

第1章 Visual Basic .NET 概述

1.1 Microsoft Visual Basic .NET 2005 概述

Visual Basic .NET 是面向对象的、支持 Web 的、事件驱动的、功能强大的、易学的、交互性强的、全新的编程语言,使用本书学习 Visual Basic .NET,不要求读者已经学习过 Visual Basic 6.0,只要具备计算机的基本知识即可进入本书的学习。

什么是 Visual Basic .NET? 它的特点是什么? 可以从 4 个方面来认识 Microsoft 的 Visual Basic .NET。

1.1.1 Visual Basic .NET 是 BASIC

首先 Visual Basic .NET 是 BASIC。

BASIC 是 1964 年由两位美国计算机科学家 G. Kemeny 和 Thomos E. Kurtz 在 FORTRAN 语言的基础上创造的一种编程语言,全名为“Beginner's All Purpose Symbolic Instruction Code”,直译为“初学者通用的符号指令代码”,简称为 BASIC,是一种适合初学者学习的易学的编程语言。

BASIC 语言的易学性使它广泛流行并不断发展多年以来,在原来的 BASIC 语言基础上不断进行功能扩充,出现了多种 BASIC 版本,如 BASICA, True BASIC, QBASIC, GWBasic, Quick BASIC, Visual Basic 等,并最终发展到 Visual Basic .NET。

由此可见,Visual Basic .NET 起源于 BASIC,注定了它是一种易学的编程语言。Visual Basic .NET 的最新版本是 2005。

1.1.2 Visual Basic .NET 是可视化的和事件驱动的

随着操作系统由字符界面的 DOS 进化为图形界面的 Windows,使程序开发者面临新的困难,编程者必须将很多精力放在开发图形界面上,这曾经让很多学习 Windows 应用程序开发的人员望而却步。Microsoft 公司 1991 年推出了 Visual Basic 1.0,与以往的 BASIC 语言相比有两方面的重大进展,即“可视化”(Visual)的用户界面设计和“事件驱动”的机制。

“可视化”用户界面设计使开发人员只用鼠标就可以“画”出所需的用户界面,然后用简单的 BASIC 语言编写程序,省却了为设计用户界面所必须编写的成百上千行程序。

“事件驱动”机制不同于传统的面向过程的程序开发,在“事件驱动”机制中所有过程都处于同等的地位,哪一个过程被执行完全取决于用户所引发的事件。事件驱动极大地提高了程序的交互性。

使用“可视化”用户界面设计突出了 Visual Basic .NET 的易学性,而“事件驱动”是轻松实现人机交互的保证。

Visual Basic 经过 2.0,3.0,4.0,5.0,6.0 几个版本发展到 .NET 版本,除了可视化用户界面设计和事件驱动机制以外,还逐渐增加了诸多的功能,如数据库编程、支持多媒体等,使 Visual Basic 成为所有 Windows 编程工具中效率最高的一个。

1.1.3 Visual Basic .NET 支持 Web 程序设计

从 Visual Basic 6.0 到 Visual Basic .NET,最大的改进是对 Web 的全面支持。Visual Studio .NET 是 Microsoft 新一代技术平台,对开发人员而言,.NET 是一个完美的开发平台,它提供了一套公共的运行库和公共语言规范。所有符合该规范的语言都可以使用这套运行库。在 .NET 平台下,各种编程语言共享公共类库,具有类似的编程模型和相差无几的功能。开发人员可以自由选择自己喜爱的语言开发程序。

Visual Basic .NET 是 Visual Studio .NET 的开发工具之一,用于开发 Windows 应用程序和 Web 应用程序等软件。

1.1.4 Visual Basic .NET 是面向对象的程序设计语言

Visual Basic .NET 是完全面向对象的编程语言,Visual Basic .NET 程序设计支持继承、多态、重载、覆盖和异常处理、多线程等面向对象编程语言的机制。面向对象的程序设计充分利用代码的重用性和事件驱动机制,是高效的程序设计语言。学习 Visual Basic .NET 程序设计就是学习最先进的程序设计方法。

总之,Visual Basic .NET 现在是现代的、强壮的、面向对象的、可视化的程序设计语言。

1.2 Visual Basic .NET 集成开发环境

1.2.1 启动 Visual Basic .NET

第一次启动 Visual Basic .NET 会出现如图 1-1 所示的“选择默认环境设置”窗口,此时应当选择“Visual Basic 开发设置”选项,然后单击 **启动 Visual Studio** 按钮,经过数秒钟之后默认环境设置完毕,出现如图 1-2 所示的“起始页”。

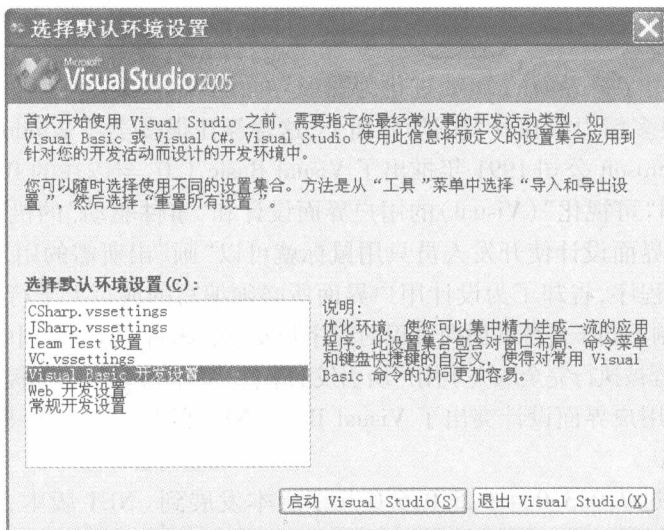


图 1-1 设置默认环境

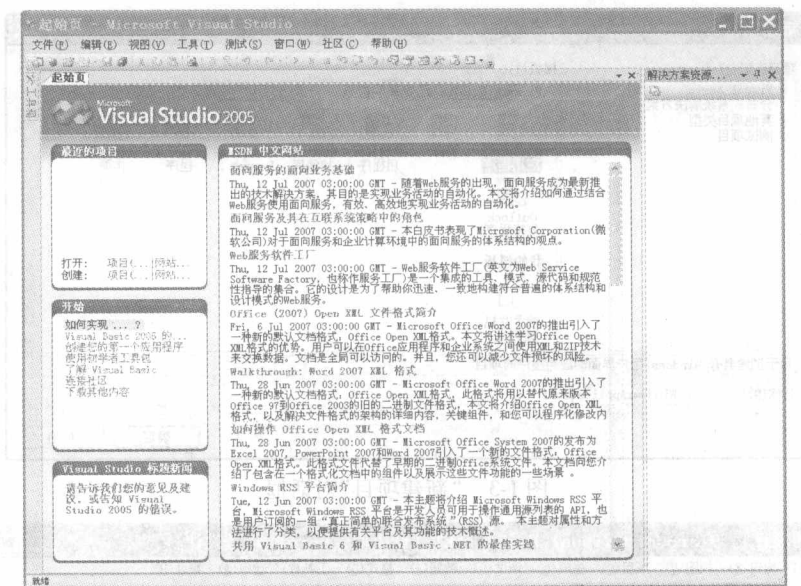




图 1-2 起始页

再次启动 Visual Basic .NET 时,图 1-1 所示窗口不再出现,若要修改曾经设置的默认环境设置,可以使用菜单命令 **工具**→**导入和导出设置**并选择“重置所有设置”单选钮,单击 **下一步** 按钮,在下一级窗口中选择“否,仅重置设置,从而改写我的当前设置”单选钮,继续单击 **下一步** 按钮,在下一级窗口中选择“Visual Basic 开发设置”选项即可。

起始页中的内容主要是一些资源链接,通常要把它关闭。要想在启动 Visual Basic .NET 时不显示起始页,可以使用菜单命令 **工具**→**选项**→**环境**→**启动**,把“启动时”选项设置为“显示空环境”即可。

1.2.2 新建项目

使用 **文件**→**新建**→**项目** 菜单命令或者直接单击工具按钮  会打开如图 1-3 所示的新建项目对话框。

在新建项目对话框中的项目类型窗口选择“Visual Basic”,在模板窗口选择一种程序模板(如“Windows 应用程序”),在名称文本框中输入项目的名称,最后单击 **确定** 按钮,就创建了一个新的项目。此时新建的项目并未保存在磁盘上,要保存项目,单击工具栏上的全部保存按钮 ,在弹出的“保存项目”对话框中重新指定项目的名称和保存的路径,最后单击 **保存** 按钮。

1.2.3 Visual Basic .NET 集成开发环境的构成

(1) Visual Basic .NET 集成开发环境概述

新建项目后出现如图 1-4 所示的用户界面,这就是 Visual Basic .NET 的集成开发环境。在图 1-4 中除了通常的标题栏、菜单栏、工具栏之外,还有以下几节集中介绍开发 Visual Basic .NET 项目必须了解的内容。



图 1-3 “新建项目”对话框

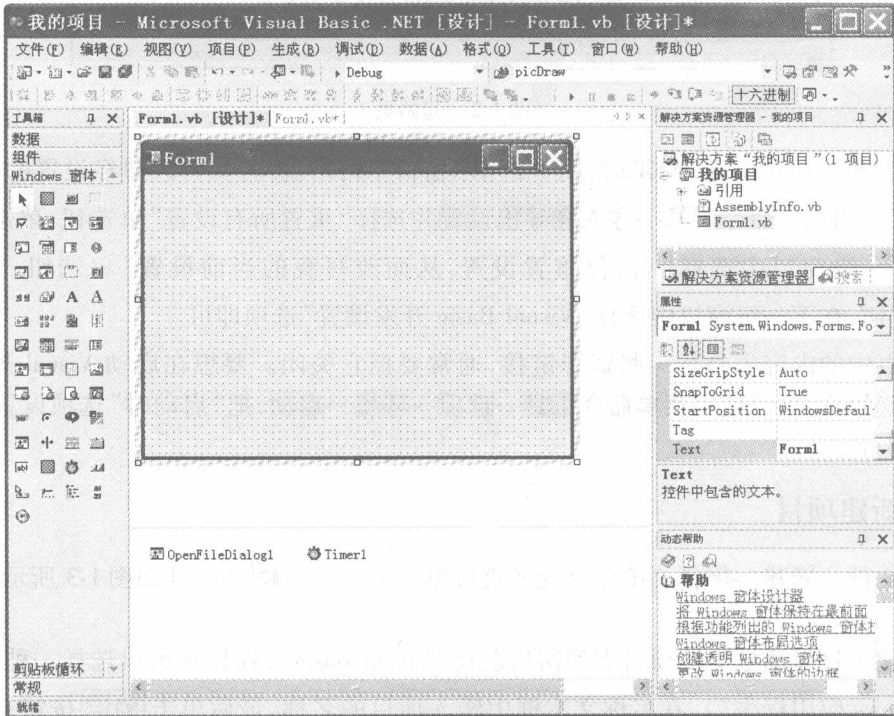



图 1-4 Visual Basic .NET 集成开发环境

(2) 窗体

窗体默认的标题是 Form1。窗体位于集成开发环境的中部称为设计区的区域内的“Form1. vb[设计]”选项卡中,是呈现给用户的用户界面,编程开发阶段在窗体上绘制的内容,就是将来用户看到的。窗体是安放各种控件的容器,设计用户界面的工作在窗体上进行。

打开窗体的常用方法有:

- ◇ 单击解决方案资源管理器中的  工具按钮。
- ◇ 单击集成开发环境中部的“Form1. vb[设计]”选项卡(见图 1-9)。

◇ 双击解决方案资源管理器树形结构的“Form1. vb”。

(3) 组件区

在打开“Form1. vb[设计]”选项卡的同时,位于集成开发环境中下部的组件区同时被打开。组件区用来安放程序使用的组件,无论编程时是否把组件添加到窗体或组件区,组件都会出现在组件区。控件和组件的不同点是,控件会出现在用户界面中,而组件在开发阶段出现在集成开发环境下部的组件区,运行时是看不见的。

(4) 从工具箱添加控件和组件

工具箱通常位于集成开发环境的左侧,通常由“数据”、“组件”、“Windows 窗体”等几部分组成。每一部分中都有很多的图标,这些图标有的称为“控件”,有的称为“组件”。图 1-5 列出了“所有 Windows 窗体”中的控件和组件。其中左图是以图标方式列出的,右图是以列表方式列出的(一部分),在右击工具箱弹出的快捷菜单中选中或取消“列表视图”选项,可以在二者之间切换。



图 1-5 “Windows 窗体”中的控件和组件

把控件或组件添加到项目的窗体中的常用方法有以下两种:

- ◇ 直接双击工具箱中的一个控件,就在窗体上出现该控件,然后调整其大小和位置。
- ◇ 选中工具箱中的一个控件,然后用鼠标在窗体拖动并绘制出控件,并调整其大小和位置。

添加组件的方法与添加控件的方法基本相同,双击或绘制都可以,只是组件出现在组件区中,并且其大小是不可调整的。

以后读者会学习到位于工具箱中的每一个控件和组件都是一个类,而添加到项目中的控件和组件被实例化为一个对象,窗体自身也是一个对象。因此,如果在窗体上总共添加了 3 个控件或组件,则连同窗体在内就创建了 4 个对象。

(5) 调整控件在窗体中的布局

为便于调整控件在窗体中的布局,Visual Basic. NET 在工具栏中提供了布局调整工具。使用布局调整工具之前应先选中两个或两个以上控件,选择多个控件的方法有两种:

- ◇ 先用鼠标单击选中一个控件,然后按住 **Shift** 键或 **Ctrl** 键并单击其他控件。

◇ 框选,用鼠标在窗体上拖动会绘出一个矩形,释放鼠标后所有被框住的控件就一同被选中,如图 1-6 所示。

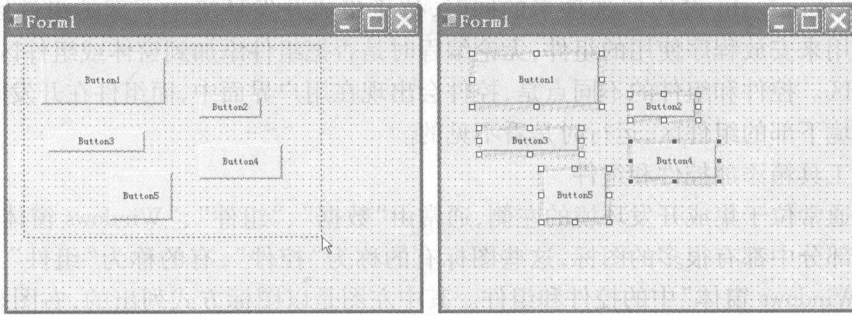


图 1-6 框选多个控件的方法

选中的控件周围有 8 个调整点,当同时选中多个控件时,有一个控件的调整点是实心的正方形,这个控件称为基准控件。当统一调整多个控件的大小和布局时,基准控件的大小和位置是调整的基准。在选中多个控件的情况下,只要单击另一个非基准控件,就会把该控件变为基准控件。

Visual Basic .NET 提供的 23 种调整工具及其说明如表 1-1 所示。

表 1-1 调整工具及其说明

调整工具	说明	调整工具	说明
	对齐到网格		增加各控件的水平间距
	与基准控件左对齐		减小各控件的水平间距
	与基准控件竖直居中对齐		去除各控件的水平间距
	与基准控件右对齐		使各控件竖直间距彼此相等
	与基准控件顶端对齐		增加各控件的竖直间距
	与基准控件水平中间对齐		减小各控件的竖直间距
	与基准控件底部对齐		去除各控件的竖直间距
	使与基准控件的宽度相同		各控件整体在窗体上水平居中
	按网格调整大小		各控件整体在窗体上竖直居中
	使与基准控件的高度相同		把控件置于最上层
	使与基准控件的宽度和高度都相同		把控件置于最下层
	使各控件水平间距彼此相等		

(6) 解决方案资源管理器

“解决方案资源管理器”通常停靠在集成开发环境的右侧,如图 1-7 所示。如果该窗口没有出现,可单击“解决方案资源管理器”选项卡打开。

在解决方案资源管理器中给出了整个项目的树形层次结构。在该窗口的工具栏中有 6 个工具按钮,其中下列 3 个工具按钮十分常用。



图 1-7 解决方案资源管理器

- 1) 打开代码窗口,以便输入、编辑程序代码。
- 2) 打开窗体窗口,以便添加、删除、编辑、调整窗体和控件。
- 3) 打开属性窗口,以便设置属性。

(7) 属性窗口

属性窗口通常与解决方案资源管理器一起停靠在集成开发环境的右侧,如图 1-8 所示。在进行 Visual Basic .NET 编程时会频繁使用属性窗口。如果属性窗口没有出现在集成开发环境中,可单击解决方案资源管理器中的 按钮打开属性窗口。

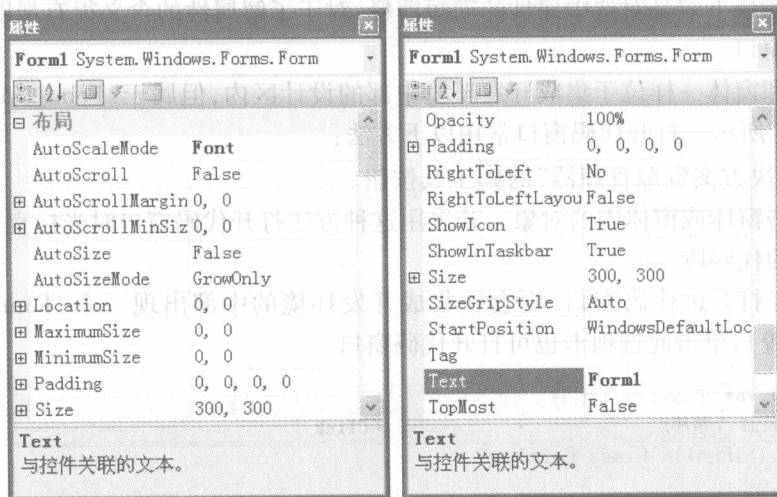


图 1-8 属性窗口的两种排列方式

属性窗口的最上方是一个对象组合框,其中包含本窗体中已经创建的所有对象(包括窗体本身)。当在对象组合框选中一个对象时,下方的属性列表列出的是该对象的属性。因此当要设置一个对象的属性时,应当首先从对象组合框中选择欲设置属性的对象,然后进行属性设置,切忌张冠李戴。


在对象组合框中选定的对象立即在窗体中成为当前对象,并在其周围出现 8 个调整点。同样,当在窗体中选定一个对象时,对象组合框中也自动同步选中该对象。



对象组合框的下一行是工具栏,单击 按钮属性按分类排序,单击 按钮属性以字母顺序排序,但无论采用哪种排序方法,属性列表的内容是相同的。建议采用以字母顺序排序。

属性窗口的中部是属性列表,是该窗口的主要内容,对象属性的设置在这里进行。列表分为两列,左侧的一列是属性名,右侧的一列是属性值。设置属性时先在左边选定欲设置的属性,然后在右侧设置属性的值。

设置属性值时有如下几种情况:

- ◇ 属性值为文本或数值。这种情况下直接输入或编辑文本和数值,Text 属性和 Size 属性都属于这种情况。
- ◇ 属性值为可枚举的列表。这种情况会出现一个下拉按钮,单击该按钮出现可选项,从中选择所需的属性值,Cursor 属性就属于这种情况。逻辑值也是这种情况的一种,对于逻辑值还有更简便的方法,就是直接双击属性名或属性值,属性值就在 True 和 False


之间来回变化,如 Enabled 属性就是这种情况。此外,对于与颜色有关的属性(如 ForeColor 属性),当单击下拉按钮  时会弹出一个调色板,从中选择一种颜色即可。

- ◇ 属性值由对话框选择。当选定某些属性时,会出现一个  按钮,单击此按钮会弹出一个对话框,从对话框中选定所需的属性值并单击  按钮就把属性值写入属性列表中,如 Font 属性和 Icon 属性等属于这种情况。
- ◇ 属性值由示意图选择。当选定某些属性(如 AnchorImage 属性、Align 属性等)时会弹出一个示意图,这种情况只须从示意图中选择一种即可。

属性窗口的最下端是对选中属性的简短解释,对于了解属性的含义很有帮助。

(8) 代码窗口

代码窗口和窗体一样位于集成开发环境中部的设计区内,但属于“Form1.vb”选项卡。代码窗口如图 1-9 所示。打开代码窗口常用以下方法:

- ◇ 单击“解决方案资源管理器”的  工具按钮。
- ◇ 直接双击窗体或窗体中的对象。若采用这种方法打开代码窗口时光标直接定位在所双击对象的代码段。
- ◇ 如果曾经打开过代码窗口,则会在集成开发环境的中部出现一个“Form1.vb”选项卡(见图 1-9),单击此选项卡也可打开代码窗口。

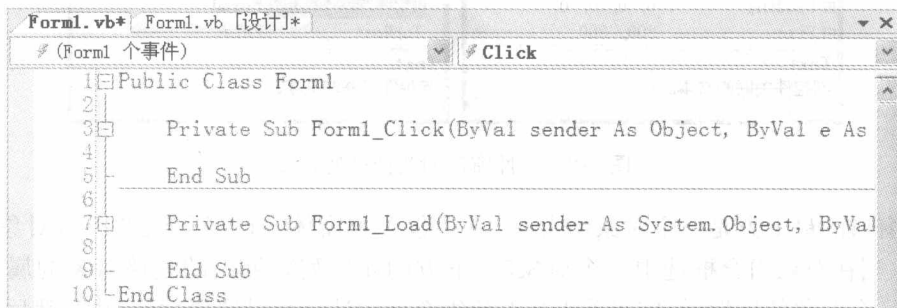


图 1-9 代码窗口

在代码窗口左上方有一个对象组合框,右上方有一个事件组合框。对象组合框中包含本窗体中已经创建的所有对象(包括窗体本身)。当在对象组合框选中一个对象时,右侧的事件组合框中列出的是该对象的所有事件。当编辑程序代码时,应当首先在对象组合框中选择一个对象,在事件组合框中选择一个事件,这时 Visual Basic .NET 自动在代码窗口中书写所选对象所选事件的事件处理程序的框架,例如:

```
Private Sub Form1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Click
.....
End Sub
```

然后在该框架中的空行处编辑程序代码。编程时必须认清是为哪一个对象的哪一个事件编程,切忌张冠李戴。





即使没有输入任何程序代码,打开代码窗口时也会出现如下的程序代码:

```
Public Class Form1
```


End Class

(9) 窗口的停靠与自动隐藏

上述的工具箱通常停靠在集成开发环境的左侧,解决方案资源管理器和属性窗口通常停靠在集成开发环境的右侧,还有其他一些陆续要学习的窗口,如帮助窗口、调试窗口等,也按常规会停靠在集成开发环境的下部或右侧。

每一个窗口的标题栏中都有一个“图钉”或。当图钉呈状态时窗口总是显示的;当图钉呈状态时,如果鼠标位于窗口内,则窗口显示,如果鼠标离开窗口,则窗口被自动隐藏,仅在集成开发环境的停靠侧呈现一个图标或选项卡,只要鼠标移动到这种图标或选项卡之上,隐藏的窗口会自动展开。

如果拖动窗口的标题栏使窗口离开停靠处,就会变成独立的窗口,同时标题栏上的“图钉”图标消失。不提倡这种破坏窗口停靠状态的做法。



1.2.4 使用帮助

学会并经常使用 Visual Basic .NET 的帮助系统对于快速学习 Visual Basic .NET 是十分有益的。Visual Basic .NET 的帮助菜单中集中了下列多种方式。

(1) 动态帮助

执行菜单命令帮助→动态帮助打开动态帮助窗口,如图 1-10 所示。在动态帮助窗口中的每一项都是一个超链接,单击其中的一项就显示帮助的内容。

(2) 目录

执行菜单命令帮助→目录或在动态帮助窗口单击按钮或在索引窗口或搜索窗口单击按钮都能打开如图 1-11 所示的帮助目录窗口。在这里可以按照目录寻求帮助。

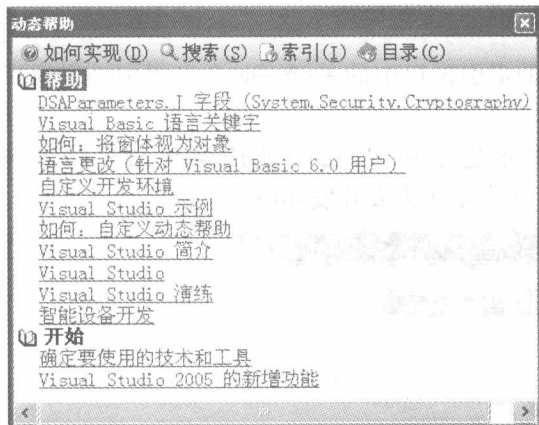




图 1-10 动态帮助窗口





图 1-11 帮助目录窗口

(3) 索引

执行菜单命令帮助→索引或在动态帮助窗口单击按钮或在目录窗口或搜索窗口单击按钮都能打开如图 1-12 所示的帮助索引窗口,使用索引寻求帮助时先在“查找”文本框中

输入要查找的关键字。随着每一个字母或汉字的输入,下面的列表框就会列出匹配的关键字,当所要的关键字出现时,只要单击该关键字,就能在“索引结果”窗口看到帮助的内容。

(4) 搜索

执行菜单命令**帮助**→**搜索**或在动态帮助窗口单击按钮或在目录窗口和索引窗口单击按钮都能打开如图 1-13 所示的搜索帮助窗口。使用搜索寻求帮助时,先在文本框中输入要查找的关键字,并根据搜索需要在右侧选择搜索的位置(如“本地帮助”),然后单击 **搜索** 按钮,就会自动打开“搜索”结果窗口,从中可看到搜索到的符合搜索要求的条目,单击其中的一个条目,就可在新添的选项卡中看到帮助的内容。

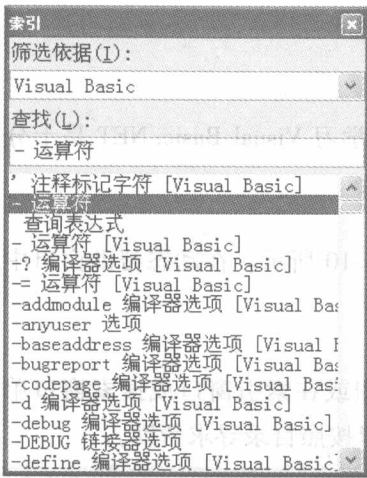


图 1-12 帮助索引窗口

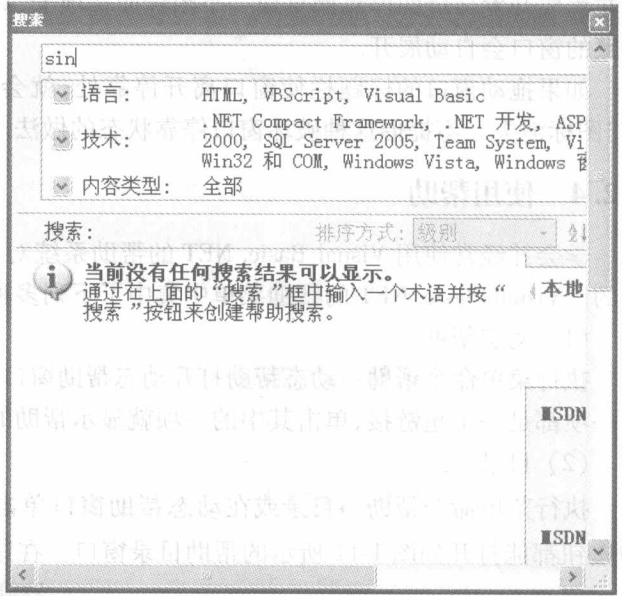


图 1-13 搜索帮助窗口

(5) 索引结果窗口

如果“索引结果”窗口是打开的,则在单击帮助索引窗口中的关键字时会在索引结果窗口显示搜索到的条目。

但是“索引结果”窗口不会自动打开,要显示这个窗口,必须使用菜单命令**帮助**→**索引结果**。“索引结果”窗口如图 1-14 所示,它通常停靠在集成开发环境的下部。

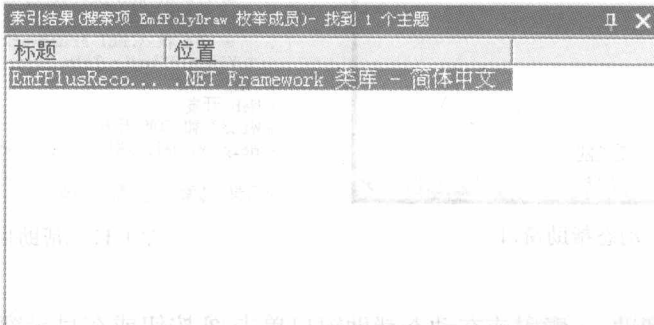


图 1-14 索引结果窗口

当在搜索帮助窗口的“查找”文本框中输入关键字并单击**搜索**按钮时,会自动打开“搜索结果”窗口,其中包含搜索到的条目,“索引结果”窗口如图 1-15 所示。

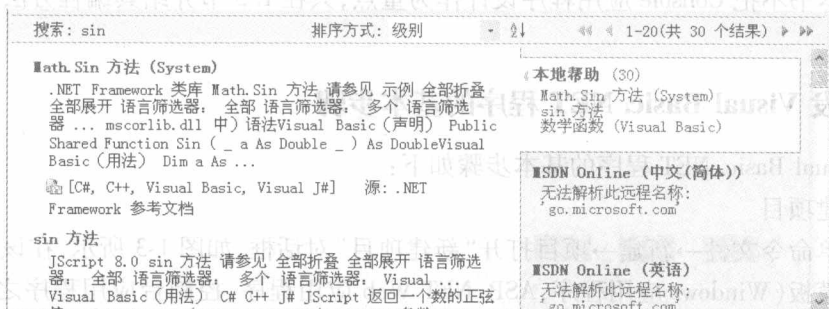


图 1-15 搜索结果窗口

1.3 创建 Visual Basic. NET 程序

1.3.1 Visual Basic. NET 的 3 种标准程序

用 Visual Basic. NET 可以创建 3 种标准程序。

(1) Windows 应用程序

Windows 应用程序最终编译成 .EXE 文件,并可在 Windows 操作系统环境下直接运行。

创建 Windows 应用程序的方法是使用工具栏上的 **新建** 按钮或执行菜单命令 **文件** → **新建** → **项目** 打开如图 1-3 所示“新建项目”对话框,“项目类型”选择“Visual Basic”,“模板”选择“Windows 应用程序”,在“名称”文本框中输入项目的名称,最后单击 **确定** 按钮,就创建了一个新的项目,同时打开集成开发环境。

本书重点介绍 Windows 应用程序的编程,而且 Windows 应用程序编程是进一步学习 Web 程序编程的基础。

(2) Web 应用程序

Web 应用程序是运行于浏览器中的 Visual Basic. NET 应用程序。创建 Web 应用程序的方法、在 Web 应用程序中添加控件的方法、书写代码的方法等都与 Windows 应用程序有很多相似之处。这一特性使得用户创建 Web 应用程序就像创建 Windows 应用程序一样方便。学习了 Windows 应用程序设计以后,会很容易转到 Web 应用程序的设计上来。创建 Web 应用程序的具体方法将在第 14 章中介绍。

(3) Console 应用程序

Console(控制台)应用程序是指 Windows 环境中基于字符界面的标准输入输出接口的程序。Console 应用程序没有图形用户界面,输出的字符直接显示在 DOS 窗口内,并接受键盘的输入。

Console 应用程序没有窗体,代码直接书写在标准模块中,程序运行时首先执行名为 Main

的过程。Console 应用程序最终被编译成 .EXE 文件,在资源管理器中双击这类 .EXE 文件会打开一个 DOS 窗口,并在该 DOS 窗口中运行程序。

相对于图形用户界面的 Windows 应用程序而言,Console 应用程序适用于交互过程比较简单的程序。本书不把 Console 应用程序设计作为重点,只在 1.5 节介绍其编程方法并给出一个实例。

1.3.2 开发 Visual Basic. NET 程序的基本步骤

开发 Visual Basic. NET 程序的基本步骤如下:

(1) 新建项目

使用菜单命令文件→新建→项目打开“新建项目”对话框,如图 1-3 所示,在该对话框中选择一种程序模板(Windows 应用程序,ASP. NET Web 应用程序,控制台应用程序之一),在“名称”文本框中输入项目名称,最后单击确定按钮,就创建了一个新的项目。

(2) 设计用户界面

对于 Windows 应用程序和 ASP. NET Web 应用程序,第二步是在窗体上添加控件,设计好用户界面。

(3) 设置对象的属性

在程序的设计阶段一般要在属性窗口设置对象的属性。虽然利用代码也能设置对象的属性,但会增加代码的长度。一般情况下,是在属性窗口中设置那些程序运行时始终不变的属性或属性的初始值。而那些属性值时常变化的属性往往使用代码设置。

(4) 编写代码

打开代码窗口,在左上方的对象组合框中选定一个对象,在右上方的事件组合框中选定一个事件。这时 Visual Basic. NET 自动生成所选对象、选事件的事件处理程序的框架,在中间的空行处输入事件处理程序。

如果要定义窗体模块级的变量、常量或输入自定义的过程、函数,应当紧接 Public Class Form1 的下一行输入;如果要继承其他类,应在紧接 Public Class Form1 的下一行使用 Inherits 关键字声明,定义窗体模块级的变量、常量或自定义的过程、函数则应放在在 Inherits 语句之后。

(5) 编译、运行、调试程序

单击工具栏上的按钮或使用快捷键 F5 启动程序,这时 Visual Basic. NET 先行把该项目进行编译,如果没有错误发生,就会在保存项目文件夹的 bin 子文件夹中生成 .EXE 文件,并运行刚刚生成的这个 .EXE 文件。

如果没有达到预定的运行效果或出现运行时错误和异常,应当修改用户界面设计或程序代码直到效果满意为止。

如果在编译过程中有错误发生,Visual Basic. NET 会在集成开发环境的下部打开一个“错误列表”窗口,其中列出了发生错误的种类、所在的文件和在文件夹中的行号。应当根据这些错误信息修改程序代码并重新编译运行,直到通过编译正常运行为止。有一些难以排除的错误需要对程序进行调试才能发现。

(6) 制作安装程序

项目开发完成后要制作安装程序以便发布。制作安装程序的目的是使程序能够安装到其

他计算机上后能正常运行,如果不通过安装程序进行发布,只是简单地把编译生成的 .EXE 文件复制到其他计算机上,很可能由于目标计算机上缺少某些支持文件导致程序无法运行。

1.3.3 第一个 Visual Basic .NET Windows 应用程序

本节开发第一个 Visual Basic .NET Windows 应用程序,其目的是增加感性认识,学习 Windows 应用程序编程的步骤和方法。本节对程序的细节不做过多的介绍。随着学习的一步深入,将来回过头来看这个程序是极其简单的。

实例:加法计算器

程序构思:

本例设计一个加法计算器,功能很简单:窗体上有 3 个文本框,分别显示被加数、加数、和,窗体上还有 3 个按钮,分别是 **计算**、**清除**、**结束**。运行程序时先在被加数和加数文本框内输入两个数,然后单击 **计算** 按钮就计算出二者的和并显示在“和”文本框中。单击 **清除** 按钮可以清除 3 个文本框中的内容,单击 **结束** 按钮程序结束。

编程步骤:

1) 设计用户界面。新建一个 Windows 应用程序项目,项目名称为“加法计算器”。打开窗体,在窗体上添加 3 个 Label (标签)控件,3 个 TextBox(文本框)控件和 3 个 Button 控件(按钮),按照图 1-16 调整好各控件的大小和在窗体上的布局。

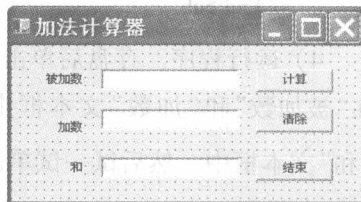


图 1-16 加法计算器窗体设计

2) 设置各个对象的属性。打开属性窗口,按照表 1-2 设置各对象的属性,凡是表中未列出的属性均取默认值。

表 1-2 加法计算器属性设置

对象	属性	属性值
Form1	Text	加法计算器
Label1	Text	被加数
	TextAlign	MiddleRight
Label2	Text	加数
	TextAlign	MiddleRight
Label3	Text	和
	TextAlign	MiddleRight
Text1	Text	空白
Text2	Text	
Text2	Text	
Button1	Text	计算
Button2	Text	清除
Button3	Text	结束

3) 输入程序代码。打开代码窗口,输入各事件处理程序的代码:





```

Private Sub Button1_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    TextBox3.Text = Val(TextBox1.Text) + Val(TextBox2.Text)
End Sub

'-----
Private Sub Button2_Click (ByVal sender As Object, ByVal e As System.EventArgs) Handles Button2.Click
    TextBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
End Sub

'-----
Private Sub Button3_Click (ByVal sender As Object, ByVal e As System.EventArgs) Handles Button3.Click
    End
End Sub

```


4) 运行程序。存盘后单击工具栏中的  按钮运行程序, 出现如图 1-17 所示的用户界面。在“被加数”和“加数”文本框中任意输入两个数, 单击  按钮, 就把计算的结果显示在“和”文本框中。然后试一试单击  按钮是否能清除各文本框中的内容, 单击  按钮能否结束程序。

要点解释:


从本例可以看出, 单击不同的按钮引发不同的事件, 从而调用不同的事件处理程序完成不同的功能。这就是 Visual Basic .NET 的事件驱动的机制。

实例: 改进加法计算器

程序构思:

为了进一步说明事件驱动机制, 对加法计算器进行改进, 删除窗体上的  按钮, 反而使程序运行得更好。

编程步骤:

- 1) 设计用户界面。打开“加法计算器”项目, 把窗体上的  按钮(即 Button1)删除。
- 2) 设置各个对象的属性。只把窗体的 Text 属性值修改为“改进加法计算器”, 其余对象的属性设置不变。这时的用户界面如图 1-18 所示。

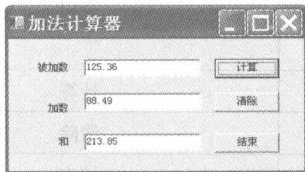


图 1-17 加法计算器的结果

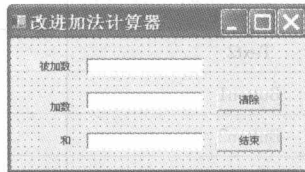


图 1-18 改进加法计算器窗体设计

- 3) 输入程序代码。打开代码窗口, 把各事件处理程序的代码修改为:


```
Private Sub Button2_Click (ByVal sender As Object, ByVal e As System.EventArgs) Handles Button2.Click
    TextBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
End Sub
```

```
Private Sub Button3_Click (ByVal sender As Object, ByVal e As System.EventArgs) Handles Button3.Click
    End
End Sub
```

```
Private Sub TextBox1_TextChanged (ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged
    TextBox3.Text = Val(TextBox1.Text) + Val(TextBox2.Text)
End Sub
```

```
Private Sub TextBox2_TextChanged (ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox2.TextChanged
    TextBox3.Text = Val(TextBox1.Text) + Val(TextBox2.Text)
End Sub
```

```
Private Sub TextBox3_TextChanged (ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox3.TextChanged
    TextBox3.Text = Val(TextBox1.Text) + Val(TextBox2.Text)
End Sub
```

4) 运行程序。存盘后单击工具栏中的  按钮运行程序，出现如图 1-19 所示的用户界面。与改进前不同的是，随着输入被加数和加数，即时地计算出结果并显示在“和”文本框内，省却了用户必须单击 **计算** 按钮的麻烦，使用起来更加方便。即使用户恶意地把“和”修改为错误的数值，程序立即把它改回到正确的结果。

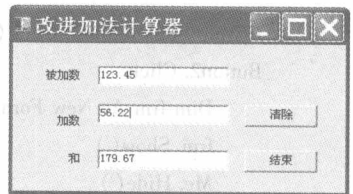


图 1-19 改进加法计算器运行效果

要点解释：

本例把上例 Button1 对象的 Click 事件处理程序移到 3 个文本框的 TextChanged 事件处理程序程序中，只要用户改变了这 3 个文本框中任何一个的内容，都会引起相应的 TextChanged 事件，立即调用事件处理程序计算出结果并送 TextBox3 显示。

实例：使用多个窗体

程序构思：

在本例中共有 3 个窗体，从 Form1 可以以不同的方法打开 Form2 或 Form3。

编程步骤：

1) 设计用户界面。新建一个 Windows 应用程序项目，项目名称为“使用多个窗体”，这时

已经自动地创建了一个 Form1 窗体,使用菜单命令 项目→添加 Windows 窗体为项目添加窗体 Form2 和 Form3。

打开 Form1 窗体,添加两个按钮控件。按照图 1-20 所示,调整好各控件的大小和在窗体上的布局。窗体 Form2 和 Form3 上没有任何控件。

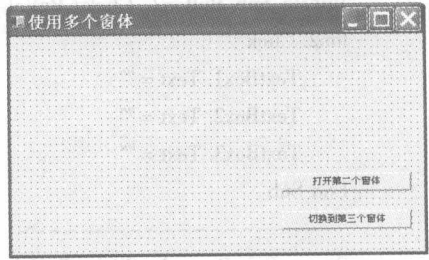


图 1-20 “使用多个窗体”项目的 Form1 窗体

2) 设置各个对象的属性。打开属性窗口,按照表 1-3 设置各对象的属性,凡是表中未列出的属性均取默认值。

表 1-3 “使用多个窗体”属性设置

对象	属性	属性值
Form1	Text	使用多个窗体
Form1 的 Button1	Text	打开第二个窗体
Form1 的 Button2	Text	切换到第三个窗体
Form2	Text	第二个窗体
Form3	Text	第三个窗体


3) 输入程序代码。打开 Form1 代码窗口,输入各事件处理程序的代码:

```
Private Sub Button1_Click ( ByVal sender As System. Object, ByVal e As System. EventArgs ) Handles
Button1. Click
    Dim frm As New Form2
    frm. Show()
End Sub
' -----
Private Sub Button2_Click ( ByVal sender As System. Object, ByVal e As System. EventArgs ) Handles
Button2. Click
    Dim frm As New Form3
    frm. Show()
    Me. Hide()
End Sub
' -----
Private Sub Form1_Closed ( ByVal sender As Object, ByVal e As System. EventArgs ) Handles My-
Base. Closed
    End
End Sub
```

在 Form2 的代码窗口不用输入任何代码,在 Form3 的代码窗口输入事件处理程序的代码:

```
Private Sub Form3_Closed ( ByVal sender As Object, ByVal e As System. EventArgs ) Handles My-
Base. Closed
    Dim frm As New Form1
    frm. Show()
```


End Sub

4) 运行程序。存盘后单击工具栏中的  按钮运行程序,首先出现如图 1-21 所示的 Form1。这时如果单击 **打开第二个窗体** 按钮,就会在保持 Form1 不变的情况下打开 Form2,并且 Form2 能够单独关闭。如果单击 **切换到第三个窗体** 按钮,就会在隐藏 Form1 窗体的同时打开 Form3 窗体,当关闭 Form3 时重新显示 Form1。当关闭 Form1 时,如果 Form2 处于显示状态,也一同被关闭。

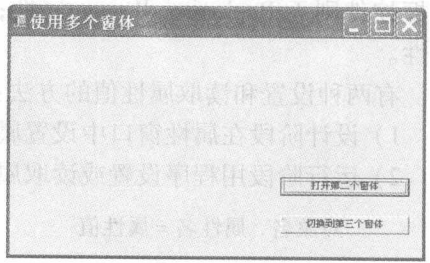




图 1-21 “使用多个窗体”的运行效果

要点解释:

1) 显示窗体用窗体的 Show 方法,隐藏窗体用窗体的 Hide 方法。

2) 通常使用窗体的  按钮关闭程序。实际上,使用  按钮只是关闭了窗体,要结束程序必须使用 End 语句。本例中如果省略了 Form1_Closed 事件处理程序中的 End 语句,则关闭 Form1 时 Form2 仍然显示着(如果 Form2 正在显示的话)。

1.4 Visual Basic. NET 的核心概念——对象、属性、事件和方法

对象、属性、事件和方法是 Visual Basic. NET 中几个最基本的和最重要的概念(在第 11 章中还要学习与“类”有关的一些重要概念),要学好 Visual Basic. NET 编程,就必须深入理解这些概念。通过前面的几个实例已经有了足够多的感性认识,在本节以下的几个小节中深入介绍这些概念。

1.4.1 对象

对象的原文为 Object,而 Object 既可译为对象又可译为物体、目标,所以 Object 可以被理解为一个“物体”、“物件”。具体说来,在 Visual Basic. NET 中对象包括窗体、控件、组件等。例如,窗体对象、按钮控件对象、文本框控件对象、定时器组件对象等等都是实实在在的“物件”,都是 Visual Basic. NET 的对象。

每一个对象都包括 3 个成员,即属性、事件和方法。所以,可以把对象看作封装了属性、事件和方法的整体。新建一个对象时就同时拥有了这些成员,同时也拥有了对这些成员操作的代码和对象可实现动作(方法)的代码。

对象是类的实例,而类是对象的基础。例如,“人”是一个类,具体的人“张三”就是“人”这个类的一个实例,即“张三”是一个对象。就 Visual Basic. NET 而言,工具箱中的 Button 控件是一个类,而实例化为窗体上的一个 Button 则是一个对象。关于类以及类与对象之间的关系将在第 11 章详细叙述。

1.4.2 属性

属性(Property)是对象的特性。例如把一个具体的人“张三”看作是一个对象,则年龄、身高、性别、学历等都是该对象区别于其他对象的特性,即这些都是这个对象的属性。每一