



Community Experience Distilled

# Python Tools for Visual Studio

Leverage the power of the Visual Studio IDE to develop better  
and more efficient Python projects

Martino Sabia  
Cathy Wang

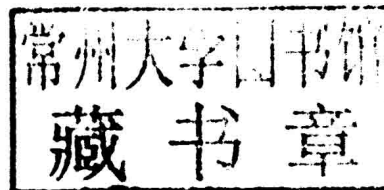
**[PACKT]** open source\*  
PUBLISHING community experience distilled

# Python Tools for Visual Studio

Leverage the power of the Visual Studio IDE to develop better and more efficient Python projects

**Martino Sabia**

**Cathy Wang**



**[PACKT]** open source\*  
PUBLISHING community experience distilled  
BIRMINGHAM - MUMBAI

# Python Tools for Visual Studio

Copyright © 2014 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: April 2014

Production Reference: 1140414

Published by Packt Publishing Ltd.  
Livery Place  
35 Livery Street  
Birmingham B3 2PB, UK.

ISBN 978-1-78328-868-7

[www.packtpub.com](http://www.packtpub.com)

Cover Image by Cathy Wang ([ms.cathywang@gmail.com](mailto:ms.cathywang@gmail.com))

# Credits

**Authors**

Martino Sabia

Cathy Wang

**Reviewers**

Steve Dower

Fabio Lonegro

Chris Marinic

**Commissioning Editor**

Anthony Albuquerque

**Acquisition Editor**

Harsha Bharwani

**Content Development Editor**

Sriram Neelakantan

**Technical Editor**

Shashank Desai

**Copy Editors**

Roshni Banerjee

Gladson Monteiro

**Project Coordinator**

Melita Lobo

**Proofreader**

Paul Hindle

**Indexers**

Monica Ajmera Mehta

Priya Subramani

**Production Coordinator**

Conidon Miranda

**Cover Work**

Conidon Miranda

# About the Authors

**Martino Sabia** is a curious-minded developer with close to 30 years of coding experience. Throughout his years of working with different platforms and languages, he has always kept his mind fresh while finding creative ways of using different technologies. Based in Italy, Martino has spent his career in various start-up companies, working in numerous roles from junior developer to software architect. Now he is the Project Lead for Deltatre; he works on consumer-facing, heavy-traffic websites and media-streaming platforms in the sports industry.

**Cathy Wang** is an experienced designer who specializes in service design and experience strategy. She has worked on many cross-channel projects and served as a design lead for enterprise services around the globe in fields ranging from Telecom to public sectors. Cathy has worked for world-class design agencies to help bring visions to life. In her free time, she builds web projects and apps. She is infinitely curious about new technologies and the experiences they can bring.

# About the Reviewers

**Steve Dower** works at Microsoft and is a developer of Python Tools for the Visual Studio team.

**Fabio Lonegro** has spent many years doing research in theoretical physics (String and Gauge theory) and collaborating with many divulgating projects, including the translation of Peter Woit's book *Not Even Wrong*. He was always passionate about web development and has spent the last 15 years working on web projects related to e-learning and data visualization. He is now a developer at Deltatre spa, where his work is focused on many fields, from the integration of complex data with multimedia streams for both mobile and desktop experiences to custom solutions for web content indexing and the development of Node.js. Currently, he uses Python for a variety of applications that involve data which comes from Arduino and Raspberry Pi shields. He is also a capoeira teacher, a passionate cyclist, and above all, a caring father.

**Chris Marinic** is an autodidact with decades of engineering experience. Growing up, he excelled at computer science, often mentoring his fellow students. He designed, developed, launched, and sold his own start-up while working full-time as the Director of Engineering at Sabre Hospitality Solutions.

# www.PacktPub.com

## Support files, eBooks, discount offers, and more

You might want to visit [www.PacktPub.com](http://www.PacktPub.com) for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at [www.PacktPub.com](http://www.PacktPub.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [service@packtpub.com](mailto:service@packtpub.com) for more details.

At [www.PacktPub.com](http://www.PacktPub.com), you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

## Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

## Free access for Packt account holders

If you have an account with Packt at [www.PacktPub.com](http://www.PacktPub.com), you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

# Table of Contents

<b>Preface</b>	<b>1</b>
<b>Chapter 1: Introduction to PTVS</b>	<b>7</b>
Step-by-step installation and configuration	7
PTVS tools overview	12
The Python Environments window	13
Python Interactive	14
Visual Studio panels with PTVS	14
Summary	16
<b>Chapter 2: Python Tools in Visual Studio</b>	<b>17</b>
Mastering IntelliSense with Python	17
Using REPL in Visual Studio	21
Navigating code with ease	24
Object Browser	28
Summary	31
<b>Chapter 3: Day-to-day Coding Tools</b>	<b>33</b>
Project handling	33
Solution	33
Project	34
Specifying Python environments	37
Defining Search Paths	41
Refactoring	42
Debugging	46
Using breakpoints	47
Utilizing watch entries	48
Summary	49



<b>Chapter 4: Django in PTVS</b>	<b>51</b>
<b>Django project template and tools</b>	<b>52</b>
Installing a Python package	53
Running the application	55
IntelliSense in Django templates	57
<b>Setting up and managing a database for a Django project</b>	<b>58</b>
<b>Setting up the admin interface</b>	<b>61</b>
<b>Creating a new Django application</b>	<b>63</b>
<b>Deploying a Django project on Microsoft Azure</b>	<b>65</b>
<b>Summary</b>	<b>71</b>
<b>Chapter 5: Advanced Django in PTVS</b>	<b>73</b>
<b>Library management</b>	<b>73</b>
<b>The Fabric library – the deployment and development task manager</b>	<b>75</b>
<b>South – the database deployment library</b>	<b>79</b>
Why use South with Django	80
Installing South	80
Schema migration with South	83
<b>Summary</b>	<b>87</b>
<b>Chapter 6: IPython and IronPython in PTVS</b>	<b>89</b>
<b>IPython in PTVS</b>	<b>89</b>
<b>IronPython</b>	<b>95</b>
Using .NET classes in Python code with IronPython	95
Using the Python code in .NET with IronPython	100
<b>Summary</b>	<b>105</b>
<b>Index</b>	<b>107</b>

# Preface

Like many other developers, Python developers have always had to find ways to manage the development workflow between different tools. Most of the time, this happens without using a comprehensive guide that is available in a complete IDE which is specifically designed for Python development.

The rare, exceptional IDEs that offer complete guides are often expensive and don't provide hands-on steps to help speed up the development process.

Visual Studio, as a matured and well-developed tool over the last few decades, has dominated the market of compiled languages and languages that are strictly oriented toward Windows and .NET. Packed with handy tools and functionalities to speed up and facilitate the workflow of developers, it helps users to render repetitive tasks, manage projects, and provide a detailed outlook into the structure of a project. However, most importantly, it helps users gain a clear view into the inner structure of the code.

In the last few years, Microsoft has started exploring how to integrate new languages into Visual Studio; as a result, Python Tools for Visual Studio (PTVS) was developed. It's a well-developed tool that is already on its second release and is commonly used by professional developers as their new IDE of choice for Python projects.

PTVS has everything that a Python developer can dream of: consistent project files management, interactive debugging and code completion features with the rock solid Microsoft IntelliSense technology, project templates, a first-class Django integration package, virtual environment management right in the IDE for REPL, and a native code-based IDE that loads and reacts fast.

This book will focus more on the integration of Python in Visual Studio than the language itself. It will try to delve into the power offered by the tool and venture into the feasibility of its day-to-day usage for a developer. We will show real examples of how to use PTVS with Django and how to deal with occasional difficulties when it comes to integrating well-known libraries into a Python project on Microsoft Windows.

## What this book covers

*Chapter 1, Introduction to PTVS*, provides a high-level overview of PTVS and the interaction between Visual Studio and a Python interpreter.

*Chapter 2, Python Tools in Visual Studio*, provides an in-depth analysis of the tools, type checking, inner functionalities, and automatisms (IntelliSense and REPL) of PTVS.

*Chapter 3, Day-to-day Coding Tools*, talks about browsing through the code and the flexible setting of Python environments. It also talks about refactoring and the debugging process.

*Chapter 4, Django in PTVS*, shows how to harness the powerful Visual Studio IDE and tooling to speed up Django development.

*Chapter 5, Advanced Django in PTVS*, provides an in-depth look at remote task management and schema migrations using the third-party Python libraries Fabric and South.

*Chapter 6, IPython and IronPython*, provides an overview of the IPython library and how it's integrated in Visual Studio. It also provides an introduction to IronPython and its integration with the .NET framework.

## What you need for this book

You will need a basic understanding of Python, a computer with Windows installed, and an Internet connection. To follow through the exercises and examples, we would suggest that you have Visual Studio as well.

## Who this book is for

This book is intended for developers who are aiming to enhance their productivity in Python projects with automation tools that Visual Studio provides for the .NET community. Some basic knowledge of Python programming is essential.

# Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "We can include other contexts through the use of the `include` directive."

A block of code is set as follows:

```
class foo:
    """
    Documentation of the class.
    It can be multiline and contain any amount of text
    """
    @classmethod
    def bar(self, first=0, second=0):
        """This is the documentation for the method"""
        return first + second

print(foo.bar())
```

Any command-line input or output is written as follows:

```
python manage.py schemamigration south2ptvs --initial
```

New terms and important words are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "Clicking on the **Next** button moves you to the next screen."



Warnings or important notes appear in a box like this.



Tips and tricks appear like this.

## Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book — what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to [feedback@packtpub.com](mailto:feedback@packtpub.com), and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on [www.packtpub.com/authors](http://www.packtpub.com/authors).

## Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

## Downloading the color images of this book

We also provide you a PDF file that has color images of the screenshots/diagrams used in this book. The color images will help you better understand the changes in the output. You can download this file from: [https://www.packtpub.com/sites/default/files/downloads/86870S\\_ColoredImages.pdf](https://www.packtpub.com/sites/default/files/downloads/86870S_ColoredImages.pdf)

## Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books — maybe a mistake in the text or the code — we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

## Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at [copyright@packtpub.com](mailto:copyright@packtpub.com) with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

## Questions

You can contact us at [questions@packtpub.com](mailto:questions@packtpub.com) if you are having a problem with any aspect of the book, and we will do our best to address it.

## Special thanks from the authors

Thanks to Packt Publishing for giving us the opportunity to publish this book for the developer community, and the help they have provided during the entire process: from the injection of the idea to the whole process of giving birth to it. It has been a journey filled with surprises and discoveries.

We'd also like to appreciate our reviewers, Fabio Lonegro and Chris Marinic, who have provided us with clear and unbiased feedback along the way, giving us great insights on untangling the details of the book.

Last but not least, we would like to thank the Microsoft PTVS team, specifically Steve Dower, who has contributed to the book personally and through providing technical support on every detail. Thanks to Shahrokh Mortazavi for reaching out to us through a tweet (<https://twitter.com/cathycracks/status/421336498748006400>). Steve and the rest of the team have given us lots of help, insights, and suggestions on how to overcome some complex but very important parts of the book. They even invited us to visit them in person to gain a greater insight into their work. We truly feel that PTVS is developed by a group of passionate people who care for the community and are eager to develop PTVS to be an even better and useful tool. The Microsoft PTVS team has done a great job with the tool so far in our opinion, and we look forward to what's yet to come.

We have enjoyed this journey so far, and we are very happy to be doing this together to bring this book to life. It has been an intimate and difficult process filled with love and with some very deep and long discussions into late nights. We hope that you enjoy and gain knowledge from this book as much as we have learned from it.

We hope that you will find this book interesting and that it will help you discover the inner power of PTVS, as Scott Hanselman described PTVS in a post on his blog, *One of Microsoft's Best-Kept Secrets - Python Tools for Visual Studio (PTVS)*, created on July 2, 2013 and found at <http://www.hanselman.com/blog/OneOfMicrosoftsBestKeptSecretsPythonToolsForVisualStudioPTVS.aspx>.

# 1

## Introduction to PTVS

Python Tools in Visual Studio (PTVS) is an extremely powerful tool because of the following reasons:

- It gives Python developers a powerful IDE with many helpful coding features and integrations in one unique environment.
- PTVS provides developers on the Windows platform the opportunity to use their favorite IDE – Visual Studio – to explore, learn, and manage one of the most commonly used scripting languages.

In this chapter, we will have a high-level overview of PTVS, starting with a step-by-step tutorial for installing and configuring it correctly followed by a quick overview of the principle tools of Visual Studio to control the Python environment and configuration. Understanding the Visual Studio windows will greatly benefit your ability to explore and manage workflows of the source code and the structure of your Python project.

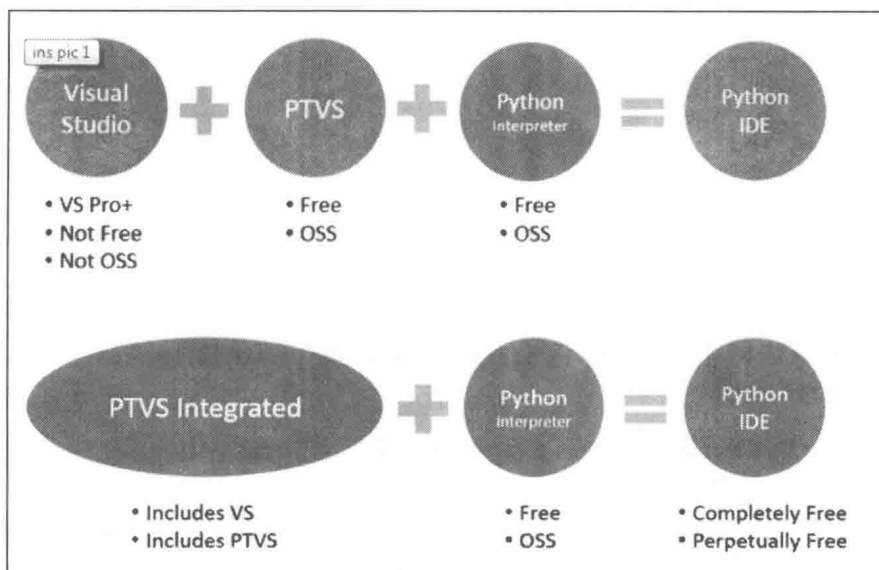
## Step-by-step installation and configuration

There are various formats of PTVS available for installation depending on your preexisting installed version of Visual Studio. PTVS is available for Visual Studio 2010, 2012, and 2013 (Pro edition or above).

If the previously mentioned versions of Visual Studio are not installed on your computer, it's possible to install a standalone version of PTVS. Visual Studio permits side-by-side installation, meaning it provides the ability to install multiple versions on one system. The only prerequisite is that the older version must be installed before the newer one.



The different types of installations possible for PTVS are described on its **CodePlex website**, <http://go.microsoft.com/fwlink/?LinkID=390659>.



The preceding figure is taken from <http://go.microsoft.com/fwlink/?LinkID=390659>.

The most important prerequisite for Visual Studio 2013 is to have Windows 7 (32 or 64 bit) or above running as your operating system.

Once you have sorted out the prerequisites and installed the PTVS package of your choice, you will need to decide on the type of Python interpreter. Choosing the appropriate Python interpreter depends on your need for your project. Refer to the PTVS CodePlex page at <http://go.microsoft.com/fwlink/?LinkID=299429> to help your decision-making process. You can choose between CPython and IronPython (32 or 64 bit). If you chose CPython, then you can choose between Python Version 2.7 and 3.3. It is recommended to use CPython 3.3 32 bit for most cases. For web development, the recommendation would be CPython 2.7 32 bit.