application will contain the UI needed to track and trace shipments. In order for the WinForms application to be able to use the logic contained in the other two class libraries, we need to add what is called a reference to the other projects.

This is done by right-clicking the project that you want to add the reference to and selecting *Add Reference* from the context menu (Figure 1-7).

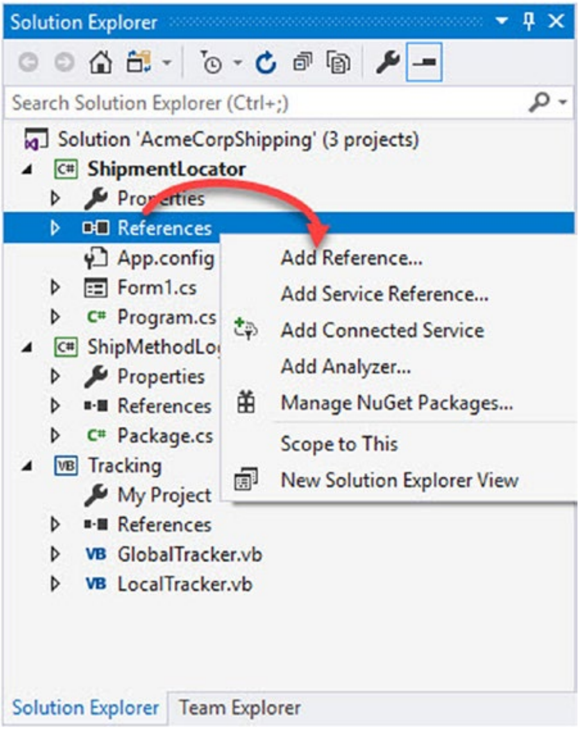
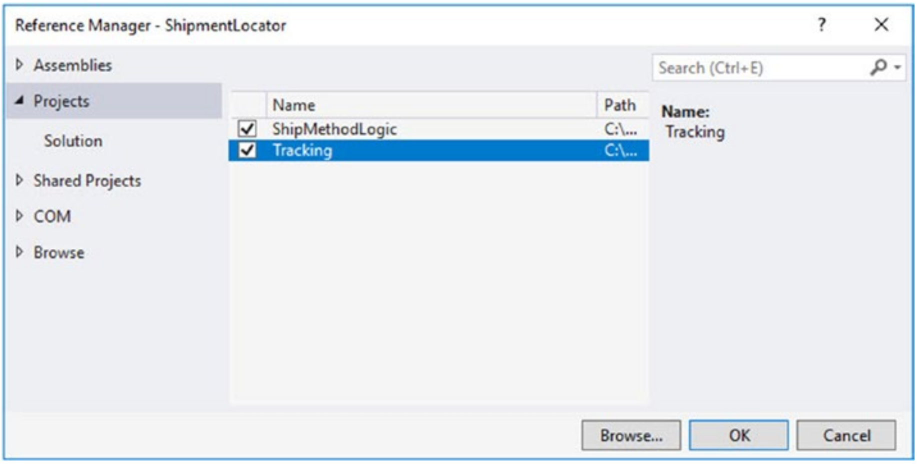


***Figure 1-7.** Adding a project reference*

When you click the Add Reference menu item, you will be presented with the Reference Manager screen as seen in Figure 1-8.



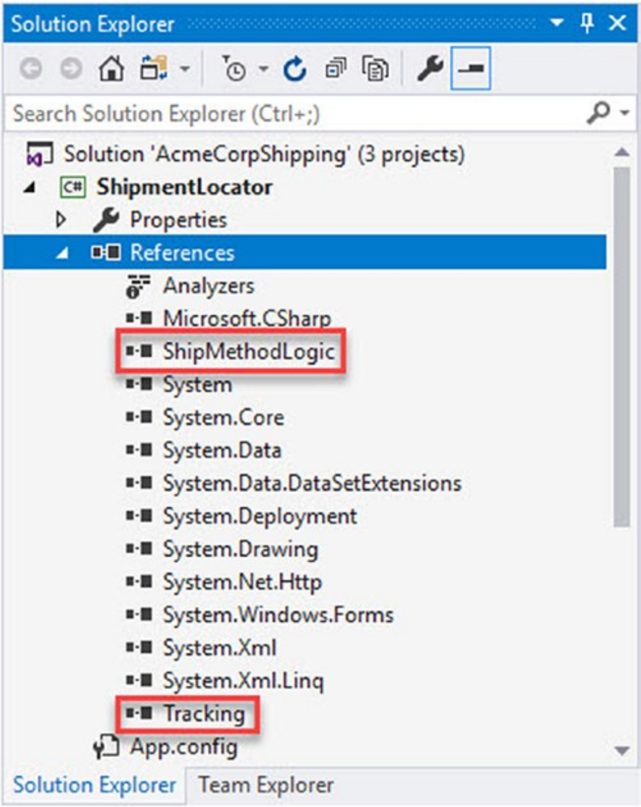
	Name	Path
☑	ShipMethodLogic	C:\...
☑	Tracking	C:\...

Name: Tracking

Figure 1-8. *The Reference Manager screen*

Under the Projects tab, you will find the other two Class Library projects in your solution. By checking each one and clicking the OK button, you will add a reference to the code in these projects.

If you had to expand the References section under the ShipmentLocator project, you will see that there are two references to our Class Library projects ShipMethodLogic and Tracking as can be seen in Figure 1-9.



Solution Explorer

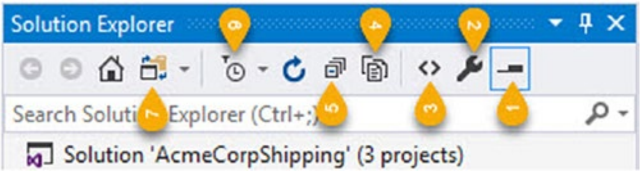
Search Solution Explorer (Ctrl+;)

- Solution 'AcmeCorpShipping' (3 projects)
  - ShipmentLocator
    - Properties
    - References
      - Analyzers
      - Microsoft.CSharp
      - ShipMethodLogic
      - System
      - System.Core
      - System.Data
      - System.Data.DataSetExtensions
      - System.Deployment
      - System.Drawing
      - System.Net.Http
      - System.Windows.Forms
      - System.Xml
      - System.Xml.Linq
      - Tracking
    - App.config

Solution Explorer | Team Explorer

Figure 1-9. *Added references*

This will now make all the code you write in the ShipMethodLogic and Tracking projects, available to the ShipmentLocator project. Having a look at the toolbar on the Solution Explorer (Figure 1-10), you will notice that it contains several buttons.



Solution Explorer

Search Solution Explorer (Ctrl+;)

Solution 'AcmeCorpShipping' (3 projects)

Figure 1-10. *The Solution Explorer toolbar*

The buttons contained here are displayed as needed. The View Code button, for example, will only show up in the toolbar when a file containing code is selected in the Solution Explorer. These buttons give you quick access to the following features as outlined in the following:

1. Preview Selected Items
2. Properties
3. View Code
4. Show All Files
5. Collapse All
6. Pending Changes Filter
7. Toggle between Solution and Folder views

I will not go through each one in detail, but of particular interest, you will notice that the Show All Files will display unnecessary files and folders such as the bin folder in your Solution Explorer. Go ahead and click the Show All Files button, and look at the Solution Explorer again.

By looking at Figure 1-11, you can see that it now displays the bin folder and the obj folder. These folders are not necessary for your code but are important to your solution.

The obj folder contains bits of files that will be combined to produce the final executable. The bin folder contains the binary files that are the executable code for the application you are writing.

Each obj and bin folder will contain a Debug and Release folder that simply matches the currently selected build configuration of your project.

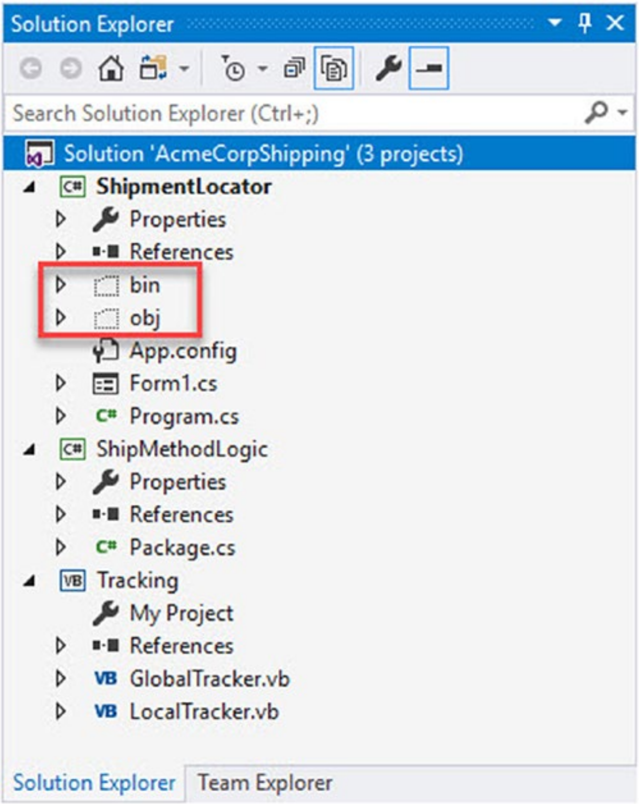


Figure 1-11. *Solution Explorer displaying all files*

You can now right-click the bin folder as seen in Figure 1-12 and click the Open Folder in File Explorer menu to quickly have a look at the contents of the folder.