

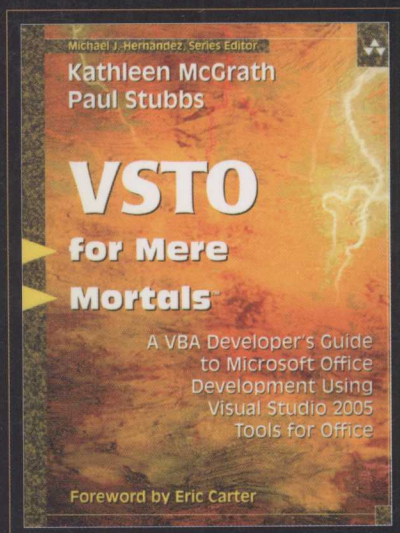
VSTO

开发者指南

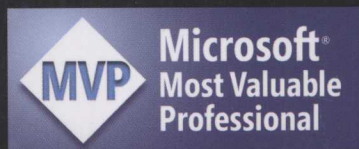
VSTO for Mere Mortals:

A VBA Developer's Guide

to Microsoft Office Development Using Visual Studio 2005
Tools for Office

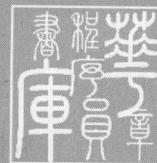


Kathleen McGrath Paul Stubbs 著
李永伦 译



机械工业出版社
China Machine Press

华章程序员书库



VSTO

开发者指南

VSTO for Mere Mortals:

A VBA Developer's Guide

to Microsoft Office Development Using Visual Studio 2005
Tools for Office

Kathleen McGrath Paul Stubbs 著
李永伦 译



机械工业出版社
China Machine Press

本书是使用 VSTO 开发 Microsoft Office 的 VBA 开发者指南。本书重点讲解了托管代码和 Visual Studio 集成开发环境 (IDE), 如何使用视图控件、数据绑定和操作窗格为 Word 2003 和 Excel 2003 创建文档级别的定制程序, 如何为 Outlook 2003 开发插件, 用 VSTO 和 VSTO 2005 SE 创建的解决方案在安全和部署方面的有用信息, 如何把 VBA 解决方案迁移到 Visual Basic 2005 和 VSTO, 以及更多关于为 Microsoft Office 2007 系统开发 VSTO 2005 SE 插件定制功能区、自定义任务窗格和 Outlook 窗体区域的详细信息。另外, 本书通过丰富的代码示例来展示 VSTO 功能, 使读者更易于理解和掌握。

本书适合 VSTO 开发人员、VBA 开发人员、.NET 开发人员以及其他开发人员参考。

Simplified Chinese edition copyright © 2008 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *VSTO for Mere Mortals: A VBA Developer's Guide to Microsoft Office Development Using Visual Studio 2005 Tools for Office* (ISBN: 0-321-42671-1) by Kathleen McGrath, Paul Stubbs, Copyright © 2007.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley.

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

版权所有, 侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2008-1755

图书在版编目 (CIP) 数据

VSTO 开发者指南/麦克格拉斯 (McGrath, K.) 等著; 李永伦译. —北京: 机械工业出版社, 2009. 1

(华章程序员书库)

书名原文: *VSTO for Mere Mortals: A VBA Developer's Guide to Microsoft Office Development Using Visual Studio 2005 Tools for Office*

ISBN 978-7-111-25060-9

I. V… II. ①麦… ②李… III. 程序语言 - 程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 136136 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 周茂辉

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2009 年 1 月第 1 版第 1 次印刷

186mm × 240mm · 26 印张

标准书号: ISBN 978-7-111-25060-9

定价: 59.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换
本社购书热线: (010) 68326294

对本书的赞誉

“真希望第一次接触 VSTO 和 .NET Framework 时就拥有本书。它对于那些考虑 VSTO 以及从 Office VBA 转到 Visual Basic .NET 的人都是极具价值的。广泛的主题为 MSDN 和其他方面的更加深入的面向开发者的文档提供了一个切入点。”

——Cindy Meister, MS Word MVP

“本书做了一件很好的事，在 VBA 和 VSTO 两个世界之间架起了一座桥。Kathleen 和 Paul 告诉我们，VBA 开发者如何在拥有丰富的 Office 知识的同时也能获得 Visual Studio 的强大功能，如可维护性和安全性。他们成功地写了一本容易接受、易于理解和令人信服的书。”

——KD Hallman, 总经理, Microsoft, Developer Division

“对于想要使用 Visual Studio Tools for Office 为 Office 进行 .NET 开发的 VBA 开发者，本书是一个不错的选择。很多示例都展示了 VB .NET 代码和对应的 VBA 代码，帮助解释关键概念和用法的区别。”

——Steve Hansen, OfficeZealot.com,
《Mastering Excel 2003 Programming with VBA》的作者
《Mastering Excel 2000 Premium Edition》的作者之一

“Visual Studio 2005 Tools for Office 是构建托管 Office 应用程序的途径。如果你正在从 VBA 开发转到托管代码开发，本书会告诉你如何做。”

——Ken Getz, 资深顾问, MCW Technologies, LLC

“对于有兴趣使用 Visual Studio 和 .NET 的强大功能进行 Office 编程的 VBA 开发者，强烈推荐本书。”

——Mei Liang, 测试软件设计工程师, Microsoft

“Kathleen McGrath 在本书里给出了很多有一定深度的代码示例，以一种易于理解的方式演示 VSTO。本书的内容包括了从简单的如何实现到高级功能。我从本书中学到了很多。”

——Justin Whitney, 技术记者

“本书为想熟悉强大的新 Office 开发环境的 VBA 和 Office 开发者提供了实用的资源，更甚于此，它还提供了许多‘值得购买本书’的提示、功能完整的示例，以及来自那些明显花费很多时间使用这些工具帮助初学者走向专家的人士的优雅方法。”

——Rory Plaire, 解决方案架构师, DigitalCommons LLC

“对于有兴趣用 .NET 编程的 Office VBA 开发者或者打算转向 Office 平台的 .NET 程序员，本书是物有所值的。如果你正在寻找一个完整的、编写完好的资料，迅速上手，以加快用 Visual Studio 2005 开发和编程 Office 解决方案，我相信本书是必须的。”

——Frank Rice, 技术作家, Microsoft Office Developer Center

“有了本书，VBA 开发者便获得了访问有价值资源的途径，这些资源帮助他们理解托管代码的必要概念和把 VBA 解决方案迁移到 Visual Basic 2005 和 Visual Studio 2005 Tools for Office 的最佳实践。本书除了介绍 VSTO 之外，还介绍了新的 VSTO 2005 SE 和 2007 Microsoft Office 系统。”

——Erika Ehrli Cabral, 网站管理员,
MSDN Office Developer Center, Microsoft

“Kathleen 和 Paul 在 VSTO 刚开始的时候就分别以不同的方式加入进来了。他们知识的深度和广度都在这本详细而权威的书里有所反映。我将本书强烈推荐给任何要从 VBA 转到托管代码的专业开发者。”

——Eric Lippert, 软件设计工程师, Microsoft

译者序

还记得第一次父亲带我去看电脑时，商家首先给我们演示的是电脑的办公功能，包括字处理、电子表格和演示文稿，其实只是分别打开安装在 Windows 3.2 上的 Word、Excel 和 PowerPoint。这是我第一次接触 Microsoft Office，也正是这次接触激发了我对 Microsoft Office 的好奇心，回去之后我买了一本关于 Word for Windows 的书，这也是我的第一本计算机书籍。

这么多年来，Microsoft Office 的功能爆炸式地增加，但增加的都是普通用户的通用功能，你不会看到为你的问题量身定做的解决方案。比如说，在我翻译这本书的过程中，要无数次插入这类内容：

程序清单 2-1 引发异常的代码

```
Private Sub Sheet1_BeforeDoubleClick(ByVal Target As _  
    Microsoft.Office.Interop.Excel.Range, _  
    ByRef Cancel As Boolean) Handles Me.BeforeDoubleClick  
  
    Dim myString As Object = Nothing  
    MsgBox(myString.ToString)  
    myString = "This is a test"  
    MsgBox(myString.ToString)  
  
End Sub
```

我有必要手工输入每一个字符吗？很明显，没有！如果你不在 Word 上进行二次开发，恐怕你的手就要受罪了。对于这个简单的例子，VBA 开发者可能已经开始在大脑里编码了，但如果提出这个需求的是一个 .NET 开发者，那么这口饭就不那么容易吃了。自从某天邂逅了 VSTO，在用 Microsoft Office 时，我变得比以前积极了，遇到问题（像上面提到的那个）时，我也更主动地去思考解决方案，而不是走出阳台无奈地望着天。通过 VSTO，我可以把现有的 .NET 技能都用到 Office 开发上，发挥想象力，把创意融入 Office，比如说，为自己量身定做一套翻译辅助工具，你可以在里面包含术语管理，甚至可以尝试实现辅助重构。

本书本身是为 VBA 开发者写的，作者在书里和你分享他们从 VBA 转到 VSTO 时的心得，帮助 VBA 开发者更平滑地转到 VSTO。如果你恰好是一个 VBA 开发者并且有兴趣加入 .NET 这个大家庭，那么本书就是你的不二之选了。如果你是一个 .NET 开发者，你也不会感到失望，作者在书里以独特的视角和你分享他们对 VSTO 的理解，而这些都和你平时所接触到的知识形成互补之势。

最后，希望本书能够成为你的好帮手。

李永伦

2008 年于上海

推 荐 序

如果你熟悉 VBA 并且考虑将要踏入使用 Visual Basic 2005 和 Visual Studio Tools for Office (VSTO) 的 Office 开发世界, 不要再等了, 这就是你要找的书。

Kathleen 和 Paul 在本书里用独特的方法为 VBA 开发者讲述 VSTO。从 VBA 转到 Visual Basic 2005 和 VSTO 时, 有很多新的东西需要学习, 让我们面对吧。Visual Basic 语言的改变比你想要的多, 而且也去掉了一些熟悉的原有的语言构造。幸运的是, 本书描述了当你从 VBA 开发转到 Visual Basic 2005 开发时将会遇到的主要改变。VSTO 的开发环境也和 VBA 的开发环境很不同。再有, 本书给你指出 VSTO 的新功能并帮助你理解如何把你辛苦得到的 VBA 技能迁移到基于 Visual Studio 的新开发环境。

最后, 从 VBA 转到 VSTO 是否值得? 我想你会发现这是值得的。VSTO 提供了更丰富的环境, 还有很多强大的功能可以在你的 Office 开发项目里使用, 最终使你事半功倍。这难道不是你想要的吗?

Kathleen 和 Paul 是引领你进入 VSTO 新世界的最好向导。Kathleen 在 VSTO 开发期间是一个技术作家, 换句话说, 她在我们开发 VSTO 的时候就开始写这方面的东西了。我发现写作是验证设计的最好方法, 事实也是如此。当 Kathleen 为 VSTO 写作时, 她会走进我的办公室, 指出很多令人迷惑或者错误的东西。最终, 她使 VSTO 变得更好。她从一个独特的角度去理解 VSTO 的功能并写下这方面的东西。

我还要提及 Kathleen 收集整理的关于 VSTO 的非常棒的录像。如果你仍在考虑是否在自己的项目里试一下 VSTO, 请看一些她的录像, 以及她作为 VSTO 组博客的贡献者为社区做的其他贡献:

<http://blogs.msdn.com/vsto2/>

在这本书里, 你不仅得到 Kathleen 的智慧, 也得到 Paul 的智慧。Paul 是 VSTO 的早期用户之一, 他通过参与 VSTO 产品早期的软件设计评审协助指导 Microsoft。因为他的帮助很大, 所以 VSTO 组雇佣了他。即使在他加入 Microsoft 之后, 他仍继续发挥 VSTO“第一批用户”这个角色的作用。如果你搜索论坛和新闻组, 就会发现 Paul 经常回复开发者贴出的 VSTO 难题。

正如你所期望的, Paul 和 Kathleen 都有很好的博客。你可以去看看:

<http://blogs.msdn.com/kathleen/>

<http://blogs.msdn.com/pstubbs/>

作为本书的早期审阅者, 我喜欢这种让 VBA 开发者感到易懂和实用的方式。你不会在本书里找到任何 C# 代码, 这很好。Visual Basic 是你可以选择的用于 Office 开发的最佳语言, 因为 Visual Basic 伴随着 Office 一起成长, 它的功能让 Office 对象模型变得更加易用。另外, 我发现本书有很多很好的提示和技巧, 让你获得最佳的 VSTO 体验, 比如说, Kathleen 和 Paul 告诉你如何设置你的环境, 使调试更加容易。

最后, 这是第一本介绍 VSTO 2005 SE 的书, 你现在可以为更多 Office 应用程序开发, 包括 Office 2007。另外, 我还想透露一下, 我们已经在开发用于 Office 2007 的下一代 VSTO, 我们会继续努力让 VSTO 变得简单和强大。VSTO 会延续下去, 而本书则会帮助 VBA 开发者实现转换。

Eric Carter

Microsoft 公司

VSTO 开发经理

序 言

Visual Studio Tools for Office (VSTO) 的目标受众是“专业开发者”。这个术语有多重含义，而我们听到的最受认可的定义是“通过写代码获取报酬的人”。换句话说，这是他的主业。他不是部门开发者，比如说，把写 Excel 宏作为会计任务的一部分的会计人员，或者定制 Word 来增加生产力的办公人员。相反，他是一个有兴趣把 Microsoft Office 作为开发平台的 .NET 开发者。

我们相信传统的 Office 开发者也会对 VSTO 感兴趣。在加入 Microsoft 之前，我们俩都是 VBA 开发者，定制 Office 应用程序，并且我们非常有兴趣了解托管代码。在这方面，我们不认为我们是唯一的。有上百万 VBA 开发者，许多人都有兴趣学习下一代 Office 开发。现在的 VSTO 图书和文档通常都不是为 VBA 开发者写的，它假设开发者熟悉 Visual Studio、面向对象编程和 .NET Framework。这就不难理解为什么它把重点更多地放在 VSTO 的功能和如何使用笨重的 Office 对象模型。

我们想为 VBA 开发者写一本书。虽然你可能不熟悉 .NET 编程，但你有一个重要的优势：Office 对象模型的知识。作为一个 Office 开发者，你很可能非常熟悉 Office 应用程序，在操作 Office 对象模型方面也有丰富的经验。我们认为学习托管代码的最佳环境是你已经熟悉的 Office 开发环境。

VSTO 把 Office 开发带到 .NET 的世界，和 VBA 相比，它既有优点也有缺点。使用 VSTO，你可以更好地定制 Word、Excel 和 Outlook，例如，创建自定义任务窗格、向文档添加智能标记和把文档上的对象绑定到数据源。使用 VSTO 2005 SE，你可以为 6 个 Office 应用程序创建插件、定制新的 2007 Microsoft Office 系统的功能区和创建应用程序级别的自定义任务窗格。

我们有幸和为 VSTO 进行设计、编码、测试和写文档的人一起工作，并从他们那里学到了很多。我们得到了内部人士对 VSTO 的看法，我们希望以一种易懂和有趣的方式把这些信息传递给你。

——Kathleen McGrath 和 Paul Stubbs

前言

世界上有上百万 Microsoft Office 开发者。大多数创建 Visual Basic for Application (VBA) 解决方案的人都不会把他们自己看作 Office 开发者，但他们是。事实上，如果你使用 VBA 定制 Office 文档，那么你就在使用其中一个最受欢迎的开发平台了。

你可能已经开始了开发生涯，在 Word 或者 Excel 里录制宏来自动化重复的任务。不久之后，你开始调整你的宏，让它们按照你所需的方式工作。你或许有过这样的经历，由于准时为老板创建了重要的文档，成为了办公室的英雄。如果你和很多人一样，那么你已经深入到 VBA 编码了。你现在从零开始创建 VBA 项目，代码的行数也激增。你甚至可能已经发现 VBA 的一些局限性，从而妨碍你创建真正想要的解决方案。

你或许已经听说 Microsoft 在向 VBA 语言添加新功能上并不积极。这并不是说 Microsoft 将会停止支持 VBA，而是说不会增加新的功能，例如 Web 服务、XML 和其他诸如代码段的功能。但是，如果你使用 Visual Studio 2005 Tools for Office (VSTO) 和 Visual Basic，你就可以利用这些新功能。

你可能已经开始转到使用 VSTO 开发 Office 并且想了解更多内容。或许你现在只是想试一下 VSTO，然后看看使用托管代码能为解决方案带来哪些增强。

简而言之，使用 VSTO 和 Visual Basic 2005 能使工作效率到达一个新的层次，而无论你在到达这个新的层次的路上走了多远，本书都会帮助你达到那个层次。

本书面向的读者群

我们是为 VBA 开发者写本书的。“VBA 开发者”是一个很宽泛的术语，它包括任何录制过简单的宏的人和创建企业级解决方案的专业开发者。本书是为那些人而写的，从只懂很少 VBA 编程的初级 VBA 开发者到已经使用 Visual Basic 6.0 或者 Visual Basic 2005 的高端开发者。

这是一本关于从 VBA 转到 Visual Basic 2005 的书。你会学到如何用 Visual Basic 2005 创建很好的 Office 解决方案，Visual Basic 2005 将是你认为可以使用的最好的 Office 开发语言。

本书的目的

我们俩有使用 VBA 创建 Office 解决方案的经验，我们认为有必要帮助其他人转向使用 Visual Studio 2005 和 VSTO 进行托管编程。和 VSTO 团队一起工作使我们对 VSTO 和 Office 开发有了更深的理解。我们和很多 VBA 开发者谈过，他们的问题是如何把他们用 VBA 写的代码翻译成 VSTO 和 Visual Basic 2005。多年来，我们开始看到 VBA 开发者所碰到的问题的模式。我们决定写这本书来满足这些需要。

阅读本书的方法

本书要按章节顺序阅读。每章都是基于前一章的概念的。这些概念的复杂度也随每章增长。

一开始，我们向你示范一些基本的东西，例如创建一个项目并运行它。如果你对 VSTO 或者 Visual Basic 2005 不熟悉，我们推荐你从头开始并按你的方式读下去。如果你熟悉 Office 和 .NET

编程，你可以直接跳到感兴趣的章节。

我们还在附录里包含了高级示例，在你熟悉 VSTO 编程之后可以使用。附录 A 示范了如何创建 XML 代码段。这些代码段允许你向解决方案插入代码，使创建应用程序变得更快更容易。附录 B 示范了一个为 Outlook Inspector 创建工具栏的方法。

本书的组织结构

这里是每章的简要概述。

第一部分：VSTO 概述

第 1 章介绍了 VSTO，示范了如何用 Visual Studio 集成开发环境 (IDE) 创建解决方案，并介绍了一些你将会在其他章节深入学习的 VSTO 功能。

第 2 章示范了如何使用 IDE 的一些功能，包括如何生成、运行和调试解决方案。本章也向你示范了如何使用集成帮助系统更容易地找到使用 VSTO 编程的相关信息。

第 3 章讲述 .NET Framework 的基础知识，包括讨论面向对象编程和 Visual Basic 2005 的新功能，也将会探讨公共语言运行库的功能和 Office 主互操作程序集 (PIA)。

第 4 章讨论了从 VBA 转到 Visual Basic 的原因，并展示了 VBA 里没有的 Visual Basic 强大的新功能。你也将会看到如何从 VBA 向 Visual Basic 2005 迁移关键的语言功能，例如从 UserForms 转到 Windows Forms。

第二部分：Word 和 Excel

第 5 章介绍了 Document Actions 任务窗格。你将会学到如何管理和设计操作窗格。你也将会学到如何使用 XML 和操作窗格创建强大的上下文相关的解决方案。

第 6 章示范了如何创建 Word 项目并解释项目背后的结构并且将介绍如何在文档上使用 Word 控件绑定数据和处理事件。

第 7 章示范了如何创建 Excel 项目并解释项目背后的结构并且将介绍如何在工作表上使用 Excel 控件绑定数据和处理事件。

第 8 章示范了如何向解决方案添加控件，并介绍如何向工具栏和菜单添加控件，如何在文档或者工作簿上使用 Windows Forms 控件，以及如何在运行时创建控件。

第 9 章讲述了如何在 Word 和 Excel 里创建文档级别的 VSTO 智能标记并介绍如何识别术语和表达式以及如何处理智能标记的操作。

第三部分：Outlook 和其他内容

第 10 章解释了如何为 Outlook 创建应用程序级别的定制程序，并将介绍 Outlook 对象模型，如何定制菜单和工具栏，如何保护和调试 Outlook 插件。

第 11 章介绍了 VSTO 的安全模型并示范了如何保护和部署 Word、Excel 和 Outlook 项目。你将会看到如何使用 Publish Wizard 和 Visual Studio Setup 项目进行部署。

第 12 章描述了把 VBA 解决方案迁移到 VSTO 的原因。你将会学到更多转换策略并演练一个转换示例。

第 13 章讨论了如何向文档添加 VSTO 定制程序 (和从文档中移除它)。本章也向你示范了如何访问服务器上的 VSTO 文档里的数据缓存。

第 14 章包含了使用 VSTO 和 Office 2007 的例子。你将会学到如何对 Office 2007 的新功能编程，例如应用程序级别的自定义任务窗格、Office 功能区和 Outlook 窗体区域。

致谢

我要感谢我的丈夫 Mark Shelton，他在整个过程里默默地支持我，审阅了许多章节，在我集中精力展开这个项目的时间内，他显示出无比的耐心。另外，非常感谢我的合著者 Paul Stubbs。感谢 Paul 在技术方面的专业知识、他对完成这个项目所付出的努力和他的幽默感。

——Kathleen McGrath

我要感谢我的家人，在长时间的写作和修订中对我的支持，尤其是我的妻子，她把一切事情都处理很好，让我可以集中精力，尽可能把书写好。我还要感谢 Kathleen 邀请我为这本书写部分章节。和 Kathleen 合作很棒，她对 VSTO 的热情使这本书成为可能。

——Paul Stubbs

我们还要感谢我们的编辑 Joan Murray，让我们有机会写本书，并且指导我们完成这个过程。特别感谢我们的系列编辑 Mike Hernandez，因为他允许这本书成为 Mere Mortals 系列的第一本与产品相关的书。

另外，我们要感谢我们的技术编辑和 Addison-Wesley 的制作和营销团队，尤其是 Kim Boedigheimer、John Fuller、Lara Wysong、Curt Johnson 和 Eric Garulay。感谢所有向我们提供宝贵建议的技术审阅者，包括 Eric Carter、Mei Liang、Justin Whitney、Peter Jausovec、Rory Plaire、Joe Kunk、McLean Schofield、Thomas Quinn 和 Frank Rice。VSTO 产品组创建了如此棒的产品，我们为每个成员感到自豪！

作译者简介

作者简介

Kathleen McGrath 在 Microsoft 做技术作家。她为 Visual Studio 2005 Tools for the Microsoft Office System (VSTO)、Visual Studio Tools for Applications (VSTA) 和 Visual Basic 写过文档。在加入 Microsoft 之前，她是一个 VBA 开发者，为金融印刷及法律行业定制 Word 应用程序。Kathleen 还创建了视频短片，演示 VSTO 和 Visual Basic 的功能，这些录像放在她的博客上。她的博客是 <http://blogs.msdn.com/kathleen>。

Paul Stubbs 在 Microsoft 的 Visual Studio Tools for Office (VSTO) 组做程序经理。除了 VSTO，Paul 还和 VSTA 组合作为 InfoPath 2007 和独立软件供应商 (ISV) 开发新的托管代码应用程序的可编程开发工具。Paul 为《MSDN Magazine》写过文章，也在诸如 TechEd 和 TechReady 的活动上演讲过。Paul 还参与 Microsoft 论坛的开发者社区，他的博客是 <http://blogs.msdn.com/pstubbs>。

译者简介

李永伦

连续 3 次获得 Microsoft MVP 称号，现就职于高知特信息科技有限公司(上海)有限公司。喜欢阅读《新发现》、《经理人》和《心理月刊》，曾参与《C# 和 .NET 实战：平台、语言与框架》一书的翻译。好于博客中长篇大论自己之见解，也醉于阅读他人短小精巧之言论。你可以在 <http://allenlooplee.cnblogs.com> 阅读到他的技术文章，也可以通过 allenlooplee@yahoo.com.cn 与他交流。

目 录

对本书的赞誉

译者序

推荐序

序言

前言

第一部分 VSTO 概述

第 1 章 VSTO 入门	1
1.1 本章主题	1
1.2 什么是 VSTO	1
1.3 为什么用 VSTO 代替 VBA	2
1.4 VSTO 的功能	3
1.4.1 宿主项和宿主控件	3
1.4.2 Windows Forms 控件	5
1.4.3 智能标记	6
1.4.4 定制任务窗格	6
1.4.5 在文档中缓存数据	6
1.4.6 托管插件	6
1.4.7 安全和部署	7
1.4.8 功能区支持	7
1.4.9 增强开发环境	7
1.5 创建 VSTO 解决方案	7
1.5.1 用 VBA 创建 Office 解决方案	7
1.5.2 用 VSTO 创建 Office 解决方案	7
1.5.3 项目类型	8
1.6 VSTO 如何与 Visual Studio 集成	11
1.6.1 把 Word 和 Excel 当作设计器	11
1.6.2 VSTO 中的代码文件	12
1.7 小结	13
1.8 复习问题	14
第 2 章 编程环境	15
2.1 本章主题	15
2.2 Visual Studio IDE 介绍	15
2.2.1 环境设置	15

2.2.2 创建 Visual Studio Tools for Office 项目	16
2.2.3 准予访问 VBA 项目系统	18
2.2.4 菜单和工具栏	18
2.3 查看 IDE 窗口	20
2.3.1 代码视图	20
2.3.2 探究 Solution Explorer	21
2.3.3 Properties 窗口	26
2.3.4 浏览 Object Browser	27
2.3.5 项目属性	28
2.3.6 查看其他 Visual Studio 窗口	29
2.4 编码工具	31
2.4.1 使用智能感知	31
2.4.2 编辑并继续	35
2.5 生成和运行代码	35
2.6 调试你的代码	36
2.6.1 设置断点	38
2.6.2 逐语句、逐过程和跳出代码	40
2.6.3 试用调试工具	41
2.6.4 使用调试窗口	43
2.7 查找和使用帮助	47
2.7.1 目录	47
2.7.2 索引	48
2.7.3 搜索	48
2.7.4 如何实现	49
2.7.5 动态帮助	50
2.8 小结	50
2.9 复习问题	50
第 3 章 托管代码介绍	51
3.1 本章主题	51
3.2 什么是托管代码	51
3.3 面向对象编程介绍	52
3.3.1 理解类和对象	52
3.3.2 分部类	58
3.3.3 泛型类	59
3.3.4 接口	59
3.3.5 面向对象概念	60
3.4 探究 .NET Framework	61

3.4.1 程序集	62	5.3 定制 Document Actions 任务窗格	99
3.4.2 命名空间	62	5.4 管理操作窗格	100
3.4.3 应用程序域	63	5.4.1 管理操作窗格的大小和位置	100
3.4.4 公共语言运行库	63	5.4.2 显示和隐藏操作窗格	101
3.4.5 公共语言规范	64	5.5 设计操作窗格解决方案	102
3.4.6 .NET Framework 类库	65	5.5.1 创建操作窗格解决方案	103
3.5 VSTO 和托管代码	67	5.5.2 添加操作窗格控件	104
3.5.1 主互操作程序集	68	5.5.3 从文档访问操作窗格	105
3.5.2 解决方案程序集	70	5.5.4 从操作窗格访问文档	106
3.5.3 运行 VSTO 解决方案	71	5.5.5 管理控件顺序	108
3.6 小结	71	5.6 创建上下文相关的解决方案	111
3.7 复习问题	71	5.6.1 创建 XML 架构	112
第 4 章 从 VBA 转到 VSTO 和 Visual Basic 2005	72	5.6.2 映射 XML 元素到你的文档	113
4.1 本章主题	72	5.6.3 创建多个操作窗格控件	114
4.2 转到 Visual Studio 2005	72	5.6.4 显示和隐藏操作窗格上的 控件	115
4.3 Visual Basic 2005 的新功能	72	5.6.5 向操作窗格控件添加功能	116
4.3.1 My 对象	73	5.7 小结	118
4.3.2 智能感知代码段	76	5.8 复习问题	118
4.3.3 语句	76	第 6 章 使用 VSTO 定制 Word	119
4.3.4 运算符	77	6.1 本章主题	119
4.4 VBA 和 Visual Basic 2005 在 语言上的差别	78	6.2 对 Word 进行编程	119
4.4.1 数据类型	79	6.2.1 把录制的 VBA 宏转换成 VSTO 里的 Visual Basic	120
4.4.2 方法	82	6.2.2 重写内置命令	122
4.4.3 变量的声明和作用域	82	6.3 Word 宿主项和宿主控件	125
4.4.4 默认属性	83	6.3.1 向文档添加宿主控件	126
4.4.5 枚举	84	6.3.2 从基础的互操作对象访问 宿主控件	131
4.4.6 异常处理	84	6.3.3 从宿主控件访问互操作对象	133
4.5 UserForm 和 Windows Forms	86	6.4 对文档上的宿主控件进行数据 绑定	134
4.5.1 比较 ActiveX 控件和 Windows Forms 控件	87	6.4.1 把数据绑定到控件	134
4.5.2 控件功能的改变	92	6.4.2 在文档里缓存数据	137
4.6 小结	97	6.5 对事件进行编程	138
4.7 复习问题	97	6.5.1 Bookmark 控件的事件	138
第二部分 Word 和 Excel		6.5.2 XMLNode 控件的事件	145
第 5 章 定制 Word 和 Excel 任务 窗格	98	6.5.3 XMLNodes 控件的事件	149
5.1 本章主题	98	6.5.4 Document 的事件	149
5.2 什么是任务窗格	98	6.6 对 Bookmark 的特别强化	154
		6.7 使 Word 文档智能化	156

6.8 小结	157	8.2 关于控件	207
6.9 复习问题	157	8.3 向工具栏和菜单添加控件	207
第7章 使用 VSTO 定制 Excel	158	8.3.1 创建工具栏按钮	207
7.1 本章主题	158	8.3.2 创建菜单和菜单项	213
7.2 对 Excel 进行编程	158	8.4 使用 Windows Forms 控件	216
7.2.1 了解 Excel 对象模型	161	8.5 向 Windows Forms 添加控件	222
7.2.2 把录制的 VBA 宏转换成 VSTO 里的 Visual Basic	162	8.6 向任务窗格添加控件	228
7.2.3 写回调从 VBA 访问 VSTO 代码	163	8.7 向 Excel 和 Word 文档添加控件	230
7.3 Excel 宿主项和宿主控件	165	8.7.1 Word 里的 Windows Forms 控件	230
7.3.1 宿主项	166	8.7.2 Excel 里的 Windows Forms 控件	233
7.3.2 宿主控件	166	8.7.3 添加数据绑定控件	234
7.3.3 向工作表添加宿主控件	166	8.7.4 向文档添加自定义控件	237
7.3.4 移除宿主控件	173	8.7.5 控件类型	238
7.3.5 在运行时添加宿主控件	174	8.7.6 在运行时添加控件	239
7.3.6 控件持久化	177	8.7.7 文档上的 Windows Forms 控件的功能	244
7.3.7 从基础的互操作对象访问宿主控件	178	8.8 小结	244
7.3.8 从宿主控件访问互操作对象	179	8.9 复习问题	245
7.3.9 在 Excel 里添加宿主项	179	第9章 Word 和 Excel 里的智能标记	246
7.3.10 Excel 的区域设置问题	179	9.1 本章主题	246
7.4 对工作表上的宿主控件进行数据绑定	180	9.2 什么是智能标记	246
7.4.1 创建数据源	180	9.3 用 VSTO 创建 Word 和 Excel 智能标记	249
7.4.2 向工作表添加数据绑定控件	182	9.4 理解智能标记的属性	252
7.4.3 使用关联数据	186	9.4.1 Actions 属性	252
7.4.4 在工作簿里缓存数据	187	9.4.2 Caption 属性	252
7.5 对事件进行编程	187	9.4.3 Expressions 属性	252
7.5.1 NamedRange 控件事件	188	9.4.4 SmartTagType 属性	252
7.5.2 ListObject 控件事件	190	9.4.5 Terms 属性	252
7.5.3 XMLMappedRange 控件事件	195	9.5 在智能标记上执行操作	253
7.5.4 Chart 控件事件	195	9.6 正则表达式介绍	258
7.5.5 工作表事件	197	9.6.1 使用正则表达式识别词组	258
7.5.6 图表工作表事件	200	9.6.2 识别文本模式	261
7.5.7 工作簿事件	201	9.7 重写智能标记识别器	264
7.6 使 Excel 更智能	204	9.7.1 创建 Word 项目	265
7.7 小结	206	9.7.2 运行代码	267
7.8 复习问题	206	9.7.3 进入离线状态	269
第8章 Word 和 Excel 里的控件	207	9.8 小结	270
8.1 本章主题	207	9.9 复习问题	270

第三部分 Outlook 和其他内容

第 10 章 使用 VSTO 为 Outlook

创建插件	271
10.1 本章主题	271
10.2 使用 VSTO 的应用程序级别的定制程序	271
10.2.1 共享插件和 VSTO 插件之间的区别	272
10.2.2 Outlook 插件项目模板	274
10.2.3 把 Outlook 宏转换成 VSTO 插件	276
10.2.4 创建一个简单的插件	278
10.2.5 Visual Studio 里的智能感知	280
10.3 Outlook 对象模型概述	281
10.3.1 Application 对象	281
10.3.2 NameSpace 对象	282
10.3.3 Explorer 对象	282
10.3.4 Inspector 对象	283
10.3.5 Outlook 文件夹	283
10.3.6 Outlook 项	284
10.3.7 事件	285
10.4 定制 Outlook 里的菜单和工具栏	285
10.5 调试插件	287
10.6 使用 VSTO 创建的 Outlook 插件的安全性	291
10.7 小结	293
10.8 复习问题	293
第 11 章 安全性和部署	294
11.1 本章主题	294
11.2 VSTO 安全模型	294
11.3 部署 Word 和 Excel 解决方案	298
11.3.1 客户端要求	298
11.3.2 Microsoft PSS VSTO 2005 Client Troubleshooter	298
11.3.3 部署模型	300
11.3.4 应用程序和部署清单	300
11.3.5 部署任务	301
11.3.6 Public Wizard	302
11.3.7 更新和回滚	309
11.3.8 Microsoft Installer 项目	310
11.4 部署 Outlook 解决方案	312

11.5 小结	313
---------------	-----

11.6 复习问题	313
-----------------	-----

第 12 章 把 VBA 解决方案迁移到

VSTO	314
12.1 本章主题	314
12.2 迁移概述	314
12.3 迁移策略	314
12.4 一个 Word VBA 项目的简单迁移	315
12.5 一个 Word VBA 项目的高级迁移	317
12.5.1 复制 Word 模板	318
12.5.2 从模板移除 VBA 代码和 UserForm	318
12.5.3 创建 Windows 窗体	318
12.5.4 启动 Windows 窗体	320
12.5.5 运行解决方案	320
12.5.6 重新设计解决方案	321
12.5.7 Bookmark 视图控件	321
12.5.8 使用操作窗格	322
12.6 VBA 和 VSTO 的互操作	324
12.7 小结	328
12.8 复习问题	328
第 13 章 VSTO 的高级话题	329
13.1 本章主题	329
13.2 ServerDocument 概述	329
13.3 附加和分离基于文档的定制程序	329
13.3.1 使用 ServerDocument 从文档分离解决方案	329
13.3.2 使用 ServerDocument 向文档附加解决方案	330
13.3.3 使用自定义文档属性向文档附加解决方案	331
13.4 读写数据缓存	332
13.4.1 缓存数据结构	332
13.4.2 创建一个有缓存数据的测试文档	332
13.4.3 读取文档数据缓存	334
13.4.4 运行解决方案	337
13.5 清除数据缓存	339
13.6 小结	340
13.7 复习问题	340

第 14 章 VSTO 2005 SE 和 2007

Microsoft Office System	341
14.1 本章主题	341
14.2 VSTO 2005 SE 介绍	341
14.2.1 所有发布版本里的功能	342
14.2.2 VSTO 2005 SE 的插件模型	343
14.3 创建插件	347
14.4 定制功能区	347
14.4.1 功能区项	348
14.4.2 和 XML 协同工作	352
14.4.3 定制内置的功能区	354
14.4.4 向功能区添加其他控件	359
14.4.5 快速访问工具栏	365
14.4.6 不受支持的功能区功能	366
14.4.7 Outlook 里的功能区	367
14.5 创建自定义任务窗格	372
14.5.1 为 PowerPoint 创建插件	372
14.5.2 打开和关闭自定义任务窗格	374
14.5.3 管理多个 Word 文档上的 任务窗格	375

14.5.4 把文档级别的定制程序转换成

VSTO 2005 SE 插件	377
14.5.5 显示多个任务窗格	382
14.6 创建自定义窗体区域	383
14.6.1 创建 Outlook 窗体区域文件	383
14.6.2 创建新的窗体区域	383
14.6.3 保存窗体区域	384
14.6.4 创建 VSTO 2005 SE Outlook 插件项目	384
14.6.5 在你的项目里内嵌窗体区域	384
14.6.6 创建和注册窗体区域清单	385
14.6.7 创建窗体区域注册表文件	386
14.6.8 创建窗体区域启动类	386
14.6.9 连接窗体区域控件	388
14.6.10 连接窗体区域	389
14.6.11 运行项目	389
14.7 小结	390
14.8 复习问题	391
附录 A 创建代码段	392
附录 B 创建 Inspector CommandBar	395

第一部分 VSTO 概述

第 1 章 VSTO 入门

千里之行，始于足下。

——孔子

1.1 本章主题

- 什么是 VSTO
- 为什么要用 VSTO 代替 VBA
- VSTO 的功能
- 创建 VSTO 解决方案
- VSTO 如何与 Visual Studio 集成
- 小结
- 复习问题

1.2 什么是 VSTO

Visual Studio 2005 Tools for Office (VSTO) 使你可以用 Visual Basic 2005 或者 Visual C# 扩展 Office 2003 应用程序 (例如 Word、Excel、InfoPath 和 Outlook)。你是使用强大的 Visual Studio 开发环境来创建你的定制程序，而不是使用 Visual Basic for Application (VBA) 或 Office 里的 Visual Basic Editor (VBE)。无论你是创建简单的数据录入应用程序还是复杂的企业解决方案，VSTO 都使之变得容易。

正如其名，VSTO 是一套用于创建自定义 Office 应用程序的 Visual Studio 工具包。要使用 Visual Studio 的 VSTO 功能，你必须安装独立的 Visual Studio 2005 Tools for the Microsoft Office System 或者 Visual Studio Team System。

要使用 VSTO 定制 Office，你得先在 Visual Studio 里创建一个新的 Office 项目。VSTO 的默认安装包含了 5 个与 Office 相关的 Visual Studio 项目模板：其中 4 个用于为 Word 和 Excel 创建文档级别的定制程序 (document-level customization)，而剩下的 1 个则用于以 Outlook 插件的形式创建应用程序级别的定制程序 (application-level customization)。VSTO 还包含一个用于 InfoPath 扩展的可选模板。然而，由于本书是为那些有兴趣从 VBA 转到 VSTO 的开发者而写的，InfoPath 的开发就不做介绍了。因为 InfoPath 并不支持 VBA，所以无法进行比较。

文档级别的定制程序是指程序的代码关联到特定的文档而不是整个应用程序。需要记住的是，那些代码并不像 VBA 定制程序那样存放在文档或模板里，而是存放在与文档相关联的代码库里，我们把这些代码库称为 程序集 (assembly)。当你创建好新的项目时，你就可以通过主互操作程序集 (Primary Interop Assembly, PIA) 访问 Office 的对象模型。Office PIA 允许 VSTO 定制程序与 Office 的对象模型进行交互。你将在第 3 章了解到更多这方面的内容。

VSTO 还提供了增强的 Office 对象，你可以用它们来编程。比如说，你可以找到 VSTO 版的 Excel 工作簿、工作表和范围 (range)，这些增强的功能在本地 Excel 对象模型里是找不到的。比如说，你可以直接在 Excel 电子表格或者 Word 文档上添加 .NET 控件 (也被称为 Windows Forms 控件) 然后把数据直接绑定到控件上。

应用程序级别的定制程序对整个应用程序都可用。VSTO 为 Outlook 2003 提供了插件模板。你可以使用 Visual Studio 2005 Tools for the 2007 Microsoft Office System (VSTO 2005 SE) 为 Office 2007 的 6 类应用程序和 Office 2003 的 5 类应用程序创建插件。你将在第 14 章了解到更多 VSTO 2005 SE 的内容。

1.3 为什么用 VSTO 代替 VBA

VSTO 为想构建 Office 应用程序的开发者提供专业的开发环境。有很多理由让你想用 VSTO 创建解决方案，也有一些理由让 VBA 成为你的解决方案的更好选择。

你可能选择 VBA 创建 Office 解决方案，如果你想做的是：

- 为比 Office 2003 专业版更老的 Office 版本创建应用程序级别的定制程序。
- 为 VSTO 或 VSTO 2005 SE 不支持的 Office 应用程序创建应用程序级别的定制程序，例如 Microsoft Access。
- 为 Office 2003 标准版创建文档级别的定制程序。
- 通过全局模板为 Word 和 Excel 创建应用程序级别的定制程序。
- 通过创建名字相同的宏，覆盖 Word、Excel 和 Outlook 的本地命令。

根据组织的需求，禁止把 .NET Framework 部署到最终用户的电脑上。

你可能选择使用 VSTO 或者 VSTO 2005 SE 创建 Office 解决方案，如果你想做的是：

- 易于向你的解决方案添加智能标记。
- 提供仅对特定文档可用的智能标记。
- 在文档和任务窗格上使用丰富的控件集。
- 在文档里储存数据以便可以离线使用，并且，文档下次在线时，数据要易于更新。
- 创建具有稳定加载和卸载机制的插件。
- 仅用几行代码就可以定制任务窗格 (task pane)。
- 易于定制 Office 2007 的功能区 (Ribbon)。
- 定制 Outlook 2007 窗体区域 (Outlook form region)。
- 访问 .NET Framework 类库。
- 使用更强大的调试工具。
- 使用 Visual Studio 集成开发环境里的众多工具，例如 Data Sources 窗口和丰富的智能感知功能。