



高职高专“工学结合”特色教材

主 编 于瑞琴

VB.NET 项目开发教程



江苏大学出版社
JIANGSU UNIVERSITY PRESS



高职高专“工学结合”特色教材

VB.NET

项目开发教程

主 编 于瑞琴

副主编 朱春晖 沈润泉

 江苏大学出版社
JIANGSU UNIVERSITY PRESS
镇 江

图书在版编目(CIP)数据

VB.NET 项目开发教程 / 于瑞琴主编. —镇江: 江苏大学出版社, 2014. 12
ISBN 978-7-81130-901-0

I. ①V… II. ①于… III. ①BASIC 语言—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 310669 号

VB.NET 项目开发教程
VB.NET XIANGMU KAIFA JIAOCHENG

主 编/于瑞琴
责任编辑/徐 婷
出版发行/江苏大学出版社
地 址/江苏省镇江市梦溪园巷 30 号(邮编: 212003)
电 话/0511-84446464(传真)
网 址/http://press. ujs. edu. cn
排 版/镇江文苑制版印刷有限责任公司
印 刷/虎彩印艺股份有限公司
经 销/江苏省新华书店
开 本/787 mm×1 092 mm 1/16
印 张/21. 25
字 数/508 千字
版 次/2014 年 12 月第 1 版 2014 年 12 月第 1 次印刷
书 号/ISBN 978-7-81130-901-0
定 价/42. 00 元

如有印装质量问题请与本社营销部联系(电话:0511-84440882)

前言

Preface

VB.NET 属于计算机软件领域,是计算机中实现网络功能的编程语言,是微软开发的基于.NET 平台的编程语言,其在调试时是以解释型语言方式运作,而输出的 EXE 程序是以编译型语言方式运作。VB.NET 是在 VB 基础上改进的语言,但却与 VB 有着天壤之别,从基础的数据类型到事件和方法的引用都有了很大的改进,在学习时应作为一种新的语言来看待,而不是 VB 的增强版。VB.NET 是一种完全面向对象的语言,它不但继承了 VB 的语言基础,而且还增加了.NET 框架的特性,能够运用.NET 特性实现程序开发。VB.NET 的应用范围包括 Windows 桌面、Web 以及移动平台 Windows Phone。

由于高职教育有别于本科教育,原来基于学科知识的教材并不适合高职教育的需要,而目前高职院校的教学中合适的 VB.NET 教材并不多,大部分教材在内容选取上还是沿用了本科教材编写的模式,即学科性强,但应用性不够,这不符合高职教育的培养目标。为此,编者根据高职教育的特点编写了本书,以培养学生职业岗位技能为目标,侧重技能应用,内容的选取本着“实用、适用、够用”的原则,通过一个完整的应用项目,系统地介绍了 VB.NET 程序开发技术。

本教材按照“工学结合”课程思路和创新模式来编写,针对学生,合理设置课程项目模块,依据“项目——模块——任务”的形式,使课程的知识结构模块化,采用“任务驱动”,在确定“模块目标”的前提下,带着“工作任务”进行“知识准备”,按照“任务实施”步骤进行讲解,让学生掌握 VB.NET 的基本原理与技术,培养学生的相应技能,具有可操作性强的特点。

本教材的教学项目选用协同报价管理系统,包括登录模块、系统主模块、供应商管理模块、物料管理模块、报价管理模块、报表以及协同报价管理模块等,课内实训项目选用书店图书管理系统。

在体系架构方面,本教材每章开头均介绍了该章学习要点、技能目标,章后设置了课内实训题、工作实践常见问题解析与练习题,用于加强对所学知识的理解和综合应用。

本教材由于瑞琴老师担任主编,朱春晖工程师和沈润泉老师担任副主编,翟云峰和江卫星老师参编,在此向对本书出版给予支持帮助的单位和个人表示诚挚的感谢!

随着 VB.NET 的发展,书中难免存在不妥之处,真诚希望得到广大专家、读者的批评和指正。

编 者

2014 年 11 月

目 录

Contents



第1章 .NET 概述

- 1.1 VB.NET 概述 / 1
 - 1.1.1 VB.NET 的特点 / 2
 - 1.1.2 VB.NET 版本的演变 / 2
 - 1.1.3 VB.NET 和 VB 的关系 / 3
- 1.2 .NET 体系结构 / 4
 - 1.2.1 .NET Framework 版本发展历程 / 4
 - 1.2.2 .NET Framework 各版本之间的关系 / 8
 - 1.2.3 .NET 框架组件 / 8
- 1.3 .NET 应用程序的编译和执行 / 9
- 1.4 VB.NET 开发环境的配置及安装 / 10
- 1.5 工作实训营 / 14
- 习 题 / 16



第2章 系统分析与说明

- 2.1 系统说明与开发背景 / 17
- 2.2 系统概要说明书 / 18
- 2.3 系统需求分析 / 21
- 2.4 系统可行性分析 / 23
 - 2.4.1 技术可行性分析的前提 / 23
 - 2.4.2 对现有系统的分析 / 25
 - 2.4.3 可选择的系统方案 / 25
- 2.5 系统详细说明书 / 25
 - 2.5.1 界面设计要求 / 25
 - 2.5.2 系统模块设计要求 / 25
- 2.6 工作实训营 / 31
- 习 题 / 31



第3章 系统设计

- 3.1 系统架构设计目标与准则 / 32
- 3.2 程序运行环境及编码规范 / 33
 - 3.2.1 文档说明 / 33
 - 3.2.2 .NET 命名与编码标准 / 33
 - 3.2.3 数据库命名标准 / 37
 - 3.2.4 界面设计标准 / 37
 - 3.2.5 程序开发及运行环境 / 40
 - 3.2.6 编码设计 / 40
- 3.3 项目文档架构 / 41
 - 3.3.1 项目文档结构说明 / 41
 - 3.3.2 程序文件架构 / 43
- 3.4 系统业务功能结构设计 / 44
- 3.5 系统业务流程图设计 / 45
- 3.6 数据库设计 / 45
- 3.7 工作实训营 / 54
- 习 题 / 55



第4章 公共模块设计

- 4.1 ADO.NET 数据库访问技术 / 57
 - 4.1.1 ADO.NET 概述 / 57
 - 4.1.2 ADO.NET 结构 / 58
- 4.2 公用数据库连接 / 64
- 4.3 存储过程 / 65
- 4.4 工作实训营 / 69
- 习 题 / 71



第5章 登录模块设计

- 5.1 工作场景导入 / 72
- 5.2 创建项目 / 73
- 5.3 Windows 窗体以及常用控件 / 74
 - 5.3.1 Windows 窗体 / 74
 - 5.3.2 VB.NET 的常用控件 / 76
- 5.4 创建一个简单的应用程序 / 80
- 5.5 VB.NET 的语言基础 / 82
 - 5.5.1 代码书写规则 / 82

- 5.5.2 基本数据类型 / 82
- 5.5.3 变量 / 83
- 5.5.4 选择结构 / 84
- 5.5.5 异常处理 / 87
- 5.6 回到工作场景 / 87
- 5.7 工作实训营 / 92
- 习 题 / 94

➤ 第6章 主窗体模块设计

- 6.1 工作场景导入 / 96
- 6.2 菜单栏的设计 / 97
- 6.3 工具栏的设计 / 104
- 6.4 状态栏的设计 / 108
- 6.5 多文档窗体的设计 / 111
 - 6.5.1 应用程序界面 / 111
 - 6.5.2 设计 MDI 应用程序 / 113
- 6.6 回到工作场景 / 116
- 6.7 工作实训营 / 120
- 习 题 / 122

➤ 第7章 供应商管理模块设计

- 7.1 面向对象程序设计 / 123
 - 7.1.1 面向对象程序设计的基本特性 / 123
 - 7.1.2 类和对象的定义 / 124
 - 7.1.3 类的继承与派生 / 134
 - 7.1.4 类的多态性 / 137
 - 7.1.5 供应商类的定义 / 138
- 7.2 供应商信息管理模块设计 / 141
- 7.3 供应商查询模块设计 / 154
- 7.4 供应商价格管理模块设计 / 159
- 7.5 供应商价格增改模块设计 / 167
- 7.6 工作实训营 / 172
- 习 题 / 173

➤ 第8章 物料管理模块设计

- 8.1 物料信息管理模块设计 / 175
- 8.2 物料查询模块设计 / 189

8.3 物料清单 BOM 管理模块设计 / 193

8.4 物料清单 BOM 增改模块设计 / 205

8.5 工作实训营 / 216

习 题 / 218

➔ 第 9 章 报价管理模块设计

9.1 报价信息管理模块设计 / 219

9.2 报价查询模块设计 / 229

9.3 报价新增管理模块设计 / 236

9.4 工作实训营 / 252

习 题 / 254

➔ 第 10 章 报 表

10.1 工作场景导入 / 255

10.2 创建图书分类报表 / 256

10.3 依据现有的报表创建新报表 / 269

10.4 回到工作场景 / 271

10.5 工作实训营 / 284

习 题 / 285

➔ 第 11 章 系统管理模块设计

11.1 协同报价管理模块设计 / 286

11.2 用户管理模块设计 / 294

11.3 用户权限管理模块设计 / 304

11.4 工作实训营 / 313

习 题 / 314

➔ 第 12 章 创建并使用 Web 服务

12.1 了解 Web 服务 / 315

12.2 创建 Web 服务 / 316

12.3 使用 Web 服务 / 321

12.4 工作实训营 / 326

习 题 / 330

参考文献 / 331

第1章.NET 概述本章要点

1. VB.NET 概述;
2. .NET 体系结构;
3. .NET 编译和执行;
4. VB.NET 开发环境的配置和安装。

技能目标

1. 掌握.NET 和 VB.NET 的特点;
2. 了解.NET 版本的发展历程;
3. 掌握公共语言运行时 CLR 和 .NET 框架类库;
4. 了解.NET 的编译和执行;
5. 掌握 VB.NET 开发环境的配置和安装。

1.1 VB.NET 概述

VB.NET 属于计算机软件领域,是计算机中实现网络功能的编程语言,是微软最新平台技术,是 .NET Framework SDK 的一种语言。VB.NET 和 VC#.NET 在功能上没有区别。编译以后生成的可执行文件被称为 Assembly,即程序集。

什么是 VB.NET? VB.NET 是新一代的 Visual Basic,但并不是简单的在 Visual Basic 6.0 上再添加一些新特性而已,微软重新设计了产品以便使开发者能够更加容易地开发分布式应用,例如基于 Web 的程序以及多层系统。

VB.NET 中有两种窗体包: Windows 窗体和 Web 窗体。一个新版的 ADO 用于接受离线数据源;新的语言移走了原来的关键字,提高了 Type 数据的安全性,提供了低级别结构,以满足高级别开发者的需求。

这些新的特性为 VB.NET 开发者提供了新的开发窗口:通过 Web 窗体以及 ADO.NET,可以快速开发可扩展的 Web 站点;通过继承,该语言实现了真正的支持面向对象编程;Windows 窗体本身支持可视继承;现在展开程序就像拷贝可执行文件和控件到另一个目录那样简单。

VB.NET 现在完全集成在其他的 Microsoft Visual Studio .NET 语言中,不仅可以使使用不同的语言开发组件,而且通过交叉语言继承,可以从一种语言编写的类中派生另一种语言编写的类。

1.1.1 VB.NET 的特点

(1) .NET 的特点

- ① 真正成为面向对象以及支持继承性的语言。
- ② 窗体设计器支持可视化继承,并且包含了许多新的特性,比如自动改变窗体大小、资源本地化支持、数据类工具内在支持 XML 数据。
- ③ 直接建立在.NET 的框架结构上,因此开发人员可以充分利用所有.NET 平台特性,也可以与其他的.NET 语言交互。

④ 为 Windows 应用程序提供了 xcopy 部署,开发者不再需要为 DLL 的版本问题担忧。

(2) VB.NET 的特点

① 代码托管。被托管的代码享受.NET Framework 提供的安全保障和垃圾回收机制,但这也同时表明程序被框在 Framework 里面了,API 变得不太方便。

② 强大的面向对象特性。VB.NET 已经是一个完全的面向对象程序。举个例子:

VB 6:

```
Form2.Show()
```

VB.NET:

```
Dim frm As New Form2  
frm.Show()
```

现在 VB 7 已经支持类的各种特性,如继承、函数的覆盖、重载、虚拟、隐藏等。

- ③ 功能强大,程序界面更标准。
- ④ 程序代码结构化更强,开发环境舒适。

1.1.2 VB.NET 版本的演变

(1) VB.NET 2002

2002 年 VB.NET 2002 问世,此后 Visual Basic 包含在 Visual Studio 套装中。该版本又被称为 VB 7.0,是随 VC#.NET 和 ASP.NET 一起在 2002 年发布的最初始的 VB.NET 版本。VC#.NET 被大力宣传为微软对付 Java 的王牌,受到媒体极力关注,然而 VB.NET(有时被称为 VB 7)影响面不广,这导致 VB 社区之外很少人关注它。那些试用过此版 VB.NET 的人在层层幕布下发现一个强大却艰深的语言,而且还有不少弱点,比如它与 VB 6 相比有一个大了十倍的运行时(RunTime),内存 footprint 也大大增加。

(2) VB.NET 2003

该版本又被称为 VB 7.1,是和.NET Framework 1.1 一起发布的。新功能包括对.NET Compact Framework 的支持和更好的 VB 升迁向导,并改进了运行状况、IDE 稳定性(尤其是后台编译器)以及运行时(RunTime)稳定性。另外,VB.NET 2003 还有一个学院版

Visual Studio.NET 2003 Academic Edition (VS03AE),它是分发给各国某些学者免费使用的版本。

(3) Visual Basic 2005

该版本又被称为 VB 8.0,是 VB.NET 的重大转变,微软决意在其软件名称中去掉 .NET部分,但这个版本的 Visual Studio 仍然是面向.NET 框架的(版本 2.0)。它同时也能开发跨平台的应用程序,如开发使用微软操作系统的手机的程序等。总体来说,它是一个非常庞大的软件,甚至包含代码测试功能。

(4) Visual Basic 2008

Visual Basic 2008 即 VB 9.0,于 2008 年发布。通过几年的发展,它已成为一种专业化的开发语言和环境。用户可用 Visual Basic 快速创建 Windows 程序,现在还可以编写企业水平的客户/服务器程序及强大的数据库应用程序。

(5) Visual Studio 2010

Visual Studio 2010 于 2010 年上市,其中包含 Visual Basic 版本 10(有时称为 VB 2010 或 VB 10)。Visual Studio 是微软公司推出的开发环境,是目前最流行的 Windows 平台应用程序开发环境。Visual Studio 2010 集成开发环境(IDE)的界面被重新设计和组织,变得更加简单明了。Visual Studio 2010 同时带来了 .NET Framework 4.0 和 Microsoft Visual Studio 2010 CTP(Community Technology Preview),并且支持开发面向 Windows 7 的应用程序。除了 Microsoft SQL Server 外,它还支持 IBM DB2 和 Oracle 数据库。

1.1.3 VB.NET 和 VB 的关系

VB.NET 是否该被看作 VB 的新版本或者是另一种完全不同的语言尚在争议中。虽然并非显而易见,但只要某个方法被弃用,它就可能会被自动转换,这说明该语言的句法并没有出现很多突破性的变化,只是增加了对新功能的支持,比如结构化意外处理机制和短循环表达式。有一种变化会让旧用户迷惑:Integer 和 Long 数据类型在长度上分别倍增。16 位的 Integer 在 VB.NET 中被认为是 Short,而 Integer 和 Long 分别是 32 位和 64 位。同样,Windows Forms GUI 编辑器和 VB Form 编辑器在外观和功能上非常相似。

出现巨大变化的是语意,从运行于基于 COM 的 deterministic 和 reference-counted 的引擎之上的编程环境,转变为全面面向对象的基于 .NET Framework 的语言。.NET Framework 包括了 CLR(使用通用垃圾收集机制和 JIT 编译引擎的虚拟机)和大量的类库。VB 开发者进入 VB.NET 要面对新增大量类库的问题。

上述变化表明了许多在运行及维护相关优化事宜下面的假定,一些 VB 6 函数和库不再存在而其他的仍然可用,但是不如 .NET 语言中自带的可替换的函数和库来得高效。即使它们可被编译,但大多从 VB 6 转换而来的应用将需要一定程度上的重构(refactoring),以全面利用新语言的优点。在句法、应用调试、部署和术语等方面的变化,已有大量的文档可以参考。

1.2 .NET 体系结构

.NET 体系结构包括了有助于创建和运行具有强大生命力、高扩展性和分布式应用程序的所有技术,其体系结构的核心是.NET Framework,它是服务和类的集合,在.NET 应用程序和操作系统之间以层的方式存在。

.NET 框架(.NET Framework)是一个多语言组件开发和执行环境,它提供了一个跨语言的统一编程环境。.NET 框架的目的是便于开发人员更容易地建立 Web 应用程序和 Web 服务,使得 Internet 上的各应用程序之间可以使用 Web 服务进行沟通。从层次结构来看,.NET 框架又包括三个主要组成部分:公共语言运行时(Common Language Runtime, CLR)、服务框架(Services Framework)和上层的两类应用模板——传统的 Windows 应用程序模板(WinForms)和基于 ASP.NET 的面向 Web 的网络应用程序模板(WebForms 和 Web-Services)。

CLR 是一种运行时环境,管理代码的执行使开发过程变得更加简单。同时,CLR 也是一种受控的执行环境,其功能通过编译器与其他工具共同展现。

在 CLR 之上的是服务框架,它提供了一套开发人员希望在标准语言库中存在的基类库,包括集合、输入/输出、字符串及数据类。

那么,在 Windows DNA(分布式集成网络应用体系结构)之后,微软提出新的.NET 框架(新托管代码编程模型)的主要原因是什么?

其一,将采用多种技术开发的一个应用程序的若干子系统进行整合。例如,一个制造企业有不同的系统,如库存管理系统、物料清单系统、财务总账系统,使用可用于应用程序开发的各种技术实现。这些系统需要集成在一起,从而形成一个更高级别的企业信息系统的组织。要做到这一点,应用程序开发人员必须使用如微软的分布式组件对象模型(DCOM)、通用对象请求代理体系结构(CORBA)、Java 远程方法调用(RMI)等技术,通过这些技术已开发的应用程序编程语言非常紧密地耦合在一起。

其二,具有跨语言的互操作性。在早期的技术中,真正的代码的可重用性是受限的,因此开发人员不得不学习被用于应用程序的开发组织用到的所有语言。例如,如果在 Visual C++ 类已经被创建,那么不可能在 Visual Basic 中开发新的类并将其扩展到 Visual C++,因此开发者将不得不用每一种项目中用到的语言重新编写同样逻辑的类。.NET 框架具有跨语言的互操作性,使代码复用得到了支持。

其三,.NET 组件和 COM 组件之间的互操作性。COM 组件注册后才可以在目标机器上使用,应用程序必须查找 Windows 注册表并加载 COM 组件。.NET 框架中,既可以在 .NET 组件中调用 COM 组件,也可以在 COM 组件中调用 .NET 组件。

1.2.1 .NET Framework 版本发展历程

.NET Framework 版本发展历程如表 1.1 所示。

表 1.1 .NET Framework 版本发展历程

版本	完整版本号	发行日期	Visual Studio	Windows 默认安装
1.0	1.0.3705.0	2002-02-13	Visual Studio.NET 2002	Windows XP Media Center Edition Windows XP Tablet PC Edition
1.1	1.1.4322.573	2003-04-24	Visual Studio.NET 2003	Windows Server 2003
2.0	2.0.50727.42	2005-11-07	Visual Studio 2005	
3.0	3.0.4506.30	2006-11-06		Windows Vista Windows Server 2008
3.5	3.5.21022.8	2007-11-19	Visual Studio 2008	Windows 7 Windows Server 2008 R2
4.0	4.0.30319.1	2010-04-12	Visual Studio 2010	
4.5	4.5.40805	2012-02-20	Visual Studio 2012 RC	Windows 8 RP Windows Server 8 RC

(1) .NET 1.0

这是最初的.NET 框架,发行于 2002 年。它可以以一个独立的可重新分发的包的形式或在一个软件发展工具包集中被获得。它是第一个微软 Visual Studio.NET 发行版的一部分。

(2) .NET 1.1

这是首个主要的.NET 框架升级版本,发行于 2003 年。它可以以一个独立的可重新分发的包的形式或在一个软件发展工具包集中被获得。它是第二个微软 Visual Studio .NET 版本的一部分(也被称作 Visual Studio.NET 2003),也是首个被 Windows 操作系统——Windows Server 2003 所内置的.NET 框架版本。

自 1.0 版本以来的改进:

① 内置了对 Mobil ASP.NET 控件的支持。这在 1.0 版本是以附加功能方式实现的,现在已经集成到框架的内部。

② 安全方面的变更使得 Windows 窗体代码以可靠的行为运行,从而可以在 Internet 环境内安全运行,并且加入了 ASP.NET 应用程序的代码安全访问功能。

③ 内置了对 ODBC 和 Oracle 数据库的支持。这在 1.0 版本是以附加功能方式实现的,现在已经集成到框架的内部。

④ .NET Compact Framework 是一个用于智能设备的.NET 框架的子集。

⑤ 对 IPv6 的支持。

⑥ 大量的 API 变更。

(3) .NET 2.0

.NET 框架 2.0 的组件都包含在 Visual Studio 2005 和 SQL Server 2005 中。通过 MSDN Universe 版可以免费下载 RTM 版本。

自 1.1 版本以来的改进:

① 大量的 API 变更。

② 新的 API 使得需要管理.NET 运行库实例的非.NET 的应用程序可以做到这点。这个新的 API 对.NET 运行库的各种功能(如多线程、存储器分配和代码加载等)提供了很好

的控制。它最初是为 Microsoft SQL Server 能够有效率地使用 .NET 运行库而设计的,因为 Microsoft SQL Server 拥有它自己的日程管理器和存储器管理器。

(4) .NET 2.0 SP1

随 .NET 框架 3.5 同时发行。

(5) .NET 2.0 SP2

随 .NET 框架 3.5 SP1 同时发行。

(6) .NET 3.0

曾用名“WinFX”。依然使用 .NET 框架 2.0 版本的公共语言运行库 (CLR), 并加入了下列适应未来软件发展方向的 4 个框架。

① Windows Presentation Foundation (WPF): 提供更佳的用户体验, 用来开发 Windows Forms 程序以及浏览器应用程序 (参见 WPF/E)。

② Windows Communication Foundation (WCF): 提供面向服务的软件构架 (SOA) 支持的安全的网络服务 (Web Service) 框架。

③ Windows Workflow Foundation (WF): 提供一个设计与发展工作流程导向应用程序基础支持的应用程序接口。

④ Windows CardSpace: 提供一个单点登录 (SSO) 的解决方案, 每个用户都有各自的 CardSpace。

(7) .NET 3.0 SP1

随 .NET 框架 3.5 同时发行, Microsoft .NET Framework 3.0 Service Pack 1 针对 Microsoft .NET Framework 3.0 发布后客户报告的问题提供了累积的汇总更新。此外, 该更新还可帮助改进安全性, 且提供了用于 Microsoft .NET Framework 3.5 的系统必备功能支持。

(8) .NET 3.0 SP2

随 .NET 框架 3.5 SP1 同时发行。

(9) .NET 3.5

这个版本将包含一个支持 C# 和 VB.NET 中心的语言特性的编译器, 以及对语言集成查询 (Language-Integrated Query, LINQ) 的支持。该版本随 Visual Studio 2008 一起发布。

同时, .NET 框架 3.5 自动包含 .NET 框架 2.0 SP1 以及 .NET 框架 3.0 SP1, 用于为这两个版本提供安全性修复和少量新增的类库。此版本提供的新功能有以下几种:

① 扩展方法 (Extension Method) 属性 (Attribute), 用于为扩展方法提供支持。

② LINQ 支持, 包括 LINQ to Object, LINQ to ADO.NET, LINQ to XML。

③ 表达式目录树 (Expression Tree), 用于为 Lambda 表达式提供支持。

④ 与语言集成查询 (LINQ) 和数据感知紧密集成。借助这个新功能, 可以使用相同的语法, 在任何支持 LINQ 的语言中编写相关代码, 以筛选和枚举多种类型的 SQL 数据、集合、XML 和数据集, 以及创建它们的投影。

⑤ 利用 ASP.NET AJAX 可以创建更有效、更具交互性、高度个性化的 Web 体验, 这些体验在所有最流行的浏览器上都能实现。

⑥ 用于生成 WCF 服务的全新 Web 协议支持, 包括 AJAX、JSON、REST、POX、RSS、ATOM 和若干新的 WS-* 标准。

⑦ Visual Studio 2008 中面向 WF, WCF, WPF 的完整工具支持, 其中包括支持工作流的服务这一新技术。

⑧ .NET 框架 3.5 基类库 (BCL) 中的新类可满足许多常见的客户请求。

(10) .NET 3.5 SP1

.NET 框架 3.5 SP1 自动包含 .NET 框架 2.0 SP2 以及 .NET 框架 3.0 SP2。该版本随 Visual Studio 2008 SP1 发布, 此版本提供了下列新功能:

① 新增的 ASP.NET 功能, 包括 ASP.NET 动态数据和 ASP.NET AJAX 附加功能, 前者提供了无须编写代码就可实现数据驱动的快速开发的丰富支架框架, 后者为管理浏览器历史记录提供了支持(后退按钮支持)。

② ADO.NET Entity Framework。

③ 对 SQL Server 2008 的数据提供程序支持。

④ .NET 框架客户端配置文档是完整版 .NET 框架的子集, 面向客户端应用程序。这改善了尚未安装 .NET 框架的计算机上的安装体验。

⑤ 改进 Windows Presentation Foundation 的性能, 其中包括启动速度的位图效果性能的提高。为 Windows Presentation Foundation 增加的功能包括对业务线应用程序的更好支持、本机闪屏支持、DirectX 像素着色器支持以及新的 WebBrowser 控件。

⑥ ClickOnce 应用程序发布者可以根据具体情况决定是否取消签名和散列, 开发人员可以以编程方式安装显示自定义署名的 ClickOnce 应用程序, ClickOnce 错误对话框支持指向网络上特定于应用程序的支持站点的链接。

⑦ 用于 SQL Server 的 .NET 框架数据提供程序 (System. Data. SqlClient) 完全支持 SQL Server 2008 数据库引擎的所有新功能。有关 .NET 框架对 SQL Server 2008 支持的更多信息, 请参见 SQL Server 中的新功能 (ADO.NET)。ADO.NET 数据平台是一种多版本策略, 它使开发人员能够针对概念性实体数据模型进行编程, 从而减轻他们的编码和维护工作。此平台提供了 ADO.NET Entity Framework、实体数据模型 (EDM)、对象服务、LINQ to Entities、Entity SQL、EntityClient、ADO.NET 数据服务及实体数据模型工具。

⑧ Windows Communication Foundation 现在提供了改进的互操作性支持, 增强了部分信任方案中的调试体验, 并扩展了集成协议支持以更广泛地应用于 Web 2.0 应用程序, 从而使 DataContract 串行化程序更易于使用。

⑨ Microsoft Visual Basic PowerPacks 命名空间引入了新的 DataRepeater 控件, 该控件以可自定义的列表格式显示数据。此命名空间还包含新的矢量形状。

(11) .NET 4.0

.NET 框架 4.0 主要增加了并行支持, 英文版于 2010 年 4 月 12 日推出。企业基础 .NET 提供开发软件的独立平台, 内置高度安全的网络系统, 相当倚重软件组件以及组件导向程序。在这方面它完全取代了 COM。

(12) .NET 4.5

.NET 框架 4.5 修复了工作可靠性、兼容性、稳定性和性能方面的问题, 同时修复了公共语言运行时的 Windows Presentation Foundation、Windows 窗体、XML、网络类库、ASP、实体框架、Windows 工作流基金会的 Windows Communication Foundation。

1.2.2 .NET Framework 各版本之间的关系

.NET Framework 各版本之间的关系如图 1.1 所示。

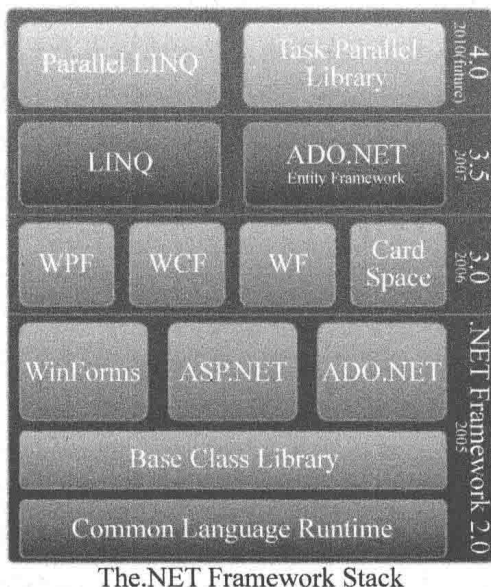


图 1.1 .NET Framework 各版本之间的关系

1.2.3 .NET 框架组件

(1) 公共语言运行时(CLR)

公共语言运行时(CLR)是所有.NET 应用程序运行时环境,是所有.NET 应用程序都使用的编程基础,它如同一个支持.NET Framework,运行一个.NET 程序也必须安装。CLR 也可以看作一个在执行时管理代码的代理,管理代码是 CLR 的基本原则,能够被管理的代码成为托管代码,反之称为非托管代码。CLR 包含两个组成部分:CLS(公共语言规范)和 CTS(通用类型系统)。可以通过理解.NET 的编程技术具体了解这两个组件的功能。

① CLS

编程语言的区别不仅仅在于类型,语法或者说语言规范也都有很大的区别。因此,.NET通过定义公共语言规范(Common Language Specification,CLS),限制了由这些不同点引发的互操作性问题。CLS 是一种最低的语言标准,制定了一种以.NET 平台为目标的语言所必须支持的最小特征,以及该语言与其他.NET 语言之间实现互操作性所需要的完备特征。凡是遵守这个标准的语言在.NET 框架下都可以实现互相调用。例如,在 C#中命名是区分大小写的,而 VB.NET 不区分大小写,这样 CLS 就规定,编译后的中间代码必须除了大小写之外有其他的不同之处。

② CTS

C#和 VB.NET 都是公共语言运行时的托管代码,它们的语法和数据类型各不相同。CLR 如何对这两种不同的语言进行托管呢? 通用类型系统(Common Type System,CTS)用于解决不同语言的数据类型不同的问题,如 C#中的整型是 Int,而 VB.NET 中是 Integer,通

过 CTS 可把它们两个编译成通用的类型 Int32。所有的 .NET 语言共享这一类型系统,在它们之间实现无缝互操作。

③ .NET 编译技术

为了实现跨语言开发和跨平台的战略目标,.NET 所有编写的应用都不是编译为本地代码,而是编译成微软中间代码(Microsoft Intermediate Language, MSIL)。它将由实时(Just-In-Time, JIT)编译器转换成机器代码。C#和 VB.NET 代码通过它们各自的编译器编译成 MSIL, MSIL 遵守通用的语法,CPU 不需要了解它,再通过 JIT 编译器编译成相应的平台专用代码,这里所说的平台是指操作系统。这种编译方式实现了代码托管,还能够提高程序的运行效率。

(2) .NET Framework 类库

.NET Framework 包括可加快和优化开发过程并提供对系统功能的访问的类、接口和值类型。为了便于语言之间进行交互操作,大多数 .NET Framework 类型都符合 CLS,因而在编译器符合 CLS 的任何编程语言中使用。

.NET Framework 类型是生成 .NET 应用程序、组件和控件的基础。.NET Framework 包括的类型可执行下列功能:① 表示基础数据类型和异常。② 封装数据结构。③ 执行 I/O。④ 访问关于加载类型的信息。⑤ 调用 .NET Framework 安全检查。⑥ 提供数据访问、多客户端 GUI 和服务端控制的客户端 GUI。⑦ 提供了一组丰富的接口以及抽象类和具体(非抽象)类。开发者可以按原样使用这些具体的类,或者在多数情况下从这些类派生用户自己的类。若要使用接口的功能,既可以创建实现接口的类,也可以从某个实现接口的 .NET Framework 类中派生类。

命名约定:.NET Framework 类型使用点语法命名方案,该方案隐含了层次结构的意思。此技术将相关类型分为不同的命名空间组,以便更容易地搜索和引用它们。全名的第一部分(最右边的点之前的内容)是命名空间名,全名的最后一部分是类型名。例如, System. Collections. ArrayList 表示 ArrayList 类型,该类型属于 System. Collections 命名空间。System. Collections 中的类型可用于操作对象集合。

此命名方案使扩展 .NET Framework 的库开发人员可以轻松创建分层类型组,并用一致的、带有提示性的方式对其进行命名。它还允许用全名(即命名空间和类型名称)明确地标识类型,这样可以防止类型名称发生冲突。库开发人员在创建命名空间的名称时应使用以下原则:公司名称. 技术名称。例如,用 Microsoft. Word 命名空间就符合此原则。

利用命名模式将相关类型分组为命名空间是生成和记录类库的一种非常有用的方式。但是,此命名方案对可见性、成员访问、继承、安全性或绑定无效。一个命名空间可以被划分在多个程序集中,而单个程序集可以包含来自多个命名空间的类型。程序集为公共语言运行时中的版本控制、部署、安全性、加载和可见性提供外形结构。

1.3 .NET 应用程序的编译和执行

当编译 .NET 应用程序时,编译器不是将源代码编译成机器语言,而是将源代码编译成一种中间语言——MSIL 或 IL(Intermediate Language)。