

VB.NET 精彩实例教程

精英科技 编著

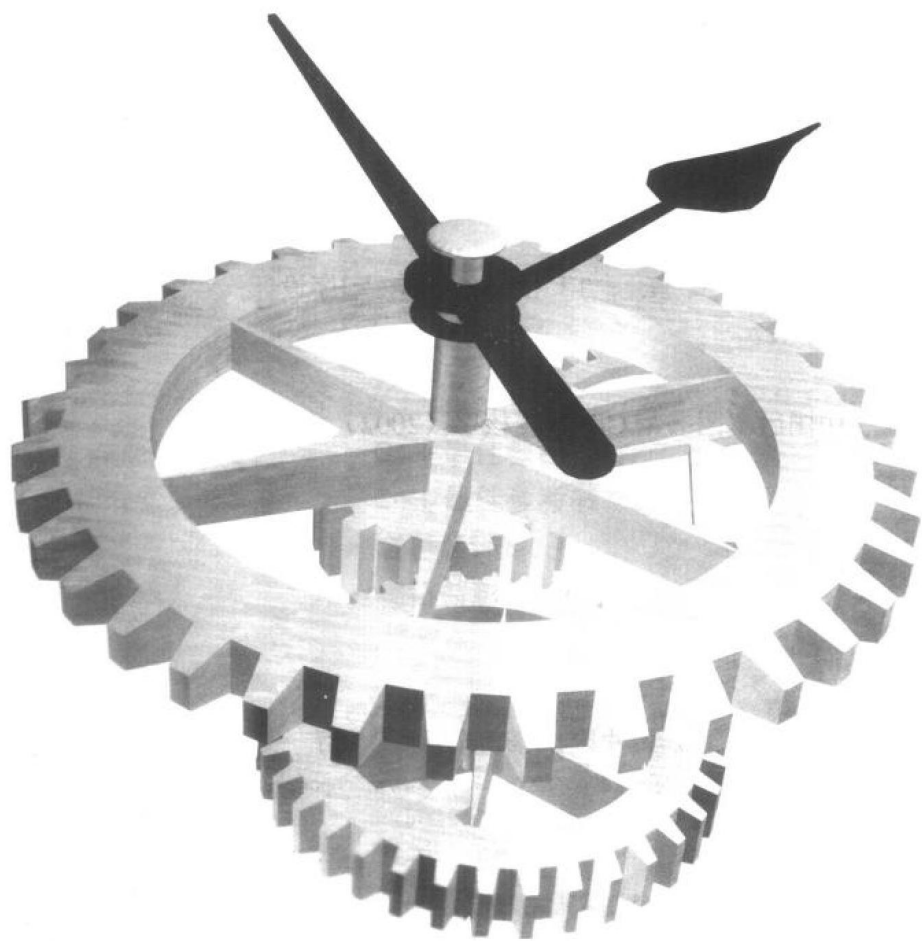


中国电力出版社

www.infopower.com.cn

VB.NET 精彩实例教程

精英科技 编著



中国电力出版社

内 容 提 要

本书首先对 Visual Basic .NET 的编程界面做了简单介绍, 并介绍了如何将 Visual Basic 6.0 的程序升级到 Visual Basic .NET 程序; 然后讲述 Visual Basic .NET 的语言特点和控件编程基础, 并介绍了常用控件的应用技巧。最后通过 Visual Basic .NET 控件编程的一系列经典实例, 深入系统地介绍了使用 Visual Basic .NET 控件技术的各项细节和应用技巧。本书在立足 .NET 自身控件类的基础上, 还介绍了一些外部控件的应用方法和开发技巧。

本书既适用于 Visual Basic .NET 的初学者, 又适用于有一定编程基础的开发人员, 也可作培训教材使用。

图书在版编目 (CIP) 数据

VB .NET 精彩实例教程/精英科技编著. -北京: 中国电力出版社, 2002. 5

ISBN 7-5083-1067-5

I . V . . . II . 精 . . . III . BASIC 语言-程序设计
IV . TP312

中国版本图书馆 CIP 数据核字 (2002) 第 032896 号

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://www.infopower.com.cn>)

汇鑫印务有限公司印刷

各地新华书店经售

*

2002 年 7 月第一版 2002 年 7 月北京第一次印刷

787 毫米×1092 毫米 16 开本 18.5 印张 418 千字

定价 26.00 元

版 权 所 有 翻 印 必 究

(本书如有印装质量问题, 我社发行部负责退换)

前 言

Microsoft Visual Basic .NET 是 Microsoft Visual Basic 的后续版本，是一套强大的编程工具。它基于 .NET 框架重新设计，用户可以用它来轻松地创建用于 Microsoft Windows 操作系统和 Web 的下一代应用程序。使用 Visual Basic .NET 使可视化开发 Web 应用程序、Web 服务、Windows 应用程序和服务器端组件将变得轻而易举。此外，由于 Visual Basic .NET 使用了 Windows 应用程序的 XCOPY 部署方案，使用户不必再提心 DLL 文件的版本问题，这就比 Visual Basic 6.0 进步了许多。所以，随着 Visual Basic .NET 的发布，困扰 Visual Basic 的“DLL 噩梦”将成为过去，Basic 语言的简单方便性得到了进一步的强化，模块化控件编程的特性更加突出，这使得 Visual Basic .NET 成为了计算机程序员的首选开发工具。

全书共分 10 章，第 1 章简单介绍了 Visual Basic .NET 的编程界面，同时介绍了如何将 Visual Basic 6.0 的程序升级到 Visual Basic .NET 程序，重点突出了 Visual Basic .NET 与 Visual Basic 6.0 的不同之处和 Visual Basic .NET 的先进性；第 2 章讲述 Visual Basic .NET 的语言特点和相关的控件编程基础；第 3 章则结合实例介绍常用控件的编程技巧，这一章从方法学的角度概括了 Visual Basic .NET 的控件编程技术，读者一定要好好研读。随后的每个章节均通过一个乃至几个控件编程的精彩实例深入讲述 Visual Basic .NET 控件编程的应用，比如计算器、通信录、FTP 客户端等。本书在立足 .NET 自身控件类的基础上，还介绍了一些外部控件的应用方法和开发技巧，这对提高读者快速开发应用程序有着极大的帮助。同时，在本书的规则和写作的过程中，按照如下的结构来规划每个实例单元，这种结构便于读者融会贯通地学习知识和把握相关实例的应用技巧。

- 实例的功能规划和运行效果预览
- 介绍相关控件知识
- 实例的制作过程
- 学习实例的总结和提高

本书既适用于初学者又适合有一定编程语言基础的开发人员。有 Visual Basic 基础的读者可以快速翻阅前几章，但是我们并不建议跳过这些章节。因为从 Visual Basic 到 Visual Basic .NET 微软做了大量细致的工作，尽量使语言更加通俗易懂。另外，Visual Basic .NET 并不仅仅是 Visual Basic 的一个升级版本，而是一个飞跃。其中的许多语法规则已经和 Visual Basic 有了很大的差别。

本书与其他有关 Visual Basic 的同类书籍相比较，它对 Visual Basic .NET 的控件编程方面介绍更为全面、知识面更为广泛，尤其是可操作性和实用性很强，对初学者来说是一本上选的入门教程。因此在编写本书时，从方法入手，对 Visual Basic .NET 进行比较全面地介绍，

FA62515

力图将最新、最全面、最实用的控件编程技术介绍给读者。通过本书的学习，可以快速掌握 Visual Basic .NET 这一优秀的开发软件，而且也今后深入地研究该软件，奠定了坚实的基础。

由于时间和作者水平所限，书中难免存在不足之处，还希望广大读者给予指正。

作者

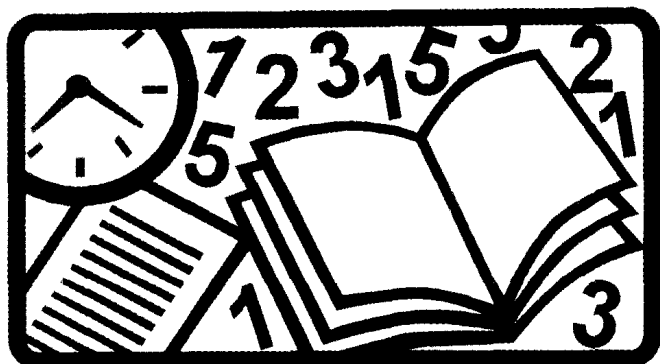
2002年5月

目 录

前 言

第 1 章	第一次接触 Visual Basic .NET	1
1.1	了解和熟悉 Visual Basic .NET	1
1.2	.NET 控件编程的变化和区别	10
1.3	升级 Visual Basic 6.0 的工程	22
第 2 章	Visual Basic 控件编程基础	24
2.1	面向对象的程序设计	24
2.2	窗体程序设计	32
2.3	Visual Basic 6.0 到 Visual Basic .NET 的语言变化总结	39
2.4	控件编程实例入门	48
第 3 章	常用控件应用说明	55
3.1	基本控件介绍	55
3.2	选择性控件介绍	64
3.3	焦点概述	74
3.4	软件热键（快捷键）	78
3.5	菜单设计	80
3.6	Visual Basic .NET 编码规范	91
第 4 章	Visual Basic .NET 程序设计流程	95
4.1	数组、自定义函数与过程	95
4.2	一个记录选票的程序	100
4.3	计算器编程实例	103
4.4	心理测试程序	122
第 5 章	图形绘制程序的设计	127
5.1	PictureBox 控件介绍	127
5.2	利用 PictureBox 控件绘制简单图形	134
5.3	图形绘制程序设计	141
5.4	总结	158

第 6 章	图像浏览程序的设计	164
6.1	使用 PictureBox 处理图像文件	164
6.2	旋转图片工具制作实例	171
6.3	图片自动演示程序	177
6.4	总结	181
第 7 章	多媒体控件	182
7.1	ActiveX 技术介绍	182
7.2	使用 Windows Media Player 播放文件示例	188
第 8 章	文件管理和数据库应用	206
8.1	文件监控器	207
8.2	FTP 文件管理器	218
8.3	通信录	228
第 9 章	Visual Basic .NET 网络控件应用	254
9.1	网络编程概述	254
9.2	FTP 客户端的开发	260
第 10 章	帮助文件的制作和调用方法	281
10.1	Windows Help Designer 的使用简介	281
10.2	帮助文件的制作	284



第 1 章

第一次接触

Visual Basic .NET

了解和熟悉 Visual Basic .NET
Visual Basic .NET 的全新特点和更新
升级 Visual Basic 6.0 的工程

内容简介:

本章将介绍 Visual Basic .NET 的基本环境。并将 Visual Basic .NET 与 Visual Basic 6.0 进行比较,指出 Visual Studio .NET 最后给出升级的方法和注意事项。Visual Basic .NET 采用了共同运行时,与 .NET 的其他语言相协调,统一了底层运行代码与高级编程风格。因此,Visual Basic .NET 比 Visual Basic 6.0 有着较显著的改进。

1.1 了解和熟悉 Visual Basic .NET

1.1.1 了解全新的 .NET 框架

Microsoft Visual Basic .NET 是 Microsoft Visual Basic 的后续版本,它是基于 .NET 框架重新设计的,用户可以用它来轻松地创建用于 Microsoft Windows 操作系统和 Web 的下一代应用程序。使用 Visual Basic .NET,可视化开发 Web 应用程序、Web 服务、Windows 应用程序和服务端组件将变得轻而易举。此外,Visual Basic .NET 使用了 Windows 应用程序的 XCOPY 部署方案,使用户不必再担心 DLL 版本问题。随着 Visual Basic .NET 的发布,“DLL 噩梦”将成为过去。

微软的 .NET 技术框架是一个革命性的新平台,它建立在开放的 Internet 协议和标准之上,采用许多新的工具和服务用于计算和通信,提供了许多基于 Web 的服务,而且更加易于使用,这使得多种语言之间以及网络机群之间的基于组件的交互访问更加方便。



.NET 技术框架包含以下三个主要部分：公共语言运行时（Common Language Runtime, CLR）、具有多层次结构的统一的类库集合 ADO .NET 技术和 ASP .NET 技术。

1. 公共语言运行时

公共语言规范（Common Language Specification, CLS）不能准确反映它的全部功能。实际上，公共语言运行时在组件的开发过程中以及软件的运行过程中，都扮演着非常重要的角色。在组件运行过程中，运行时负责管理内存分配、启动或取消线程和进程、实施安全性策略、同时满足当前组件对其他组件的需求。在开发阶段，运行时的作用有些变化：与现今的 COM 相比，运行时的自动化程度大为提高（比如可自动执行内存管理），因而开发人员的工作变得非常轻松。尤其是映射功能将使代码编写量锐减，这些代码是开发人员在将业务逻辑转化成可重复使用的组件进行编程时所必需的。

对编程语言而言，运行时这个概念并不新奇：实际上每种编程语言都有自己的运行时。Visual Basic 开发系统具有最为明显的运行时（名为 VBRUN），Visual C++ 跟 Visual FoxPro、Jscript、SmallTalk、Perl、Python 和 Java 一样，有一个运行时 MSVCRT。 .NET 框架的关键作用是它提供了一个跨编程语言的统一的编程环境，这也是它能独树一帜的根本原因所在。CLS 和相关技术所体现的思想，以及和当前已经实现的技术使得符合该规范的语言编写的程序，可以在 .NET Framework 上无缝地集成。

从技术上讲，符合该规范的语言和编译器，将可以把源代码编译成 CLR 所能识别的“微软中间语言”（Microsoft Intermediate Language, MSIL）和“元数据”（metadata）。由于使用统一的运行时和符合规范的类库（不要忘了，除了包括 .NET Class Library 外，还可能包括其他厂商提供的类库，条件只是符合规范。在 .NET SDK beta 2 中“符合 CLS 规范”称为 CLS-compliant），不同语言代码经过符合规范的编译器编译成的 IL 代码是一致的，从而是可相互操作的。IL 代码和相应的 metadata 在运行时，经由 CLR 的 JIT（Just In Time, 适时）编译器编译成在特定平台上运行的机器码并最终运行。

通过这样的机制，.NET 框架具有了支持几乎所有语言的互操作的特性，只要为该语言实现了可以将源代码编译成 MSIL 代码的编译器。在本书撰稿时，.NET SDK 已经提供了 C#、Managed Extensions for C++（一个 C++ 的针对 .NET 的扩展，使之可以和 .NET Framework 集成起来）、Visual Basic .NET 的编译器。在不久的将来，会有其他语言的 .NET 编译器出现（例如 Cobol、Perl），使得各种语言的程序员可以更方便地为 .NET 平台开发组件和应用程序。在本书编写时，已经有 APL、COBOL、Component Pascal、Eiffel、Perl、Python、SmallTalk、Standard ML 等语言的 .NET 开发工具发布。

另外，这种机制也带来了类似于 Java 的平台无关性。当然这种平台无关性是“潜在（potential）”的。 .NET 会以 Windows 为目标的运行平台。但仅从技术上讲，只要在除 Windows 之外的平台上实现了 CLR 和 Class Library，就可以使 .NET 和基于 .NET 的应用程序在该平台上运行。这种思路从市场的角度看也是有可能的。在编写本书的时候，著名的开放源码 GNU 已经启动了一个名为 Mono 的计划，该计划将会为其他非 Windows 平台构建一个与 .NET



在很大程度上兼容的平台无关的框架和运行环境，以免 .NET 这种伟大的设计思想被垄断在少数公司手里。同时，CLR 和 Java 虚拟机一样也是一个运行时环境，它负责资源管理（内存分配和垃圾收集），并保证应用和底层操作系统之间必要的分离。

为了提高平台的可靠性，以及为了达到面向事务的电子商务应用所要求的稳定性级别，CLR 还要负责其他一些任务，比如监视程序的运行。按照 .NET 的说法，在 CLR 监视之下运行的程序属于“受管理的”（managed）代码，而在 CLR 之下、直接在裸机上运行的应用程序或者组件属于“非受管理的”（unmanaged）代码。

CLR 将监视形形色色的常见编程错误，许多年来这些错误一直是软件故障的主要根源。其中包括：访问数组元素越界、访问未分配的内存空间、由于数据体积过大而导致的内存溢出等等。

然而，这种对受管理代码的运行监视是有代价的。虽然当前还不可能精确地得到监视程序运行所需要的开销，但从当前 Beta 测试版的性能表现来看，正如 Microsoft 所承认的那样，我们可以预料由它导致的性能降低程度至少达到 10%。当然，如果监视程序运行能够将稳定性和可用性提高到一个新的档次，我们可以怀疑 10% 的性能降低是否还可以称为一件好事。

在处理器性能改善方面，摩尔定律已经一再被证明是正确的。因此，从系统的安全性和健壮性考虑使用受管代码技术是利大于弊的。

2. 具有多层次结构的统一的类库集合

.NET 框架中的类为开发人员提供了一个统一的、面向对象的、层次化的和可扩展的类库集 (API)。现今，C++ 开发人员使用的是微软基础类库，Java 开发人员使用的是 Windows 基础类库，而 Visual Basic 用户使用的又是 Visual Basic API 集。简而言之，.NET 框架统一了微软当前各种不同的框架。这样，开发人员不再需要学习多种框架就能顺利编程。远不局限于此的是，通过创建一个公共的跨编程语言的 API 集，.NET 框架可实现跨语言继承性、错误处理功能和调试功能。实际上，从 Jscript 到 C++ 的所有编程语言，只要符合 Common Language Specification，那么都可以用来实现 .Net 应用。开发人员可以自由选择理想的编程语言。

3. 革新的 ADO .NET 数据库技术

ADO .NET 是在 ADO 的基础上发展起来的，但又和 ADO 有很大区别，最重要的一点区别是 ADO .NET 遵循更通用的原则，不再专门面向数据库。ADO .NET 集合了所有允许数据处理的类，这些类中拥有典型数据库功能（如索引、排序和视图）的数据容器对象。可以说 ADO .NET 是 .NET 数据库应用程序的权威解决方案。尽管如此，但从总体设计上来看，它不像 ADO 模型那样以数据库为中心，这是 ADO .NET 的一大特点。

同时 ADO .NET 又保持着与以前的 ADO 模型有关的一些主要概念，不过这些概念在 ADO .NET 中都已经得到了更好的完善，并且将不同的数据来源统一成同一种结构化的数据（一个基于 .NET 平台的文本文件），然后从数据库管理系统获得的相关数据或者是分级的 XML 数据。可以说在 ADO .NET 中所有数据库连接都按照一个相容的、标准化的设计模型来执行，这就使得数据处理简单得多。



4. ASP .NET 技术

ASP .NET 是使用 .NET 框架提供的类库构建而成的，它提供了一个 Web 应用程序模型，该模型由一组控件和一个基本结构组成。有了它，Web 应用程序的构建变得非常容易。开发人员可以直接使用 ASP+控件集，该控件集封装了公共的、用于超文本标识语言（HTML）用户界面的各种小组件（诸如文本框、下拉菜单等）。实际上，这些控件运行在 Web 服务器上，它们将用户界面转换成 HTML 格式后再发送给浏览器。在服务器上，控件负责将面向对象的编程模型呈现给 Web 开发人员，这种编程模型能提供面向对象的编程技术拥有的丰富功能。ASP+还提供一些基本结构服务（诸如会话状态管理和进程循环），这些服务进一步减少了开发人员要编写的代码量，并使应用程序的可靠性得到了大幅度提高。ASP+还允许开发人员将软件作为一项服务进行传送。通过使用 ASP+ Web 服务功能，ASP+开发人员只需进行简单的业务逻辑编程，而由 ASP+基本结构负责通过 SOAP 传送服务。

ASP .NET 是建立在公共语言运行时（CLR）库上的应用程序框架。它用来在服务器端构建功能强大的 Web 应用程序。ASP .NET 提供了以下几个超越以前 Web 开发模式的优点：

- 增强的性能。ASP .NET 是运行在服务器端的编译后的 CLR 代码，而不是像 ASP 那样解释执行。ASP .NET 利用提前绑定、即时编译、本地优化和缓存服务来提高性能。所有这一切，性能远远大于你以往写的每一行代码。
- 世界级水平的开发工具支持。在 Visual Studio .NET 的集成开发环境（IDE）中，ASP .NET 框架由丰富的工具箱和设计器组成。所见即所得的（What You See Is What You Get, WYSIWYG）编辑方式、拖放服务器控件、以及自动部署，仅仅是这一强大工具所提供的一少部分功能。
- 强大而富有弹性。由于 ASP .NET 是基于 CLR 的，因此整个 .NET 平台的强大和富有弹性，同样可以应用于 Web 应用程序开发者。.NET 框架的类库、消息以及数据访问解决方案，都可以无缝集成到 Web。ASP .NET 也是语言中立的，因此你可以选择你最熟悉的语言，或者通过几种语言来共同完成一个应用。而且，CLR 的互用性可以保证你升级到 ASP .NET 的时候，现存的基于 COM 的开发投资依然保留。
- 简单。ASP .NET 使执行常用的工作变得很容易，比如从简单的表单提交、客户端验证，到部署和站点配置。例如，ASP .NET 允许你建立用户接口，实现页面和逻辑代码的分离；同时，就像 Visual Basic 的表单执行模式那样来处理事件（也就是说，由页面驱动模式变成了事件驱动模式）。此外，CLR 简化了部署，用来管理代码服务，例如自动参照和垃圾回收。
- 易于管理。ASP .NET 使用一个基于文本的、分层次的配置系统，它简化了服务器端环境和 Web 应用程序的设置。由于配置信息用纯文本格式保存，新的设置不需要本地管理工具的支持。这种“零本地支持”的理念也同样应用到了部署 ASP .NET 应用程序。ASP .NET 应用程序部署到服务器，简化为复制必要的文件到服务器。在部署，甚至替换正在运行的变异代码的时候，也不需要重新启动服务器。
- 可伸缩性和有效利用性。ASP .NET 被设计成可伸缩的，能为集群和多处理器环境



设计提高性能。而且，ASP .NET 运行时刻密切监视和管理进程，这样，如果发生了错误的行为，例如漏洞和死锁，新的进程会在当前位置建立，来帮助你的应用程序继续处理事件请求。

- 可订制和扩展。ASP .NET 提供一种良好的扩充结构，允许开发者在适当的级别“插入”他们的代码。事实上，使用自己编写的组件可以扩展或者替换 ASP .NET 运行时刻的任何子组件。执行自定义验证或状态服务变得前所未有的容易。
- 安全性。内建的 Windows 验证和每一个应用程序的独立配置，你可以认为你的应用程序是安全的。
- ASP .NET 在改进应用程序功能方面创造出令人难以置信的奇迹：在现有基于 ASP 的应用程序性能基础上，性能优化了三倍之多，更为激动人心的是生产效率再度攀升。

5. 小结

因此，可以说 Visual Basic .NET 的开发着眼于世界各地 Visual Basic 开发者的要求。现在 Visual Basic 语言已经是真正的面向对象语言，并支持实现继承。窗体设计器支持可视化继承，并包含窗体自动调整大小、资源本地化和访问选项支持等新特性。现在的数据工具继续支持 XML 数据，并可以共同使用设计时数据绑定和断开数据。此外，Visual Basic .NET 是基于 .NET 框架直接创建的，因此可以使用所有的平台特性，并能够与其他 .NET 语言协同工作。

1.1.2 熟悉 Visual Basic .NET 集成开发环境

第一次启动 Visual Basic .NET 时，用户首先注意到的不会是 Visual Basic 优秀的面向对象支持，而应该是它的 IDE (Integrated Development Environment, 集成开发环境)。Visual Studio .NET 的 IDE 给人以非常熟悉的感觉，因为设计这个 IDE 的工作组以前就设计过 Visual Basic 的 IDE，Visual Studio .NET IDE 的改进建立在设计 Visual Basic IDE 的经验之上。让我们通过创建一个新的工程来熟悉一下 Visual Basic .NET 的开发环境。

创建新工程分为三个步骤：

(1) 打开 Visual Basic .NET。用户在【开始】|【程序】菜单中选择【Microsoft Visual Studio .NET】中的【Microsoft Visual Studio .NET】选项，打开 Visual Basic .NET，启动结果如图 1.1 所示。

(2) 新建工程。选择【File】菜单中的【New】选项，然后在【New】菜单下选择【New Project】，可以看到如图 1.2 所示的对话框。在对话框里输入工程名称，例如，可以把这个工程叫做 Test。在下面的对话框里输入建立项目的位置，也就是项目文件存放在硬盘的位置。Visual Basic .NET 的新特性之一就是在建立项目的同时就把项目存在硬盘里，这样就避免了以前因为在项目调试中突然死机，而导致整个项目数据丢失的情况。最后单击【OK】按钮。





建立项目。

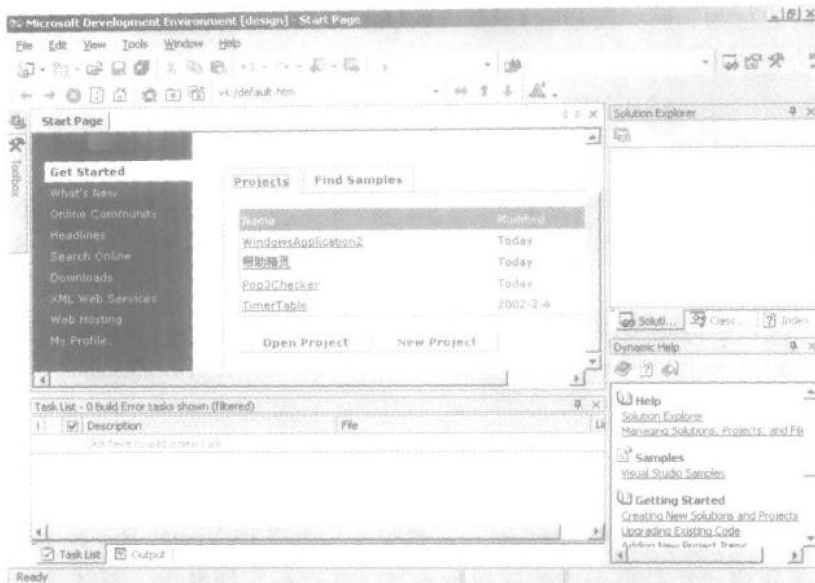


图 1.1 Visual Basic .NET 开发平台界面

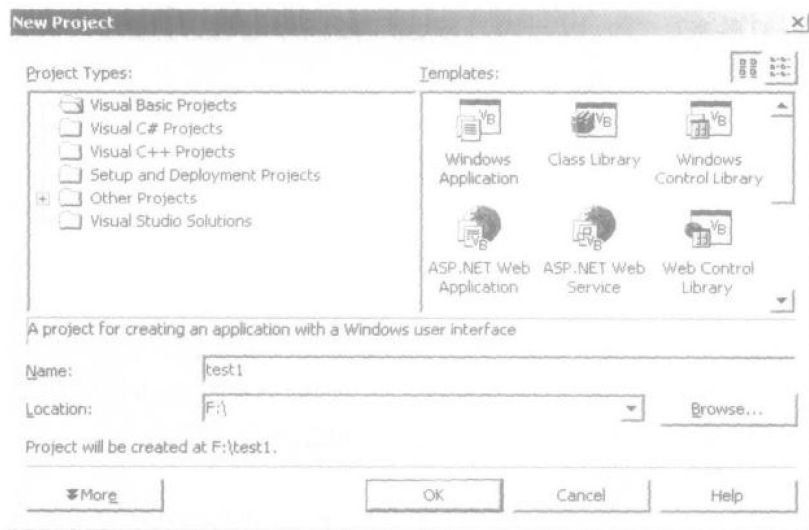


图 1.2 新建 Test 工程图

(3) 选择【View】菜单中的【Properties】选项，打开工程属性对话框，如图 1.3 所示。在该对话框中可以设置该工程的各种属性，如该工程的版本信息等。

Visual Basic 对于 IDE 的改进并不停留在表面上，IDE 内提供的新工具非常全面和强大。例如，所有的设计窗口都可以自动隐藏（就像 Windows 任务条一样），从而显著地减少了混乱的感觉。



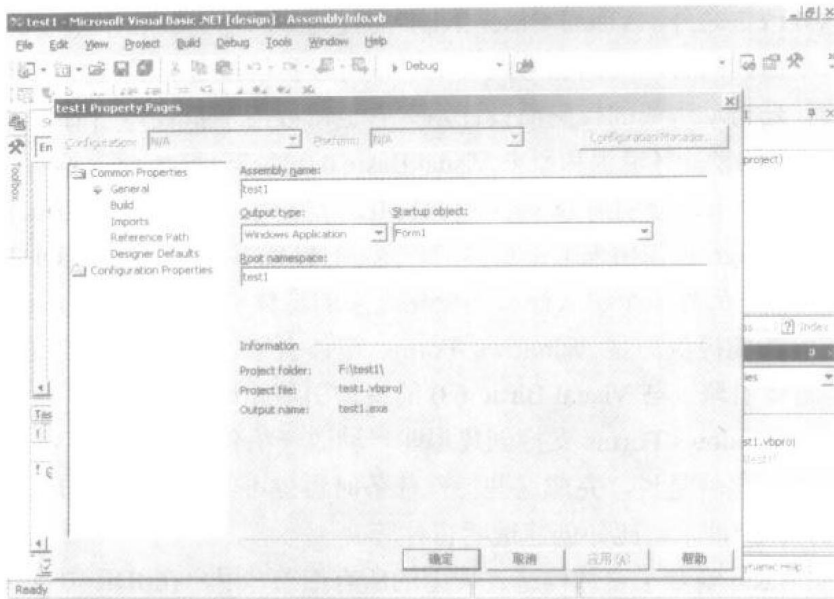


图 1.3 设置工程属性

- 隐藏工具框。默认情况下，工具框大多数时候都是隐藏的，单击它的标题可以把它“拉出来”。对于开发者来说，这里最大的变化在于工具框中的控件现在改为垂直排列，每个图标都加上了文字说明，而不像 Visual Basic 6.0 那样只有一个图标。
- 卡式子窗口（Tabbed Child Window）。Visual Studio .NET 利用位于屏幕上方的卡式子窗口布局取代了原来的 MDI 子窗口布局。卡包括：Home Page、窗体布局窗口、代码窗口和帮助屏幕。
- 任务列表（Task List）。Task List 是一个重要的改进，它帮助开发者跟踪所有待完成的任务。代码生成器、编译器会在 Task List 中自动加上一些内容。手工添加任务项目可以按照如下方式进行：找到代码中需要修改的地方，然后插入一行以“TODO:”开头的注释，此后该任务就会自动出现在任务列表中。单击 Task List 中的某个任务，代码编辑器中就会出现完成该任务的位置。这种快速定义任务、返回任务的机制将极大地提高开发效率。
- 方案资源管理器（Solution Explorer）。它替代了原来的工程资源管理器（Project Explorer），与 Visual InterDev 中的资源管理器非常相似。和 InterDev 中的资源管理器一样，Solution Explorer 显示了工程所用到的各种资源，包括用其他语言编写的代码模块、各种非代码的资源。其中 References 是 Solution Explorer 中一种全新的资源形式，References 管理着命名空间，类似于 Visual Basic 6.0 中的组件引用。
- 属性（Properties）框。Properties 框与 Visual Basic 6.0 中的属性框非常相似，但它现在支持更多的布局方式（显示方式）。
- 窗体布局（Form Layout）工具条。在 Visual Basic 6.0 中，在窗体上排列控件的任务通过一个名为“Form Editor”的工具条完成。改进后的新工具条提供更多的功能，而且在 WinForms 类型的工程中默认打开。



除了上面六处改进之外，Visual Basic .NET 的代码编辑器还有一个重要的新功能：在代码编辑窗口中，单击代码行左边的加号或者减号，可以扩展或者折叠代码块。这个功能原本为隐藏可视化设计器自动生成的代码而设计，但它也可以用于隐藏与当前任务无关的代码。

Web Forms 工程的设计环境也和原来 Visual Basic 6.0 的设计环境略有不同。在表单布局窗口的下面只显示了两个卡，分别为 Design 和 HTML。这两个卡的功能和 Visual InterDev 中同一位置的卡相似。Design 窗口用来编辑窗体的外观，HTML 窗口用于查看 DHTML 代码。

Visual Basic .NET 另外一个引人注目的变化是它的窗体引擎。Microsoft 放弃了原来的窗体引擎，现在它给我们提供的是 Windows Forms 窗体引擎。所有基于 CLR 的语言都使用 Windows Forms 窗体引擎，与 Visual Basic 6.0 的窗体引擎相比，Windows Forms 有着许多突出的优点。例如，Windows Forms 支持创建那些自动改变组件大小的窗体，允许把控件锚定到窗体的特定位置。也就是说，完成这些特殊任务时再也不需要第三方的工具了。Windows Forms 还有一些有趣的技术，比如创建透明窗体等。

以前，Visual Basic 隐藏了窗体构造过程中的所有细节。用户在 IDE 中设计窗体，然后把代码加入到 Initialize 事件，但对这两者之间的处理过程却无法进行任何控制。现在，窗体成了类，类里面包含了构造窗体的全部代码。大多数程序员都不会去看这种描述窗口构造细节的代码，如果说有某种操作一定会破坏应用，这个操作就是搞乱这里的代码。另一方面，由于这些代码允许用户深入控制 Visual Basic .NET 构造窗体的幕后细节，有经验的用户能够利用这些代码实现各种高级功能。如果你不想看到这些代码，可以不看，因为新的代码编辑器有扩展和折叠代码区域的功能，描述窗体构造过程细节的代码默认处于折叠状态。代码编辑器还有其他一些新的实用功能，比如自动缩进、内建的行号显示功能等。

下面简单介绍 Visual Basic .NET 开发环境中常用的工具栏和便捷窗口。

(1) **【菜单栏】**：包含 Visual Basic .NET 的所有控制功能，如图 1.4 所示。

File Edit View Project Build Debug Tools Window Help

图 1.4 菜单栏

(2) **【工程】窗口**（也称为方案资源管理器，Solution Explore）：显示目前项目所含的窗体和类，如图 1.5 所示。

(3) **【属性】窗口**（也称为属性框，Properties）：显示目前的选择对象，主要用在项目设计阶段，将所需的对象，安置在窗体上，如图 1.5 所示。

(4) **【类浏览】窗口**：显示在此工程中所建的所有类及其内部结构，如图 1.5 所示。

(5) **【工具栏】窗口**：提供常用功能按钮，如图 1.6 所示。如果设计员移动鼠标至某个按钮上，就有该按钮的功能名称出现。

(6) **【动态帮助】窗口**：动态地显示用户当前操作的帮助，如图 1.7 所示。

(7) **【工具箱】窗口**：存放着 Visual Basic 里面常用的控件，如图 1.7 所示。用户也可以根据自己的需要添加和删除一些按钮。



(8)【查找帮助】窗口：用户可以输入关键字，对需要了解的知识点进行帮助查询，IDE将调用 MSDN 进行查询，如图 1.7 所示。

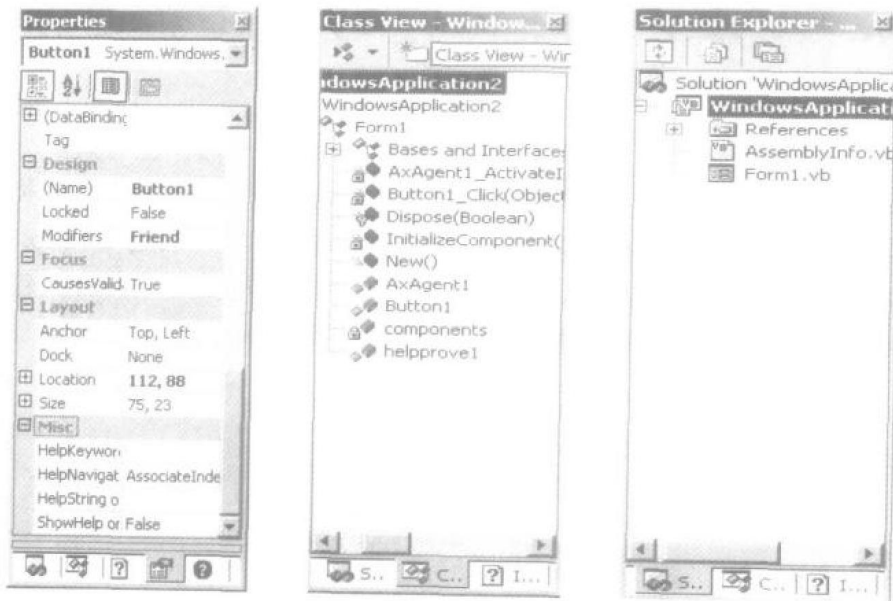


图 1.5 属性窗口、类浏览窗口、工程管理窗口



图 1.6 工具栏

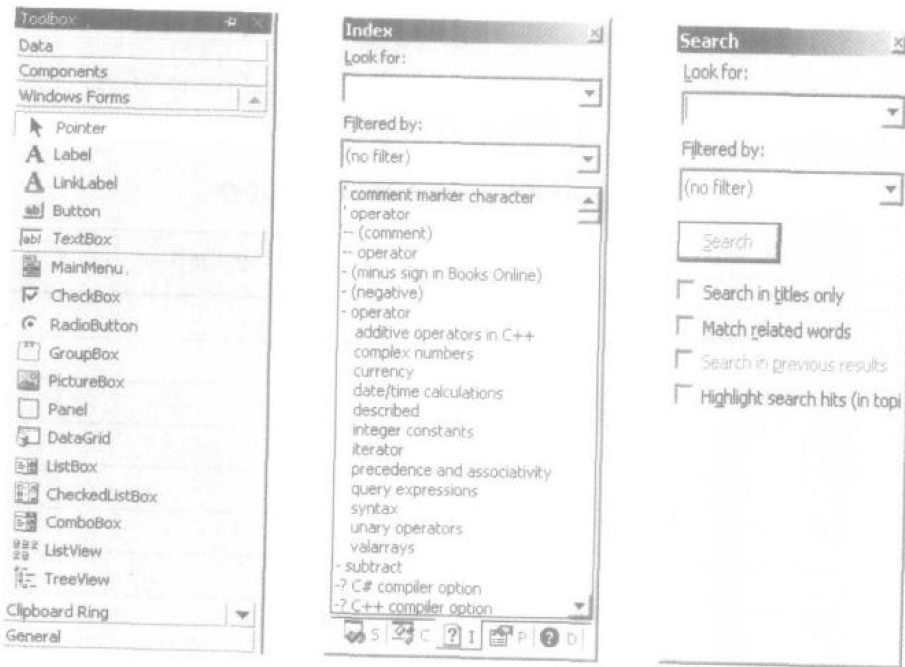


图 1.7 工具栏、动态帮助和查找帮助窗口





1.2 .NET 控件编程的变化和区别

从内部构造来看, Visual Basic .NET 控件与传统的 ActiveX 控件存在很大差别, 因为 .NET 控件是从其他类的层次结构创建的, 如图 1.8 所示。这些 .NET 控件所继承的类有若干个, 每一个类都经过优化并添加了更多的功能。控件的类层次结构的顶层是 Control 类。同时, 用户自定义控件则很可能是从 UserControl 类继承而来的, 因为 UserControl 类包含控件定制过程常用的属性和事件, 而且还能为用户绘制所有的构成控件。

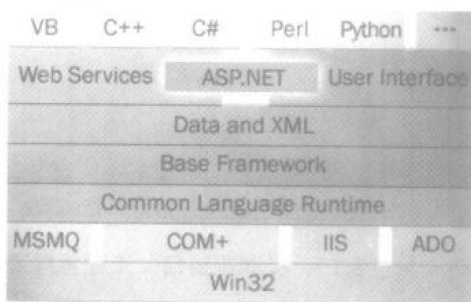


图 1.8 Visual Basic .NET 内部构造

1.2.1 被重命名的控件

使用 .NET 创建用户界面时, 用户会发现很多熟悉的 Visual Basic 6.0 控件已经不见了或者已经换了名称。如表 1.1 所示。

表 1.1 在 .NET 中被重命名的 ActiveX 控件

Visual Basic 6.0 标准控件	等效的 .NET 控件
Picture	PictureBox
Label	Label
TextBox	TextBox
Frame	GroupBox
CommandButton	Button
CheckBox	CheckBox
OptionButton	RadioButton
ListBox	ListBox
ComboBox	ComboBox
HScrollBar	HScrollBar