

第 1 章

初识 Visual Basic 2015

本章主要内容：

- 事件驱动编程
- Visual Basic 2015 的安装
- Visual Basic 2015 集成开发环境(Integrated Development Environment, IDE)
- 创建简单的 Windows 程序
- 使用集成的帮助系统

用 Visual Basic 2015 和 Windows 8 或新的 Windows 10 进行编程是令人兴奋的。Windows 10 是微软最新发布的 Windows 操作系统，其中包含许多使 Windows 编程更有趣的新功能。Windows 用户界面和 Visual Basic 2015 已经获得重大改进，非常便于编写专业的 Windows 应用程序、Web 应用程序和 Web 服务。你还没有升级到 Windows 8 或 Windows 10 吗？不必担心，Visual Basic 2015 也可作为 Windows 的以前版本编写专业的应用程序。

即使以前从未编写过任何程序，通过对本书的学习，也仍然可以快速掌握 Visual Basic 2015 编程语言。我们将循序渐进地介绍 Visual Basic 2015 的基础知识，并在后续章节中根据所学到的相关知识构建应用程序。首先请深吸一口气，并缓慢吐出，并告诉自己可以毫不费力地掌握它。

编写计算机程序非常类似于教小孩系鞋带。要想成功，必须找到给出指令的正确方式。使用 Visual Basic 2015 语言就可以告诉计算机该如何去做。但计算机像小孩子一样，只有把事情解释得非常清楚，它才会明白。如果你以前从未编写过程序，可能学习起来会费劲些。然而，Visual Basic 2015 是一门可用来解释某些复杂任务的简单语言。能理解最基础的知识固然很好，而 Visual Basic 2015 能把编程人员从编写复杂 Windows 程序的繁杂工作中解放出来，让他们集中精力去解决实际问题。

Visual Basic 2015 有助于创建运行在 Windows 操作系统(如 Windows 7/8/10、Windows Server 2008 和 Windows Phone)下的应用程序。如果正在阅读这本书，那么你可能希望或需要创建这类程序。即使你以前从未接触过编程，但通过阅读本书中“试一试”部分的内容，也会逐渐熟悉 Visual Basic 2015 语言的方方面面，以及 Microsoft .NET Framework 的基础知识。你会发现，其实它比我们想象的要

容易。在了解这一点之前，你将感受到可以轻松地使用 Visual Basic 2015 创建各种不同的程序。

Visual Basic 2015 也可用于创建 Web 应用程序和 Web 服务，以及可在 Pocket PC 或智能手机上运行的移动应用程序。然而，在扩展到其他平台之前，首先应该先学习 Windows 应用程序。

1.1 事件驱动编程

Windows 程序完全不同于过去的 MS-DOS 程序。DOS 程序自始至终遵循一条相对严格的路径。虽然这不一定限制程序的功能，却限制了用户编程时采用的方法。编写 DOS 程序就像穿过一条走廊，要到达终点，必须穿过整条走廊，清除途中可能遇到的所有障碍，而沿途只能打开一些特定的门。

另一方面，Windows 提供了事件驱动编程的方法。这些事件包括：单击按钮、改变窗口大小、改变文本框中的项等。我们编写的代码可以响应这些事件。用走廊的示例进行类比：在 Windows 程序中，要到达走廊的终点，只需单击一下终点就行了，而不用考虑这条走廊。如果到达了终点，却发现那不是我们想去的地方，那么可以直接奔向新的目的地，而不用返回到起点。程序会对这个动作做出反应，并且执行必要的动作来完成指定的任务。

Windows 编程的另一大优点是硬件的抽象化。这就是说，Windows 负责与硬件的通信。我们不需要知道市场上每种激光打印机的内部工作原理，就可以打印出需要的文件；也不需要研究图形卡的原理就可以编写游戏。Windows 提供了能与硬件制造商编写的驱动程序通信的通用例程，并封装了这个功能。这就是 Windows 如此成功的主要原因。这些通用例程称为 Windows API(应用编程接口)，.NET Framework 中的类负责与这些 API 通信。

在 1991 年 Visual Basic 1.0 问世以前，开发人员必须精通 C 和 C++ 编程以及 Windows 系统本身的基本构建块(即 Windows API)。这一复杂性意味着，只有经过正规培训的专业人员才能编写出在 Windows 上运行的软件。Visual Basic 的问世改变了这一切，据估计，现在用 Visual Basic 语言编写的代码量远高于使用其他任何一种语言编写的代码量。

Visual Basic 取消了用户界面(UI)繁杂的代码编写工作，使 Windows 编程焕然一新。它允许程序员绘制自己的 UI，以便集中精力去解决一些业务问题。一旦创建了 UI，程序员就可向其中添加代码来响应事件。

Visual Basic 从一开始就是可扩展的。第三方供应商很快看到了可对开发人员提供帮助的可重用模块市场。这些模块或控件最初被称为 VBX(以其文件扩展名命名)。在 Visual Basic 5 之前，如果不喜欢某一按钮的使用方式，那么可以购买或自定义该按钮。但是，这些控件必须用 C 或 C++ 编写。数据库访问实用程序就是这类控件中最早的一批。Visual Basic 5 版本中引入了 ActiveX 的概念，允许开发人员创建自己的 ActiveX 控件。

Visual Basic 3 的出现又一次改变了编程世界。现在可以用 Visual Basic 来创建可供用户直接访问的数据库应用程序(即所谓的前端应用程序)，不再需要依赖第三方的控件。微软使用数据访问对象(Data Access Object, DAO)来完成这项任务，它允许程序员像操作用户界面那样处理数据。

为了把开发人员吸引到 Windows 95 平台，Visual Basic 3.0 又升级到 Visual Basic 4 和 5。它们使开发人员更容易编写代码，而这些代码又可供使用其他语言的开发人员使用。Visual Basic 6 集成了 ActiveX Data Object (ADO)，提供了一种访问数据库的全新方法。微软开发 ADO 的目的是帮助 Web

开发人员使用 Active Server Pages (ASP)来访问数据库。经过多年的改进后, Visual Basic 已经在编程领域占据了主导地位。它可以帮助开发人员在数字化时代编写出健壮的、可维护的应用程序。

1.2 安装 Visual Basic 2015

Visual Basic 2015 有如下两个版本:

- 作为 Visual Studio 2015 的一部分, Visual Studio 2015 是一个套装语言软件, 其中还包含 C#(读作 C-sharp)和 Visual C++。Visual Studio 2015 产品系列包括 Visual Studio Professional Edition 或 Visual Studio Tools Team Edition。Team Edition 附带的工具可用于创建和管理大型企业级应用程序的开发。
- Visual Basic 2015 Express Edition(针对学生和初学者的免费版本), 包括 Visual Basic 2015 语言和 Visual Studio 2015 附带的一些工具和功能的简化版本。

这两个版本都可以用于创建在 Windows 平台下运行的应用程序, 安装过程都非常简单。事实上, Visual Studio 安装程序可以非常智能地确定计算机需要安装什么内容。

下面的“试一试”部分基于 Visual Studio 2015 CTP 6 的安装过程。大多数安装过程都十分简单, 在大多数环境下只需要接受默认安装设置即可。不管安装的是哪一种版本, 只要选择默认的安装选项, 安装过程都可顺利完成。

试一试: 安装 Visual Basic 2015

可以使用两种常用的方法来安装 Visual Studio。首先可以从下载的镜像文件中刻录 DVD, 也可以使用 Web 安装程序。如果使用 Web 安装程序, 则需要按照屏幕上的提示进行操作。而如果选择下载 ISO 文件, 则应按照下列步骤进行操作:

(1) 下载 vs2015.ctp_ult_enu.iso, 右击该文件, 选择上下文菜单中的 Mount。Visual Studio 2015 DVD 会在新的文件浏览器窗口中打开。需要运行 DVD 根目录下的 vs_ultimate.exe。接着单击 OK, 启动安装程序。安装程序初始化后, 会出现如图 1-1 所示的界面。这个对话框显示了安装的地址和大小。

(2) 还需要在这个屏幕上选择安装类型, 应选择 Typical。之后, 单击 Install 开始安装。



提示: 如果你是 Windows Vista、Windows 7 或 Windows 8 用户, 那么可能会被提示需要运行安装程序, 这种情况下, 要拥有相关权限才能让安装程序继续。一旦运行了安装程序, 就可以休息一下, 直到所有功能设置完毕。根据所选择的安装功能以及计算机的速度, 整个安装过程可能要花费 20 分钟或者更长时间。

(3) 一旦完成安装, 就会显示一个说明安装状态的对话框。之后可能需要重启计算机。现在让我们畅游 Visual Basic 2015 世界吧!

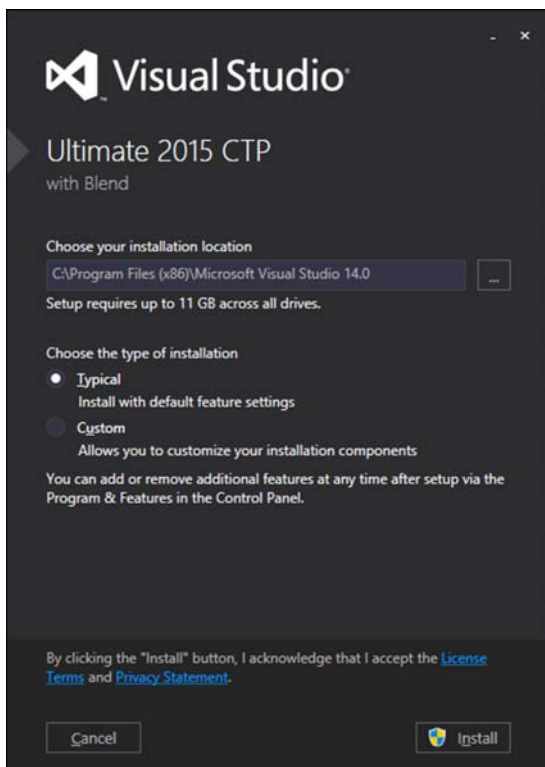


图 1-1

1.3 Visual Basic 2015 集成开发环境

用 Visual Basic .NET 语言编写应用程序并不需要 Visual Basic 2015，.NET Framework 具备运行 Visual Basic .NET 代码的能力。所以，使用诸如 Notepad 的文本编辑器就可以编写所有的 Visual Basic .NET 代码。可用鞋子充当锤子来敲钉子，但灵巧的气动钉枪要有效得多。同样，到目前为止，编写 Visual Basic .NET 代码最简单的方法是使用 Visual Studio 2015 集成开发环境(IDE)，即运行 Visual Basic 2015 时出现的窗口、对话框等。IDE 提供了普通文本编辑器所没有的丰富功能，例如代码检查、已完成应用程序的可视化表示，以及一个显示构成项目的所有文件的浏览器。

1.3.1 配置启动页面

IDE 集成了一组工具，使开发软件变得更为简便。如果 Visual Studio 2015 没有运行，就启动它，看看会得到什么。如果使用默认安装选项，就应进入 Window 应用程序或搜索菜单，再选择 Visual Studio 2015 CTP。此时会显示一个闪屏，以便登录。因为本书是脱机工作，所以应选择 Not Now, Maybe Later。创建一个免费账户后，Visual Studio 会保存该设置，使所有计算机与该设置同步。可以更直接地访问一些在线资源。

接着会出现 Choose Default Environment Settings 对话框。选择 Visual Basic Development Settings 选项和颜色方案，然后单击 Start Visual Studio，如图 1-2 所示。

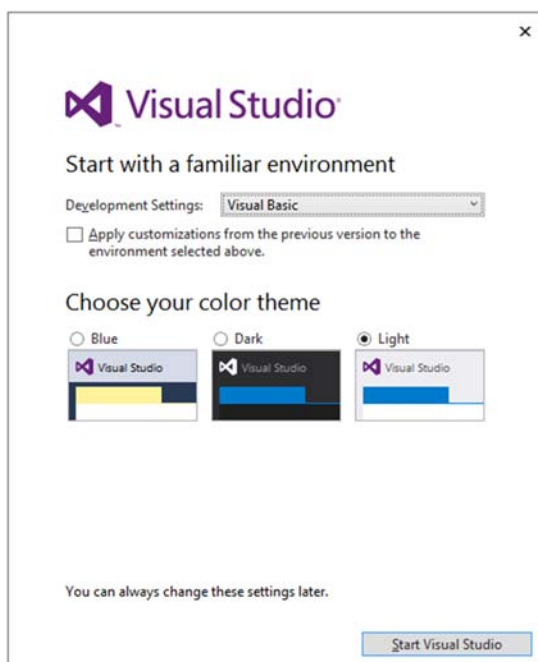


图 1-2

Visual Studio 根据所选的设置配置环境后，会显示 Microsoft Development Environment，如图 1-3 所示。

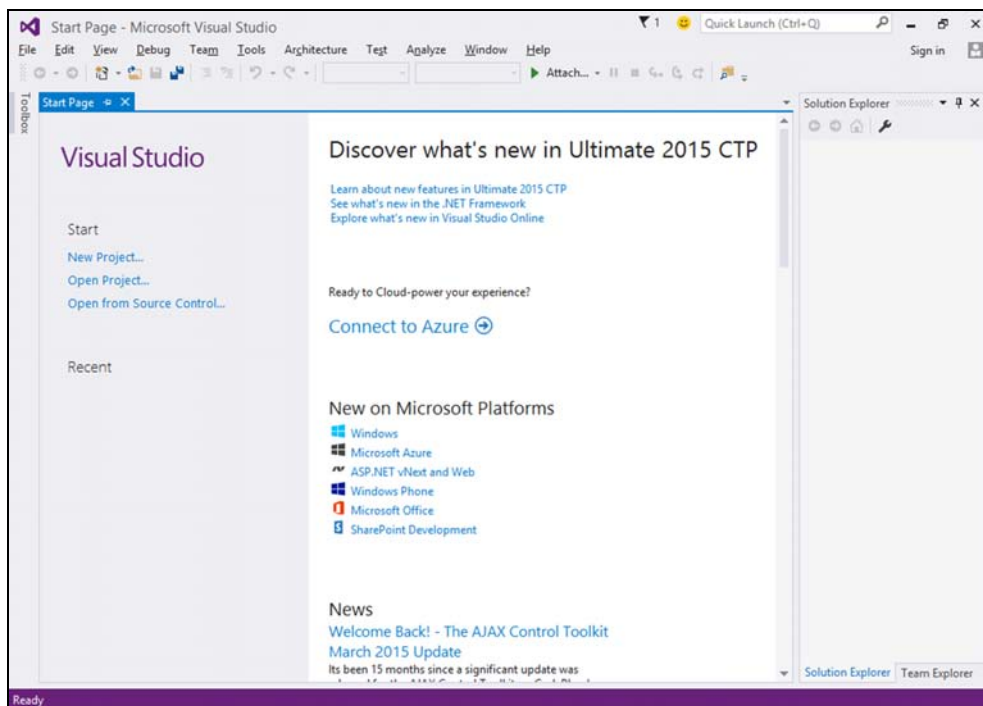


图 1-3

1.3.2 菜单

至此,你可能希望开始编写代码。但请等一下,我们将从菜单和工具栏开始 Visual Basic 2015 探险之旅。实际上,这些菜单和工具栏与其他 Windows 应用程序中的菜单和工具栏没有太大区别(但与 Office 2007 的功能区和某些较新的 Windows 应用程序不同)。

Visual Studio 2015 的菜单是动态的,可以根据需要添加或删除菜单项。在浏览空白 IDE 时,菜单栏中只有 File、Edit、View、Build、Debug、Team、SQL、Data、Format、Tools、Unit Test、Analyze、Window 和 Help 菜单。但是,当开始一个项目时,就会显示 Visual Studio 2015 的完整菜单,如图 1-4 所示。



图 1-4

其实没有必要详细介绍每个菜单,在学习本书的过程中,你会逐渐熟悉它们。下面简述每个菜单的功能:

- **File:** 大部分软件程序都有 File 菜单。它已经成为一个标准,通常可从中找到退出应用程序的选项。这个菜单还有打开、关闭单个文件和整个项目的选项。
- **Edit:** Edit 菜单提供的选项有 Undo、Redo、Cut、Copy、Paste 和 Delete。
- **View:** 通过 View 菜单可以快速访问 IDE 中的窗口,如 Solution Explorer、Properties、Output 和 Toolbox(工具箱)窗口等。
- **Debug:** 通过 Debug 菜单可以在 Visual Basic 2015 IDE 中启动和停止运行应用程序。同时还可以访问 Visual Studio 2015 调试器。调试器允许单步执行代码,以观察它的执行情况。
- **Team:** 该菜单可以连接到 Team Foundation Server。当与一个团队合作开发软件时可以使用该菜单。
- **Tools:** Tools 菜单有配置 Visual Studio 2015 IDE 的命令,以及启动其他已安装的外部工具的链接。
- **Architecture:** 这个菜单提供了高级设计工具选项。
- **Test:** Test 菜单可以创建和查看应用程序的单元测试,以便在不同场景中运行源代码。
- **Analyze:** 这个菜单帮助检查代码。使用这个菜单运行性能和代码分析工具,以帮助编写出更好的代码。
- **Window:** Window 菜单也是标准的。它允许像 Word 和 Excel 那样同时打开多个窗口。通过这个菜单中的命令可以在 IDE 中切换窗口。
- **Help:** 通过 Help 菜单可以访问 Visual Studio 2015 文档。访问该文档有多种方式,如通过帮助目录、索引或搜索。Help 菜单中的选项还可以连接到微软的 Web 站点上,从而获取更新内容或报告问题。

1.3.3 工具栏

IDE 中有许多工具栏,包括 Formatting、Image Editor 和 Text Editor,通过选择 View | Toolbars 命令,可以在 IDE 中添加或删除这些工具栏。每个工具栏都提供了对常用命令的快速访问,而不必选择相应的菜单项。例如,单击图 1-5 所示的默认工具栏(标准工具栏)最左边的图标(New Project),就相当于选择了 File | New Project 命令。

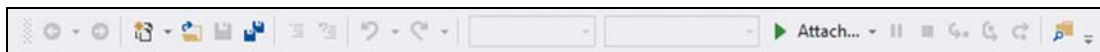


图 1-5

在这个工具栏中，相关的选项被分组并用竖线分开。

- **navigation:** 第一组图标可用来在代码中移动光标。在代码中可以使用这些图标来前后移动光标。
- **project and file options:** 通过使用接下来的 4 个图标，可以实现 File 和 Project 菜单所实现的常用项目和文件操作，比如打开和保存文件。
- **code commenting:** 第 3 组图标用于注释和取消注释某段代码。在调试时，如果需要注释一段代码，以确定不执行这些代码会有什么结果，就可以使用这组图标。
- **managing code edits:** 第 4 组图标用于执行撤消与恢复操作，以及在代码中移动光标。
- **code step through:** 第 5 组图标用于启动(绿色的三角形)、暂停和停止运行应用程序。同时，还可以使用该组的 3 个图标单步执行代码、跳过整块代码和跳出过程。可以使用解决方案配置来生成项目，以便调试代码或者产生可向用户发布的软件。第 8 章将深入介绍这些图标。
- **find in files dialog:** 最后一个图标可以访问 Find In Files 对话框。这是一个常用且非常重要的功能。此外，还可以通过按下 Ctrl+F 组合键访问该对话框。

如果忘了某个图标的用途，可以把鼠标指针悬停在它上面，就会出现一个工具提示，显示这个工具栏选项的名称。

如果想查看 IDE 中的每个窗口，可以单击 View 菜单，并选择相应的窗口；但此时这些窗口都是空的，因此没有太大的意义。了解 IDE 功能的最好方法是用它来编写一些代码。

1.4 创建简单的应用程序

如果想完成对 IDE 的探索工作，需要创建一个项目，从而使前面介绍的图 1-3 所示的窗口内出现一些有趣的内容。

试一试： 创建 Hello User 项目

下面的“试一试”练习将创建一个非常简单的应用程序 Hello User，该程序允许用户输入一个人名，并在消息框中显示对这个人的问候信息。

- (1) 单击工具栏上的 New Project 按钮。
- (2) 在 New Project 对话框中，选择左侧 Installed Templates 树状列表中的 Visual Basic，右边的 Templates 面板将显示所选项目类型的所有可用模板。选择 Windows Forms Application 模板。
- (3) 在 Name 文本框中输入 Hello User 并单击 OK 按钮。New Project 对话框如图 1-6 所示。

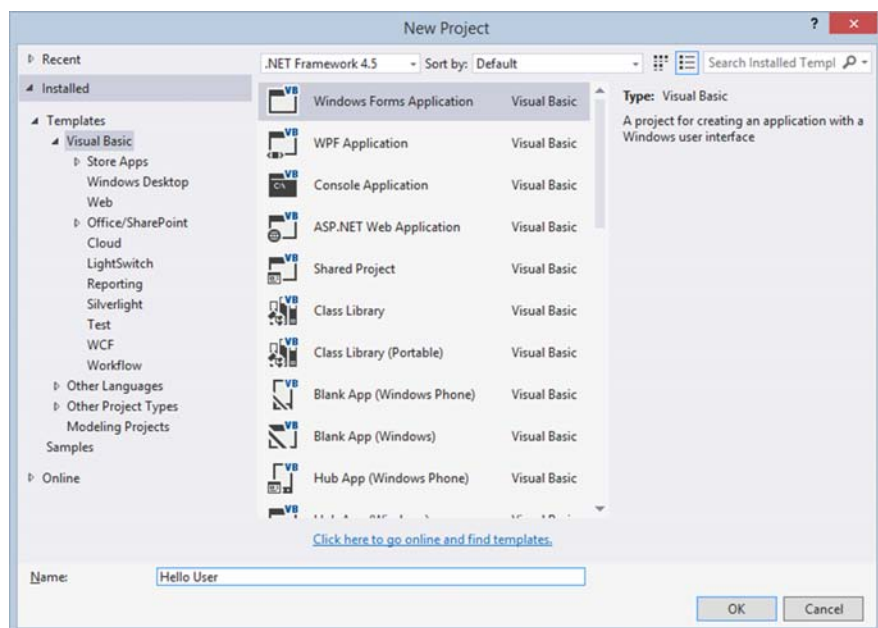


图 1-6

Visual Studio 2015 允许将应用程序面向特定版本的 .NET Framework。New Project 对话框中 Templates 窗格顶部的组合框中默认选择了 .NET Framework 4.5，但也可以将应用程序面向 .NET Framework 的更早版本。

IDE 将创建一个空的 Windows 应用程序。目前，Hello User 程序只包含一个空窗口 Windows Form(有时只是一个窗体)，其默认名称为 Form1.vb，如图 1-7 所示。

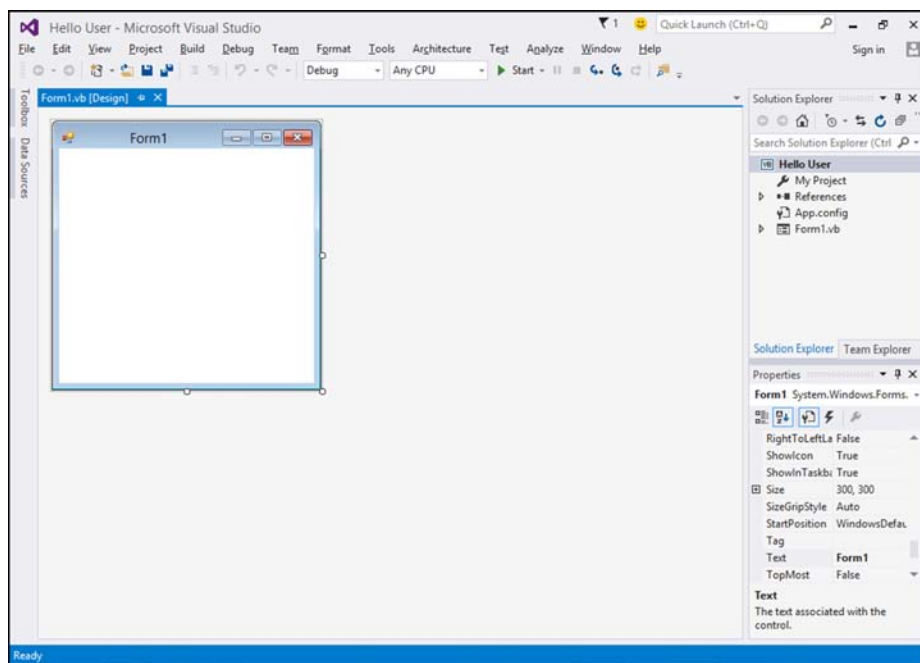


图 1-7

无论何时,不管是由 Visual Studio 2015 创建一个新文件作为项目创建过程的一部分,还是由用户创建一个新文件,都会用一个名称来描述这个文件(本例中是 Form),并在其后加上一个数字。

Visual Studio 2015 集成开发环境中的窗口

现在,IDE 中的各种窗口将一一亮相。在继续上面的示例之前,先简要介绍一下这些窗口。



注意: 如果某个窗口在屏幕上未出现,可以使用 View 菜单显示它。如果对某个窗口的位置不满意,可以单击该窗口的标题栏并将其拖到一个新位置。IDE 中的窗口可以浮动,也可停靠(见图 1-7)。

下面介绍最常用的窗口:

- **Toolbox:** Toolbox 包括可添加到应用程序的可重用控件和组件。其中包括按钮、数据连接器、可购买或自行开发的定制控件。
- **Design:** 许多操作都是在 Design 窗口中进行的。在 Design 窗口中,可以绘制用户界面。这个窗口有时也称为设计器。
- **Solution Explorer:** Solution Explorer 窗口可分级显示解决方案。一个解决方案可以包括多个项目,而一个项目可以包含解决某个特定问题的窗体、类、模块和组件。
- **Properties:** Properties 窗口显示了所选对象的可用属性。虽然可以在代码中设置这些属性,但有时在设计应用程序时设置它们会更容易,例如,在窗体上绘制控件。注意 File Name 属性的值为 Form1.vb,这是窗体代码和布局信息的物理文件名。

试一试:

继续创建 Hello User 项目

接下来,对窗体命名并设置一些属性。

(1) 把窗体的名称改为更能反映这个应用程序的名称。在 Solution Explorer 窗口中单击 Form1.vb,接着在 Properties 窗口中把 File Name 属性的值从 Form1.vb 改为 HelloUser.vb。然后按 Enter 键,如图 1-8 所示。

(2) 当更改相关属性时,必须按 Enter 键或者单击其他属性,以便使所做的更改生效。注意,此时 Solution Explorer 中窗体的文件名也更新为 HelloUser.vb。

(3) 单击 Design 窗口中显示的窗体。Properties 窗口就会显示窗体的 Form 属性(取代了刚才的 File 属性)。



注意: Properties 窗口已经有了很大的不同,原因是同一文件有两个不同的视图。当窗体的名称在 Solution Explorer 窗口中被突出显示时,Properties 窗格显示的是窗体的物理文件属性。当窗体的名称在 Design 窗口中被突出显示时,Properties 窗口显示的是窗体的可视化属性和逻辑属性。

Properties 窗口十分便于设置控件的属性。属性是某个对象的内部数据集,它们常用于描述对象

的外观或行为。在图 1-9 中, 属性按字母顺序显示, 同时, 属性也可以按 Accessibility、Appearance、Behavior、Data、Design、Focus、Layout、Misc 和 Window Style 类别来进行分组。

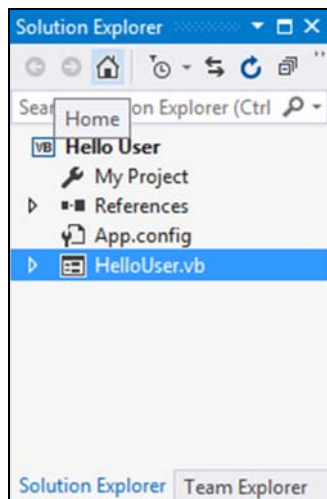


图 1-8

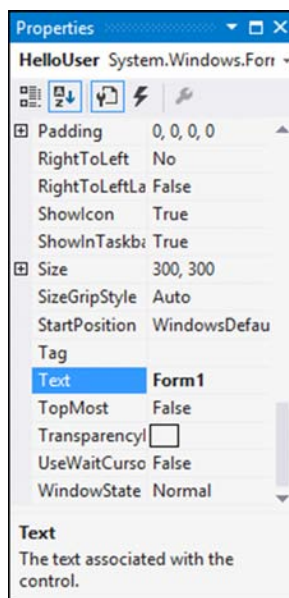


图 1-9

(4) 现在, 窗体的标题(显示在顶部, 即 Text 属性)是 Form1, 它的描述性不强, 下面修改它, 使其反映这个应用程序的作用。选择 Properties 窗口中的 Text 属性, 将它的值改为 Hello from Visual Basic 2015, 并按下 Enter 键。注意, 这时窗体的标题已更新, 以反映所做的更改。



注意: 如果找不到属性, 可单击 Properties 窗口顶部工具栏中的 AZ 按钮, 使属性按名称而不是按类别排序。

(5) 现在就完成了应用程序的创建。单击 Visual Studio 2015 工具栏上的 Start 按钮(绿色三角形), 运行该应用程序。在阅读本书时, 如果遇到“运行项目”或“启动项目”, 就是指单击 Start 按钮。此时会显示一个名为 Hello from Visual Basic 2015 的空窗口。

一切都非常简单, 但此刻这个小程序什么也不能做, 下面给它添加一些交互功能。为此, 先在窗体中添加一些控件: 一个标签、一个文本框和两个按钮。在这些添加操作中, 可以看出工具箱是如何使添加功能变得非常简便的。当你查看相关代码时可能会感到惊讶! Visual Basic 2015 的一个优点是它可以完成应用程序中的许多工作, 却不用编写任何代码。当然, 代码依旧存在, 只是在后台, 并且由 Visual Basic 2015 编写。

1.4.1 工具箱

要访问工具箱, 可以选择 View | Toolbox 命令, 也可以单击 Standard 菜单栏中的 Toolbox 图标, 还可以按下 Ctrl+Alt+X 组合键。Toolbox 标签显示在 IDE 的左边, 把鼠标指针停在这个标签上, 就会使工具箱窗口显示出来, 并覆盖部分窗体。

工具箱包含以节点视图形式出现的各种控件和组件，它们都可以放到窗体中，方法是选择文本框、按钮、单选按钮和组合框等控件，将它们拖放到窗体中。Hello User 应用程序只使用 Common Controls 节点中的控件。图 1-10 列出了 Windows Forms 应用程序中的一些常用控件。

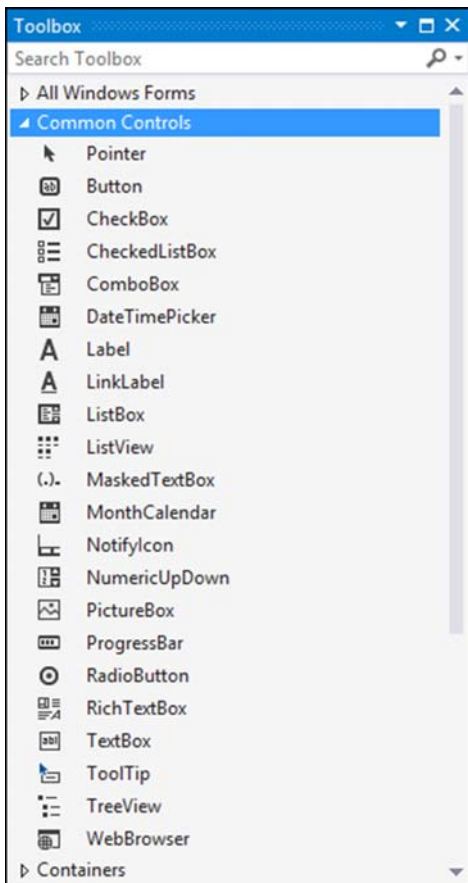


图 1-10

可以按任意顺序将控件添加到窗体中，所以，在文本框控件后添加标签控件，或者在标签控件前添加按钮都是可行的。

试一试：**向 Hello User 应用程序添加控件**

在接下来的“试一试”练习中，我们开始添加一些控件。

(1) 如果项目正在运行，请首先将其停止运行，因为下面要在窗体中添加一些控件。停止项目最简单的方法是单击窗体右上角的“关闭”(X)按钮，或者单击 IDE 中的蓝色方块(如果把鼠标指针悬停在它上面，就会显示工具提示 Stop Debugging)。

(2) 给窗体添加一个 Label 控件。单击工具箱中的 Label，将其拖放到窗体设计器的合适位置(要在窗体上添加控件，还可以在工具箱中双击所需的控件；或者在工具箱中单击所需的控件，然后将其拖放到窗体上)。

(3) 如果刚才拖放的 Label 控件不在希望的位置，或者大小不合适，那也不必担心。将该控件拖

放到窗体中后,就可以调整它的尺寸,或者在窗体中移动它。图 1-11 所示为 Label 控件被拖放到窗体中的样子。要移动它,可以单击该控件,将它拖到需要的位置。Label 控件会根据 Text 属性的设置自动调整尺寸。

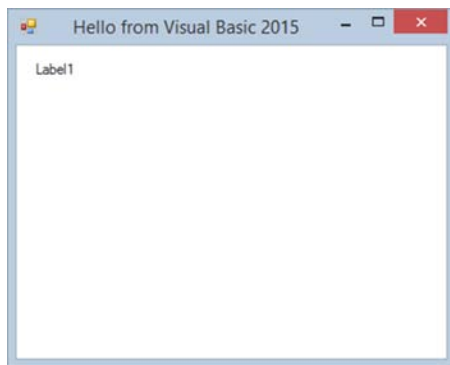


图 1-11

(4) 在窗体中绘制好一个控件后,就可以为其设定名称和要显示的文本。Designer 窗口右边的 Properties 窗口现在已经变为 Label1,说明当前正在查看它的属性。在 Properties 窗口中,将新标签的 Text 属性设置为 Enter Your Name。注意,只有按下 Enter 键或单击另一个属性后,窗体上的标签才会根据 Text 属性的设置值自动调整尺寸。现在将它的 Name 属性设置为 lblName。

(5) 在标签的下面添加一个文本框,以便输入一个名称。重复添加标签的过程,但这次是从工具箱中选择 TextBox 控件。一旦把这个控件拖放(或双击)到合适位置,就可以使用 Properties 窗口,将它的 Name 属性设置为 txtName,如图 1-12 所示。注意控件左边和右边的尺寸调整句柄,它们可以水平地调整文本框的尺寸。

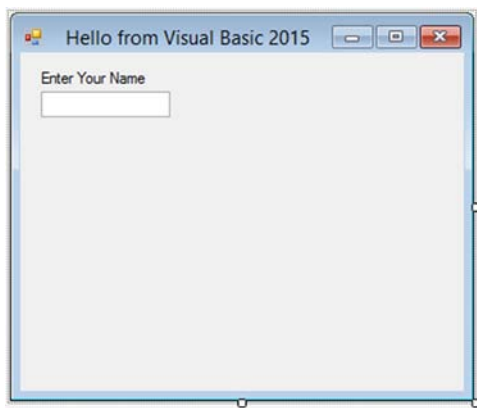


图 1-12

(6) 在窗体的左下角,添加一个 Button 控件,方法与添加标签和文本框的方法完全相同。将它的 Name 属性设置为 btnOK,将 Text 属性设置为&OK。现在窗体应如图 1-13 所示。

按钮的 Text 属性中使用了&符号,以创建键盘快捷方式(即热键)。&符号后面的字母会添加下划线,如图 1-13 所示,说明除了使用鼠标外,还可以同时按 Alt 和这个字母键来选择这个按钮(在一些配置中,在用户按下 Alt 键之前,不会给该字母添加下划线)。在这个示例中,按下 Alt+O 组合键与单击 OK 按钮的效果相同,这一切都不需要编写任何代码。

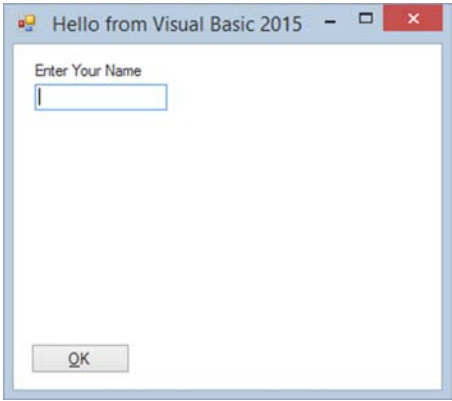


图 1-13

(7) 将 Button 控件从工具箱拖放到窗体上，在窗体的右下角添加第 2 个按钮。注意仔细观察窗体右下角，会看到一条蓝线，如图 1-14 所示。通过该蓝线可以将新加的 Button 控件与窗体上已有的 Button 控件对齐。根据新控件的位置，该蓝线可以将控件彼此左、右、上或下对齐。这条蓝线还可以使控件和窗体之间的边距保持一致。将该按钮的 Name 属性设置为 btnExit，将 Text 属性设置为 E&xit。这时，窗体将如图 1-15 所示。



图 1-14

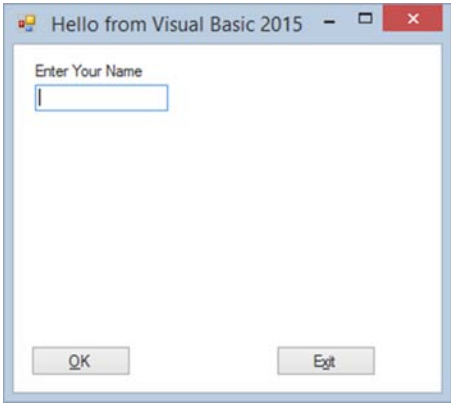


图 1-15

在完成这个示例应用程序之前，我们还要简要探讨一些应遵循的编码规则。

1.4.2 改进型匈牙利表示法

读者可能已经注意到，控件的名称似乎有点古怪。每个名称的前面都要加上一个简写的标识符来描述此控件的类型。这样一来，在浏览代码时，就很容易知道所使用控件的类型。例如，把一个控件简单地命名为 Name，而不带上 lbl 或 txt 前缀，就不清楚该控件是一个可输入名称的文本框，还是一个可显示名称的标签。在前面的示例中，如果把标签命名为 Name1，把文本框命名为 Name2，我们将立刻陷入困境。如果在一两个月之后想对这个程序做些修改，会怎样呢？

在与其他开发人员合作时，采用一致的编码样式是非常重要的。在用许多语言开发应用程序时，控件名称最常用的样式是由 Charles Simonyi 博士提出的(他在加入微软之前在 Xerox Palo Alto 研究中

心(XPARC)工作)。他提出的简短前缀记忆法可以让程序员轻而易举地识别出变量包含的信息类型。因为 Simonyi 博士是匈牙利人,而且该前缀使名称看起来有点像外来词,所以这种命名方式就称为匈牙利表示法。最初的表示法用于 C 和 C++ 开发,Visual Basic 2015 中的表示法就被称为改进型匈牙利表示法。表 1-1 列出了本书常用的一些前缀。

表 1-1 Visual Basic 2015 中常用的前缀

控 件	前 缀
Button(按钮)	btn
ComboBox(组合框)	cbo
CheckBox(复选框)	chk
Label(标签)	lbl
ListBox(列表框)	lst
MainMenu(主菜单)	mnu
RadioButton(单选按钮)	rdb
PictureBox(图形框)	pic
TextBox(文本框)	txt

在查看别人编写的代码或自己数月前编写的代码时,使用匈牙利表示法的代码将大大节省我们的时间,然而,最重要的是它使控件的命名规则保持一致。当开始编写代码时,应选择一个命名约定。我们会看到许多不同类型的命名规则;不要固守其中一种,因为进入其他项目时,必须使用该项目所用的命名规则。一旦选择了命名约定,就应一直使用它。当修改别人的代码时,就应使用他们的命名约定。如果一个项目始终遵循标准的命名规则,维护应用程序时就可以节省大量时间。现在再回到应用程序,并编写一些代码。

1.4.3 代码编辑器

前面定义了 HelloUser 窗体,接下来添加一些代码,让它完成一些有意义的操作。在窗体中添加控件非常简单,给屏幕上的元素添加功能也不难。要为控件添加代码,只要双击控件,打开主窗口中的代码编辑器,如图 1-16 所示。

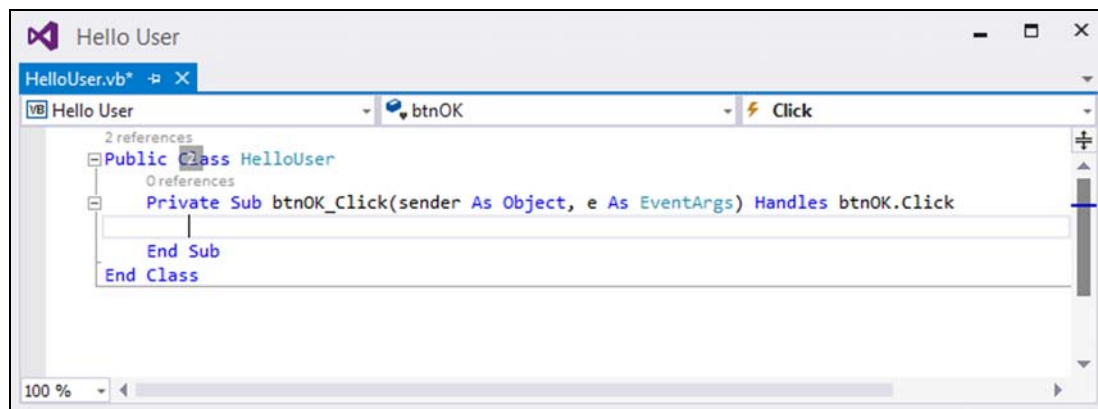


图 1-16

注意，在主窗口中创建了另一个标签，因此现在有 Design 标签和 Code 标签，它们都包含了当前窗体的名称。前面在 Design 标签的窗体上绘制了一些控件，下面在 Code 标签的窗体上编写代码。要注意的是，Visual Studio 2015 已为代码创建了一个文件。窗体的可视化定义及后台代码都放在单独文件中，即 HelloUser.Designer.vb 和 HelloUser.vb。这也是为什么使用 Visual Basic 2015 建立应用程序会如此灵活简单的原因。使用 Design 模式可以可视化方式布局应用程序，然后，使用 Code 视图可以添加部分代码，实现所需的功能。

代码编辑器的顶部有两个组合框，它们提供了到达代码各个部分的快捷方式。左边是 Class Name 组合框，如果展开该组合框，就会列出窗体中的所有对象。右边是 Method Name 组合框，打开该组合框，会显示 Class Name 组合框中所选对象的所有已定义的函数和事件。如果在这个窗体中有许多后台代码，那么这两个组合框可以很快定位到我们希望查看的区域——跳转到所选的代码区域。然而，因为该项目所有的代码都在窗口中，所以不必担心找不到它们。

试一试： 向 Hello User 项目添加代码

(1) 在添加必要的代码之前，先单击 Design 标签，再次显示窗体，然后双击 OK 按钮。代码窗口将再次打开，并显示如下代码，这只是按钮的 Click 事件的“外壳”。我们可以在此输入单击按钮时希望运行的代码。这段代码称为事件处理程序，有时也称为事件过程。

```
Private Sub btnOK_Click(sender As Object, e As EventArgs)
    Handles btnOK.Click

End Sub
```

由于出版时受印刷限制，不可能把 Sub 声明放在一行中。Visual Basic 2015 允许使用下划线符号(_)把代码行断开，以表示这是前一行的续行，而下划线前必须有空格，下一行中代码前的所有空格可忽略。某些情况下，也可以像下面代码所显示的那样，不使用下划线而直接断开代码行。后面将介绍更多关于断开代码行的相关规则。

```
Private Sub btnOK_Click(sender As Object,
    e As EventArgs) Handles btnOK.Click

End Sub
```

Sub 是一个关键字。在编程术语中，关键字是一种特殊的词，用来让 Visual Basic 2015 完成一些特殊的操作。在本例中，Sub 告诉 Visual Basic 2015 这是一个子例程(一个不返回值的过成)。在 Private Sub 和 End Sub 之间输入的代码构成了 OK 按钮的事件过程。

(2) 在这个过程中添加下述突出显示的代码：

```
Private Sub btnOK_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnOK.Click
    'Display a message box greeting to the user
    MessageBox.Show("Hello, " & txtName.Text & _
        "! Welcome to Visual Basic 2015.", _
        "Hello User Message")
End Sub
```


在本书中,有一些代码应被输入到程序中。通常,我们会提示应在什么位置输入这些代码,并解释其作用。黑体代码是应输入到程序中的代码。

(3) 添加完代码后,返回 Design 标签,并双击 Exit 按钮。在 btnExit_Click 事件过程中添加下面突出显示的代码:

```
Private Sub btnExit_Click(sender As Object,
    e As EventArgs) Handles btnExit.Click
    'End the program and close the form
    Me.Close()
End Sub
```

(4) 代码添加完毕后,运行该应用程序。首先,为了保存工作成果,在菜单栏中选择 File | Save All 命令或者单击工具栏上的 Save All 按钮,打开 Save Project 对话框,输入名称和保存项目的位置。

默认情况下,项目保存在以项目名命名的文件夹中,本例中是 Hello User。因为这是解决方案中唯一的项目,所以不需要为与项目同名的解决方案创建单独的文件夹,因此不必选中 Create directory for solution 复选框。

(5) 单击工具栏上的 Start 按钮, Visual Studio 2015 就会编译这些代码。编译过程就是提取刚才编写的 Visual Basic 2015 源代码,把它转换成机器能理解的形式。编译完成后, Visual Studio 2015 将运行(也叫执行)程序,并显示运行结果。

如果 Visual Basic 2015 遇到错误,就会把错误作为任务显示在 Error List 窗口中。双击一个任务,就会显示有错误的代码行。第 9 章将详细讨论如何调试代码中的错误。

(6) 当应用程序载入时,会显示主窗体。输入一个名称并单击 OK 按钮(或按下 Alt+O 组合键),如图 1-17 所示。

这时就出现了如图 1-18 所示的消息框,内容是欢迎 Chris 进入 Visual Basic 2015 世界,Chris 是刚才输入到窗体文本框中的人名。

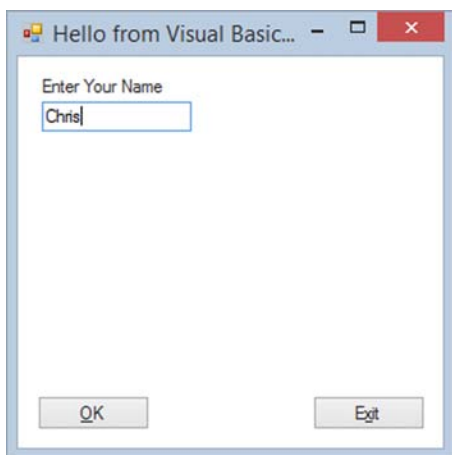


图 1-17

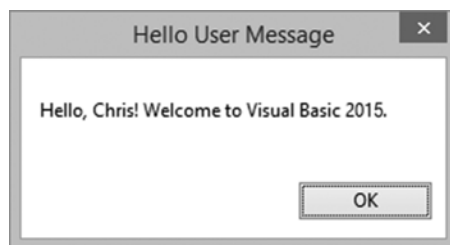


图 1-18

(7) 单击 OK 按钮,关闭消息框后,再单击窗体中的 Exit 按钮,就关闭了应用程序,返回到 Visual Basic 2015 IDE。

示例说明

为 OK 按钮的 Click 事件添加的代码提取了在文本框中输入的名字，并把它作为图 1-18 所示对话框中显示的消息的一部分。

在这个过程中输入的第一行文本('Display a message box greeting to the user')实际上是一个注释。这行文本仅由编写或维护代码的编程人员阅读，计算机并不理会。Visual Basic 2015 中的注释以一个单引号(')开头，其后的所有内容都被编译器视为注释而被忽略。有关注释的内容详见第 3 章。

MessageBox.Show 方法显示一个消息框，它可以接收各种参数。在本例的代码中，其参数是要在消息框中显示的字符串文本。该文本连接了括在引号中的文本所定义的字符串常量。将多个字符串连接成一个长字符串应使用宏字符(&)来实现。

在下面的代码中，连接了一个字符串常量"Hello, "、后面紧跟在 txtName 文本框控件的 Text 属性中包含的值，以及字符串常量"! Welcome to Visual Basic 2015."。传递给 MessageBox.Show 方法的第 2 个参数是用在消息框标题栏中的标题。

最后，在代码行的末尾使用下划线字符(_)，从而允许把代码拆分到多个行中。这告诉编译器参数的剩余代码在下一行中。在创建较长的字符串时，这是很有用的，因为它允许在代码编辑器中查看全部代码，而不需要向右滚动代码编辑窗口。

```
Private Sub btnOK_Click(sender As Object,
    e As EventArgs) Handles btnOK.Click
    'Display a message box greeting to the user
    MessageBox.Show("Hello, " & txtName.Text & _
        "! Welcome to Visual Basic 2015.", _
        "Hello User Message")
End Sub
```

下一个要添加代码的过程是 Exit 按钮的 Click 事件。这里仅输入了代码：Me.Close()。Me 关键字表示窗体本身。窗体的 Close 方法会关闭窗口体，并释放与该窗体相关的所有资源，然后结束程序。

```
Private Sub btnExit_Click(sender As Object,
    e As EventArgs) Handles btnExit.Click
    'End the program and close the form
    Me.Close()
End Sub
```

1.5 使用帮助系统

Visual Basic 2015 的帮助系统在原有版本的基础上进行了改进。随着对 Visual Basic 2015 的进一步学习，用户会越来越熟悉这个帮助系统。为了帮助用户快速搜索到信息，现在概述一下这个帮助系统。

Help 菜单如图 1-19 所示。

如你所见，这个菜单包含的项比一般的 Windows 应用程序多。主要的原因是文档太大，几乎没人能把这些信息全部记住。所幸的是，这并不是问题，因为总是可以简捷地使用帮助系统，或者在论坛上搜索有经验的完成过类似编程任务的人。就把它当作你大脑的安全网络吧。

在代码编辑器中单击某个关键字或按下 F1 键，就可以快速访问特定主题的帮助文档。

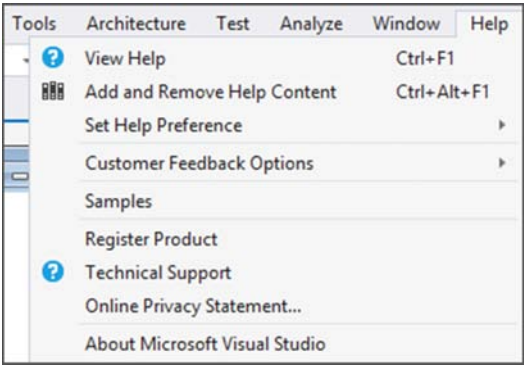


图 1-19

1.6 小结

可以看出，使用 Visual Basic 2015 创建基本的应用程序并不难。本章介绍了 IDE，以及如何运用它快速开发软件。利用工具箱可在窗体上添加控件并便捷设计用户界面；Properties 窗口能迅速配置这些控件；Solution Explorer 提供了一个概览图，列出了构成项目的所有文件。此外我们还编写了少量的代码。

后续章节将详细介绍如何编写代码。在进一步学习 Visual Basic 2015 之前，第 2 章将首先介绍 .NET Framework。这个架构将使所有的 .NET 语言都变得易学易用，并便于进行交互操作。

练习

在附录 A 中可以找到本书中每一章结尾处练习的答案。

1. 使用一个 Textbox 和一个 Button 控件创建一个 Windows 应用程序，当用户单击按钮时在文本框中显示所输入的内容。

本章的主要内容	
主 题	概 念
集成开发环境(IDE)	如何在 IDE 中创建项目，如何在设计视图和代码视图之间切换，如何运行和调试项目
在设计器中为窗体添加控件	如何使用工具箱把控件拖放到窗体上，如何在窗体上移动控件、调整控件的大小
设置控件的属性	如何显示控件中的文本，给控件指定有意义的名称
在代码窗口中为窗体添加代码	如何添加代码来控制程序的行为