

Visual Studio.NET

雷之宇 编著

十大经典案例



清华大学出版社

Visual Studio.NET 十大经典案例

雷之宇 编著

清华大学出版社
北京

内 容 简 介

Microsoft 推出新一代的开发工具 VS.NET, 标志着以 .NET 为核心技术的网络时代已经到来。帮助现在所有的 Windows 程序员尽快熟悉 .NET 平台, 并使用 VS.NET 快速开发基于 .NET 平台的应用程序, 是作者编写本书的目的。

本书的读者应是有过 Windows 或 Web 编程经验的中、高级程序员, 所以本书没有对 C# 或其他编程语言的语法进行描述, 而是精选出十个最具有代表性的案例为大家进行解析, 使读者可以快速进入开发阶段, 尽快掌握 VS.NET 的开发技巧。

这十个案例的内容涉及“文件操作”、“数据库管理”、“网络通信”、“多线程编程”、“XML 访问”、“混合编程”、“Web Service 编程”、“系统移植”等方面。既有基于 B/S 结构的 Web 应用, 也有基于 .NET 平台的应用程序, 每一个都是开发人员最关心和最常用的功能。

本书中的案例多数用 C# (VS.NET 所提供的, 专为 .NET 设计的编程语言) 编写而成, 每一个都是经过调试的实用程序, 读者稍加完善后就可以直接运用在自己的系统中。

最后, 希望本书的阅读能够带给您一次愉快的、神奇的 VS.NET 使用体验, 谢谢大家!

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

Visual Studio.NET 十大经典案例/雷之宇编著 —北京: 清华大学出版社, 2003.3
ISBN 7-302-06343-5

I. V... II. 雷... III. 计算机网络—程序设计 IV. TP393

中国版本图书馆 CIP 数据核字(2003)第 009472 号

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编: 100084)

<http://www.tup.tsinghua.edu.cn>

<http://www.tup.com.cn>

责任编辑: 林庆嘉

印 刷 者: 北京密云胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 17.25 字数: 431 千字

版 次: 2003 年 3 月第 1 版 2003 年 4 月第 2 次印刷

书 号: ISBN 7-302-06343-5/TP·4789

印 数: 3001~4500

定 价: 24.00 元

序

每一次技术的重大变革总都会有一批新的优秀程序员涌现出来,所有程序员都需要站在同一个起跑线上重新学习和掌握这些新的技术和技巧,每一个程序员都面临着新的机会和挑战。

而现在 .NET 的出现,标志着以网络为系统平台时代已经来临。

全新的 Visual C # .NET 正是为开发 .NET 应用程序而量身定做的编程语言,是 .NET 系统的开发利器(甚至 .NET 平台本身就是使用 C # 程序实现的)。

而面对全新的 ASP.NET 结构,以前 ASP 的程序员也可以领略 .NET 技术在 Web 方面所带来的惊喜:“面条”程序不复存在;页面与程序完全剥离;客户端与服务器端程序边界的模糊……而实现这些功能的编程语言还是 C #。

本书由案例入手,通过不同的程序案例来让大家从各个侧面对 .NET 有个全面的认识。

本书的读者对象应是有一定 Windows 或 Web 编程经验的程序员。所以,与其他同类图书不同,本书不对 C # 或 .NET 进行全面的介绍,而从实用的角度入手。所有章节都是围绕开发一个 .NET 应用中所需的各种知识而展开的。对大家感兴趣的,也是开发 .NET 应用中常用的技术、技巧则以较重的笔墨加以描述(例如:数据库操作、XML 的控制、文件系统的访问、Web Mail 的实现),希望能对读者日后开发 .NET 应用程序有所帮助。

和 Visual Studio.NET 最初的接触是在 2000 年底,当时只是带有试用性质地在 Visual Studio.NET 中编写了一些 C # 的小程序(如:黑白棋,已经修正后附在本书中)。正式、全面地对 Visual Studio.NET 进行学习和测试是因为受微软(中国)公司的邀请,作为嘉宾出席 Visual Studio.NET 发布会,并向大家介绍相关的功能,才真正静下心来对 Visual Studio.NET 中的新的特性进行逐一的验证。

编写本书的最初想法是以笔者多年的编程经验,以及对 Visual Studio.NET 的一些了解,为打算使用 Visual Studio.NET 开发程序的读者提供一些案例和指导。后来,该想法得到了笔者的两个至交好友的全力支持(她们就是本书另外的两位作者),更重要的是该想法得到了清华大学出版社的丁岭女士和林庆嘉先生的认可,想法才得以变成读者手中的这本图书。

在这本书出版之际,我想将它献给我远在欧洲的妻子孙皓。这是我的心愿也是我写作的动力之一。在她离开的这段日子,我仿佛回到了简单、苍白的单身生活,业余时间一下子变得空闲了起来,这才真正静下心来将这些年的经验加以整理并编辑成书。

同时,作者之一的符昀华已经远赴美国留学,另一位作者周烨也打算移民加拿大。想想当初大家一起在茶馆策划本书的情景,着实让人怀念。没有她们的全力配合,本书不可能完成。

本书所有源代码可在清华文泉网站:<http://www.tupwq.net/>资源下载/“VS.NET 十大案

例”中获得。运行环境(如.NET Framework, MDAC27 等)可在微软公司站点(www.microsoft.com)下载或直接与笔者联系。

最后,虽然笔者在本书的编著过程中力求完美,但由于能力所限,疏漏和错误在所难免,请广大读者不吝赐教。我的邮件地址是:gark@263.net

编者
2002 年 11 月

目 录

第 1 章 前言	1
1.1 因特网的发展过程	2
1.1.1 第一代因特网,HTML 的时代	2
1.1.2 第二代因特网,Browser/Server 的流行	2
1.1.3 第三代因特网,Web Service 与 XML 最佳组合	3
1.2 什么是 .NET	3
1.3 .NET 的系统体系结构	5
1.4 无处不在的 CLR(Common Language Runtime)	6
1.5 VS.NET 简介	7
1.6 .NET 的未来与前景	10
1.7 VS.NET 安装与配置	10
1.7.1 VS.NET 所需的系统环境	10
1.7.2 安装 VS.NET	11
1.7.3 VS.NET 基本界面元素	13
第 2 章 案例一:Web 文件管理器	18
2.1 ASP.NET 基础	18
2.1.1 ASP 与 ASP.NET	18
2.1.2 ASP.NET 程序结构详解	21
2.1.3 Web Control 介绍	21
2.1.4 File System Object 对象	23
2.2 设计思路	23
2.3 案例代码解析	24
2.4 本章小结	36
第 3 章 案例二:网络通信簿实现	37
3.1 ADO.NET 基础	37
3.1.1 ADO.NET 的最新特性	37
3.1.2 ADO 与 ADO.NET 功能对比	38
3.1.3 ADO.NET 组件	39
3.1.4 C# 通过 ADO.NET 访问数据库	40
3.1.5 数据结果的显示(Grid, List, Text)	41
3.2 设计思路	42
3.3 分类目录的创建与维护	43

3.4	个人信息的维护	45
3.5	本章小结	57
第 4 章	案例三:网络呼叫器的实现	58
4.1	Socket 编程基础	59
4.1.1	TCP 与 UDP 协议的比较与介绍	59
4.1.2	TCP 协议通信的响应流程	60
4.1.3	TcpListener 和 TcpClient 对象介绍	61
4.1.4	Network Stream 对象介绍	63
4.1.5	Thread 对象介绍	64
4.2	设计思路	64
4.3	案例代码解析	65
4.4	本章小结	91
第 5 章	案例四:Web Mail 的实现	92
5.1	Web Mail 编程基础	92
	支持 Mail 的两个类(Session、Send Mail)	92
5.2	网络安全基础	93
5.2.1	全新的 Passport 安全技术	93
5.2.2	用户身份验证	95
5.3	设计思路	95
5.4	通过 SMTP 发送邮件	100
5.5	邮件服务器的用户安全验证	103
5.6	接收、维护邮件列表	105
5.7	邮件的回复、转发	112
5.8	本章小结	114
第 6 章	案例五:Web Service 应用案例	115
6.1	Web Service 基础	115
6.1.1	Web Service 特性	116
6.1.2	Web Service 的体系结构	119
6.1.3	怎样开发 Web Service 组件	124
6.1.4	Web Service 的适用环境	125
6.2	设计思路	126
6.3	案例代码解析	127
6.4	本章小结	132
第 7 章	案例六:从 VB 6 移植到 VB.NET	133
7.1	VB.NET 基础	133
7.1.1	VB 与 VB.NET	133
7.1.2	VB.NET 中的最新特性	143

7.1.3 从 VB 到 VB.NET	143
7.2 试验环境的准备	144
7.3 将 VB 6 的应用移植到 VB.NET	146
7.4 本章小结	161
第 8 章 案例七:VB.NET 面向对象编程	162
8.1 VB.NET 面向对象基础	162
8.1.1 VB.NET 面向对象的功能	162
8.1.2 在 VB.NET 中实现继承与重载	170
8.1.3 全新的异常处理方式(Exception)	177
8.2 设计思路	183
8.3 案例代码解析	184
8.4 本章小结	186
第 9 章 案例八:VB.NET 中多线程的实现	187
9.1 多线程编程基础	187
9.1.1 程序、进程与线程	187
9.1.2 何时使用多线程	188
9.1.3 创建并启动线程	188
9.1.4 线程控制	189
9.1.5 参数传递和参数返回	191
9.1.6 线程同步	193
9.2 设计思路	197
9.3 案例代码解析	197
9.4 本章小结	207
第 10 章 案例九:企业组织结构图的实现	208
10.1 XML 基础	208
10.1.1 XML 简介	208
10.1.2 支持 XML 访问的几个类	212
10.1.3 为什么要使用 XML	212
10.1.4 使用 C# 访问 XML	218
10.2 设计思路	226
10.3 传统树型结构的表示	226
10.3.1 树型结构的存储	226
10.3.2 使用递归访问树型结构	227
10.4 使用 XML 实现企业组织结构图的优势	228
10.5 XML 文件结构的设计	229
10.6 XML 方法的实现	229
10.7 本章小结	230

第 11 章	案例十:混合编程的实现	232
11.1	混合编程基础	232
11.1.1	使用 Web Service 实现混合编程	232
11.1.2	使用 .NET 组件实现 VB.NET 和 C# .NET 的混合编程	233
11.1.3	通过 Socket 实现异种操作系统之间的混合编程	238
11.2	设计思路	239
11.3	案例代码解析	241
11.4	本章小结	268

第 1 章 前 言

计算机网络的应用以各种形式逐渐出现在我们身边的各个角落。信用卡的普及使我们可以通过网络交纳各种费用。因特网的流行使我们足不出户购买所需的任何东西。我们的生活已经和计算机网络密不可分。毫无疑问,未来的世界是以网络为中心的世界。面对这个已经或即将来临的网络世界,每个 IT(信息技术)巨人都在思考。

Sun 公司早在 1997 年就提出了网络就是计算机(Network is Computer)的概念,并及时地发布网络编程语言 Java。Java 的出现无疑极大地促进了计算机网络的发展,并一度成为网络编程语言的代名词。同时,Sun 通过许可授权的方法,允许其他公司开发属于自己的 JVM (Java Virtual Machine)。通过这样的策略,使众多 IT 巨人汇聚一堂,打造出全面 Java 产品系列,终于使 Java 成为计算机网络世界中最耀眼的明星。这些巨人中既包括许多我们所熟悉的计算机界的老大哥 IBM、Oracle、Borland 公司等,同时也使一些提供 Java 解决方案的中小型企业,在 Java 应用走天下的大潮中占据一席之地。

当然,当年 Microsoft(微软公司)也是 Java 阵营中的中坚力量,在众多研发人员的努力下,微软推出了自己基于 Java 语言的集成开发环境 Visual J++,并且在最短的时间里由 1.1 版本升到了 6.0 版本。这不仅仅是数字上的变化,集成在 Visual Studio 6.0 中的 Visual J++ 6.0 的确有了质的变化,不但虚拟机(JVM)的运行速度大大加快,而且增加了许多新特性,同时支持调用 Windows API,这些特性使得 Visual J++ 成为强有力的 Windows 应用开发平台,并成为业界公认的优秀 Java 编译器。

不可否认,Visual J++ 具有强大的开发功能,但它主要运用在 Windows 平台的系统开发中,Sun 公司认为 Visual J++ 违反了 Java 的许可协议,即违反了 Java 开发平台的中立性,使得微软离开 Java 阵营。

微软公司当然不甘心轻易放弃日趋成熟的计算机网络市场,最终于 2000 年的秋天发布了 Microsoft .NET(以后简称 .NET)(其中包含了微软新一代的编程语言 Visual C#),表明微软将以网络为中心,彻底转换产品研发的方向,改变其产品的侧重点和服务的范围。

自从 .NET 发布以来,一直就是媒体的追逐目标,大量程序员参与到 .NET 的测试和研究中,因特网上也涌现出了很多 .NET 的技术论坛和技术站点。微软董事会主席兼首席技术官比尔·盖茨及首席执行官鲍尔默在媒体上频频亮相,为宣传 .NET 作演说,鼓励全世界程序员使用 .NET 系列产品。很多商业网站已经决定采用 .NET 构架自己的应用、服务平台(如全球最大的拍卖站点 ebay.com)。

微软新一代软件开发平台,称为 Visual Studio .NET(以下简称 VS.NET),它是微软 .NET 战略产品的重要组成部分。Visual Studio .NET 集成了 Visual Basic .NET(以下简称 VB.NET)、Visual C# .NET(以下简称 C#)、Visual C++ .NET(以下简称 VC.NET)的开发环境,并可用于 ASP.NET 应用的开发。

C# 沿用 C/C++ 的语法格式(与 99% 的 C/C++ 的语法兼容)。而微软第一次统一了

VB.NET 和 VC.NET 的底层对象,使 VB.NET 和 C# 能够在同一个运行库的基础上访问相同的组件属性和方法。这使得 VB.NET 不再和以前版本的 VB 程序兼容(但令人欣慰的是 VS.NET 提供一些工具允许用户将以前的 VB 项目 Upgrade(升级)到 VB.NET)。也使 C# 可以使用类似于 VB 的简洁的语法编写高效的应用程序。在这以前,大多数人认为, C++ 程序员比 VB 程序员更高级些。在 .NET 时代,这种差别将不复存在。

.NET 平台包括用于创建和操作新一代服务的 .NET 基础结构和工具;可以启用大量客户机的 .NET User Experience;用于建立新一代高度分布式的数以百万计的 .NET 积木式组件服务;以及用于启用新一代智能因特网设备的 .NET 设备软件。

随着 .NET 的推出,一个全新的概念被引入:软件作为服务(Software as Service)。并以 Windows.NET、Office.NET、Visual Studio.NET 等为标志的产品的推出,意味着软件史上一次新的变革正在拉开帷幕。

1.1 因特网的发展过程

在微软为推出 .NET 而做的一系列宣传推广活动中,我们总是听到这样的概念,.NET 是新一代网络应用平台,并以可编程网络服务的特征使第三代因特网应用成为现实。因此,在这里,我们不得不重温一下因特网的发展过程,以进一步体味何为第三代因特网应用。

1.1.1 第一代因特网,HTML 的时代

因特网的发展是以人们无法预测的加速度在行驶着。现在,回溯从前来谈论第一代因特网,我们发现,那时的因特网所能传递给大家的信息,只是在客户端浏览器上显示的 HTML 格式的静态网页,所有的商业逻辑和数据都是存放在后端,并以类似客户/服务器的模式运行。用户从客户端发出一个 HTTP 请求,服务器端接收到这个请求并反应,产生静态的 HTML 网页传送给客户端显示结果。在这个 HTML 时代,尽管有很多遗憾和无法实现的应用,但它毕竟让我们看到了“芝麻开门”后,因特网所带来的魅力空间。

1.1.2 第二代因特网,Browser/Server 的流行

随着因特网在商业应用中的日益重要性,新的需求导致了新的技术革命。在这个时期,人们越来越依赖因特网,很多应用和服务也在从客户端向网络迁移,人们已不满足在单独的客户机上进行各种商业和娱乐活动,他们需要的是网络化的应用服务。为此,微软提出了 DNA(Distributed Internet Architecture)结构,其核心是网络和客户/服务器应用发展模型的集成,也就是人们所说的 Browser/Server 模式,将组件、动态 HTML、网络浏览器和服务、脚本、事务处理、信息安全性、目录数据库和数据存取、系统管理以及用户界面,这些服务以统一的方式公开于各个层面,供应用程序使用。这里,浏览器支持动态的网页(DHTML),以及用户接口的处理逻辑。在后端,通过组件将应用扩展到多用户环境,一些常见的商业应用被称为“三层”或“多层”应用,应用逻辑层、展示服务、商业服务层、信息检索层和管理服务层被划分

为不同的组件,通过网络直接相互通信。在这个阶段,开发人员更有效地利用各种组件来开发出满足各种需求的应用,而人们也从这些应用中感受到了科技带来的生活和工作的变化。

1.1.3 第三代因特网, Web Service 与 XML 最佳组合

经过几十年的发展,因特网的应用已经渗透到人们生活中并正在改变着人们的生活方式。对于企业来说,因特网的应用也改变了它们的商务模式。当人们从因特网应用中实现了信息共享、商业互动、在线营销等要求之后,新的要求又随之而来。

人们需要获得个性化的网络体验,需要“私人信息空间”来对网络信息进行编辑、分析和共享,这个要求对目前的网络技术来说,变得困难了。因此,在这个互连时代,可编程网络的概念被推到了网络应用技术的最前沿,现在我们想做的是,让所有的计算机群、相关设备和服务商协同工作,提供更广泛更丰富的解决方案。

为此,人们可以获得个性化的网络体验,控制在何时以何种方式传递何种信息给自己。而因特网也从基于 HTML 协议的单纯陈述飞跃到基于 XML 标准的可编程信息。

Web 服务被视为一种应用程序,它可以使用标准的因特网协议,如 HTTP 和 XML。开发人员将 Web 服务视做 Web 上的组件编程,通过 Web 应用编程接口(API),将 Web 服务集成到应用程序中。微软将这些理念融合到 Microsoft.NET 中,为人们展示了第三代因特网的发展方向,并带领人们走进“信息在你的指尖”的未来世界。

1.2 什么是.NET

.NET 的出现、发展离不开 IT 业界数年来围绕着 Internet 的起落和沉浮。微软在这一领域中曾经落后过,它一直在奋起直追,业界激烈的竞争环境和接连不断的法律纠纷,使其产品的号召力逐渐降低。为了扭转这种局面,微软必须提出一个崭新的战略和概念来重新树立大家对自己在软件行业的信心和地位。于是历时数载,耗费数以亿计的投资,成就了微软有史以来最为庞大的系统变革:Microsoft.NET。

对于.NET到底是什么这个问题,有着各种各样的说法。作为首席执行官的鲍尔默对.NET的描述应该最能代表微软公司的观点。

他说:“Microsoft.NET 代表了一个集合、一个环境、一个可以作为平台支持下一代 Internet 的可编程结构。”

确实,这句话基本上简单扼要地表述了.NET 的外在特性。

也就是说,.NET 不是某一个软件产品的名字,而是一场技术革命的代名词。.NET 的核心技术包括分布式计算、XML、组件技术、即时编译技术等。

- 分布式计算可以将计算机应用物理地分布到多台计算机上执行,这使得在网络上部署大型应用成为可能。用户对某个功能的请求并不需要知道该功能具体分布在哪一台计算机上。也就是说,分布式计算对于用户的请求是完全透明的。不仅如此,.NET还可以根据用户的请求对系统进行动态的负载均衡(load balance)。
- XML 是新一代电子数据交换(Exchange electronic Data Interface)的标准,正是由于 XML

的出现使异种网络进行数据交换变得简单。同时,由于其灵活性、安全性等显著特点,使得已经有越来越多的软件采用 XML 作为数据流转、传递的格式(例如:微软新一代的数据访问组件 ADO.NET 就是使用 XML 作为数据驱动格式)。

- 组件技术是软件技术多年来的发展成果。程序设计员可以将实现某些特定功能的程序用面向对象的技术封装成为组件,以便于这些程序代码可以重复利用;它使程序设计员从大量的 API 中解放出来,不需要为每一个应用从头开发程序。我们可以像使用积木一样轻松地使用组件搭建应用。
- 即时编译技术被应用在全新的 Web 平台 ASP.NET 上(.NET 的一个重要组成部分,详见后面 ASP.NET 的章节),这是微软 Web 技术的一项重要变革。大家知道,以往的 ASP 程序是解释型语言。也就是说,程序在用户请求的时候解释执行。解释程序的一个致命的缺陷是效率低下。因为守护进程会为每一次用户的请求解释执行程序代码。新一代的采用即时编译技术的 ASP.NET 使应用程序在第一次运行时(Runtime),根据主机的硬件和软件环境进行代码优化,并进行编译。

什么是 .NET 呢? 实质上,.NET 是 Microsoft 对所有以网络技术为核心,用于新一代多种设备互连系统产品的一个统称。

下面让我们来看看 Microsoft 对 .NET 产品的定义。

Microsoft .NET 平台包括用于创建和操作新一代服务的 .NET 基础结构和工具;可以启用大量客户机的 .NET User Experience;用于建立新一代高度分布式的数以百万计的 .NET 积木式组件服务;以及用于启用新一代智能因特网设备的 .NET 设备软件。

Microsoft .NET 产品和服务——包括 Windows.NET,连同建立积木式服务的核心集成套件;MSNTM .NET;个人订购服务;Office.NET;Visual Studio.NET。

在 Microsoft 服务器端产品中,如基于 Windows DNA 2000 技术的 BizTalk Server 2000 和 Application Center 2000,都被冠以 .NET 头衔。然而,.NET 最令人感兴趣的特色却在于它的新一代的编程语言和组件技术。

.NET 首先是一个环境。这是一个理想化的未来因特网环境,微软的构想是一个“不再关注单个网站、单个设备与因特网相连的因特网环境,而是要让所有的计算机群、相关设备和服务商协同工作”的网络计算环境。

简而言之,因特网提供的服务,要能够完成更高层次的自动化处理。未来的因特网,应该以一个整体服务的形式展现在最终用户面前,用户只需要知道自己想要什么,而不需要一步步地在网上搜索、操作来达到自己的目的。这是一种理想,但的确是因特网的发展趋势所在。

.NET 谋求的是一种理想的因特网环境。而要搭建这样一种因特网环境,首先需要解决的问题是针对现有因特网的缺陷,来设计和创造一种下一代 Internet 结构。这种结构不是物理网络层次上的拓扑结构,而是面向软件和应用层次的一种有别于浏览器只能静态浏览的可编程 Internet 软件结构。因此 .NET 把自己定位为可以作为平台支持下一代 Internet 的可编程结构。

.NET 的最终目的就是让用户在任何地方、任何时间,以及利用任何设备都能访问他们所需要的信息、文件和程序。而用户不需要知道这些东西存在什么地方,甚至连如何获得等

具体细节都不知道。他们只需发出请求,然后只管接收就是了,而所有后台的复杂性是完全屏蔽起来的。因此对于企业的 IT 人员来说,他们也不需要管理复杂的平台以及各种分布应用之间的工作是如何协调的。

通过 .NET, Microsoft 为我们提供了一种全新的开发平台,这个平台将推动以新体系为基础的协同 Web 应用开发。根据 Microsoft 的说法, .NET 至少将成为一种“对 Web 编程”的方法。当前,第一版 Visual Studio .NET 已经出现,由此出发,我们或许可以比较精确地勾画出 .NET 平台最终的全貌。

可以说,无论是在技术上还是在战略上, Microsoft 都对 .NET 寄予了厚望。当前的 DNA 2000 技术并未包含新的 .NET 平台;相反, .NET 将是一种全新的技术,有望糅合许多根本性的、深层次的创意。

作为一种全新的平台和技术, .NET 带来了许多新的产品,但这些产品和现有技术的兼容性未能获得百分之百的保证。 .NET 支持 27 种编程语言,它们将共享一组提供基本服务的类。 .NET 应用不再直接在裸机上运行,由于看好一种运行在虚拟机 CLR (Common Language Runtime) 上的称为 MSIL 的中间语言, Microsoft 放弃了 Intel x86 代码。

1.3 .NET 的系统体系结构

1. .NET 系列产品和服务

如 MSN.NET、OFFICE.NET、Visual Studio.NET、bCentral.NET 等。

2. .NET 的技术特征

.NET 包括 4 个重要特点,一是软件变服务,二是基于 XML 的共同语言,三是融合多种设备和平台,四是新一代的人机界面。这四个特点基本上覆盖了 .NET 的技术特征。

(1) 把软件当作一种服务 (Software as a service)

史蒂夫·鲍尔默在谈到软件服务时说道:“今天的软件产品仅仅是一张光盘,用户购买软件,亲自安装、管理和维护。但是软件服务是来自因特网的服务,它替用户安装、更新和跟踪这些软件,并让它们和用户一同在不同的机器间漫游。它为用户存储自己的信息和参考资料。这些就是软件和软件服务各自不同的风格。”

Orchestration 可视化编程工具产生基于 XML 的 XLANG 代码,它和 BizTalk 服务器、.NET Framework,以及 Visual Studio.NET 都曾是 Windows DNA 2000 战略的重要部分。

伴随着 ASP 产业的兴起,软件正逐渐从产品形式向服务形式转化,这是整个 IT 行业的大势所趋。在 .NET 中,最终的软件应用是以 Web 服务的形式出现并在 Internet 发布的。Web 服务是一种包装后的可以在 Web 上发布的组件, .NET 通过 WSDL 协议来描述和发布这种 Web 服务信息,通过 DISCO 协议来查找相关的服务,通过 SOAP 协议进行相关的简单对象传递和调用。

微软的 .NET 战略意味着:微软公司以及在微软平台上的开发者将会制造服务,而不是制造软件。在未来几年之内,微软将陆续发布有关 .NET 的平台和工具,用于在因特网上开

发 Web 服务。那时,工作在 .NET 上的用户、开发人员和 IT 工作人员都不再购买软件、安装软件和维护软件。取而代之的是,他们将定制服务,软件会自动安装,所有的维护和升级也会通过因特网进行。

(2) 基于 XML 的共同语言

XML 是从 SGML 语言演化而来的一种标记语言。作为元语言,它可以定义不同种类应用的数据交换语言。在 .NET 体系结构中,XML 作为一种应用间无缝接合的手段,用于多种应用之间的数据采集与合并,用于不同应用之间的互操作和协同工作。具体而言,.NET 通过 XML 语言定义了简单对象访问协议(SOAP)、Web 服务描述语言(WSDL)、Web 服务发现协议(DISCO)。SOAP 协议提供了在无中心分布环境中使用 XML 交换结构化有类型数据的简单轻量的机制。WSDL 协议定义了服务描述文档的结构,如类型、消息、端口类型、端口和服务本身。DISCO 协议定义了如何从资源或者资源集合中提取服务描述文档、相关服务发现算法等。

(3) 融合多种设备和平台

随着 Internet 逐渐成为一个信息和数据的中心,各种设备和服务已经或正在接入和融入 Internet,成为其中的一部分。.NET 谋求与各种 Internet 接入设备和平台的一体化,主要关注在无线设备和家庭网络设备及相关软件、平台方面。

(4) 新一代的人机界面

新一代人机界面主要体现在“智能与互动”两个方面。.NET 包括通过自然语音、视觉、手写等多种模式的输入和表现方法;基于 XML 的可编辑复合信息架构——通用画布;个性化的信息代理服务;使机器能够更好地进行自动处理的智能标记等技术。

Microsoft .NET 产品和服务——包括 Windows .NET,连同建立积木式服务的核心集成套件;MSNTM .NET;个人订购服务;Office .NET;Visual Studio .NET;以及用于 .NET 的 bCentralTM。

1.4 无处不在的 CLR(Common Language Runtime)

不知不觉中,我们已经被各种各样的通信设备所包围。从以前的寻呼机、移动电话到现在流行的掌上电脑(Personal Digital Assistant 以下简称为 PDA),信息家电的出现甚至可以让我们所熟悉的电视机、电冰箱也成为信息终端。

面对各种形形色色的数字设备,让我们突然认识到不仅只有计算机才可以充当因特网惟一的终端工具。而我们身边这些数字产品往往拥有众多的使用者,一旦摇身一变成了数字处理终端,计算机立刻就会变成一个专业市场。

微软会坐视这种危机在它统治的软件帝国出现吗?答案当然是否定的。微软应付这种情况的武器就是 Common Language Runtime(以下简称为 CLR)。它的全称是“通用语言运行环境”。

Common Language Runtime 这个名词大家也许有些陌生。但是,相信熟悉 Java 的朋友对 JVM(Java Virtual Machine)一定不会陌生;就像熟悉 VB 的朋友对 VBRUNXX.dll 一定不会陌生。不错,CLR 就是一个这样一个平台,能够自动管理内存分配、启动和中止线程和进程、强化安全系数的智能的运行环境。当然,CLR 的最大优势还在于它能够针对不同的终端,提供

不同版本的运行环境,使应用软件不用做任何改动和调整或重新编译就可以运行在不同的系统上。

从图 1-1 中,我们可以看到两个 CLR 的优势所在:

(1) 如同 Sun 对 Java 的期望一样,微软也提出了类似的“Code once, run anywhere”的口号。通过上图我们也不难看出,通过在不同的终端设备上提供相应的 CLR,最终可以使我们所编写的应用程序运行在各种环境下。甚至只要 CLR 提供基于 UNIX 或 Linux 的版本,我们用 VB.NET 开发的程序也可以在 UNIX 上运行了(在笔者编撰此书时,已经有一些软件厂商宣布开发 Linux 上的 CLR,相信在不远的未来,我们已经可以将部分应用移植到 Linux 上)。

(2) 不仅仅程序可以一次编写到处运行,就连编程语言也在极大程度上得到了统一。微软第一次统一了 VB.NET 和 VC.NET 的底层对象,使 VB.NET 和 C# 能够在同一个运行库的基础上访问相同的组件的属性和方法。也就是说,我们可以在 .NET 所支持的近 30 种开发语言中,使用自己所熟悉的来开发 .NET 应用。



图 1-1 无处不在的 CLR

1.5 VS.NET 简介

随着分布式系统的不断发展和使用,互操作性是一直困扰开发人员的一个主要问题,为编写应用程序,开发人员使用松散调用的软件组件,致使应用程序变得日益庞大和复杂,不同开发人员选用不同的编程语言、对象模式和操作系统,更使得部件间的互用性极大地削弱。解决这个问题的关键是语言互用性。设想一下,如果开发人员可以使用不同编程语言

编写各个组件,再将这些组件连接成一个大型的应用程序。开发人员只负责构建和出售这类 Web 服务,使其成为其他开发人员可以提取引用的庞大软件资源。这样,应用程序的功能将会变得十分强大和易于构建。而 .NET 框架的出现使得这样的问题迎刃而解。

.NET 框架通过糅合各种编程语言,极大地简化了当今开发人员对大量复杂编程的杂务处理。.NET 框架具有强大的跨语言继承性、错误处理和调试功能,可在不同的编程语言之间进行调用。从相对简单的 JScript 到复杂的 C++ 语言,所有的编程语言一律是等同的。.NET 框架的核心部件是通用语言运行(Common Language Runtime),它由三个主要组件组成:

- 类库系统,支持许多基于当前流行语言的类和操作。
- 元数据系统,在编译时,将元数据和源代码中类型信息绑定在一起,运行时,被执行系统读取。
- 执行系统,它用来驱动运行环境,执行 .NET 程序,生成元数据信息以及应用程序执行过程中的内存管理等服务。

通用语言运行可被认为是各种编程语言特性的集合体,它定义了标准类库,同时也允许开发人员定义自己的类和特定的函数,它支持面向对象编程,具有封装、多态、继承等特性,同时也支持非面向对象编程。

.NET 编程语言是符合通用语言规范(Common Language Specification)的程序语言,提供访问由通用执行引擎和一个丰富的类库组成的 .NET 平台,支持大约 20 种语言。目前微软提供 Visual Basic.NET、C#、C++ With Managed Extension、及 JScript.NET,而其他厂商也将提供支持 .NET 的语言,包括 APL、COBOL、Pascal、ML、Perl、Scheme、Smalltalk 等。

.NET 编程语言具有如下优点:

(1) 可以使用任何一种程序设计语言。.NET Framework 使得开发人员可以使用任何一种程序设计语言,而且用任何一种语言编写的应用程序能够彼此进行深度的集成,开发人员可以使用当前的开发技能而无需进行再培训。

(2) 编写更少的代码。.NET Framework 采用了高度组件化的、无需进行拼装的设计,使得开发人员可以将重点放在编写商务逻辑上。

1. VB.NET

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,你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。

Visual Basic.NET 增加了很多新的和改进的特性,如继承、重载等,使它更大限度地发挥了面向对象编程语言的设计理念。Visual Basic.NET 与 .NET Framework 和 CLR 完全整合,提供语言互用,资源回收、更可靠的安全性和更高版本支持。此外,VB 开发人员现在可以通过使用外部多线程创造可升级的多线程应用。Visual Basic.NET 的另外一些语言新特性还有异常例外处理,自定义属性和符合通用语言规范(CLS)。Visual Basic.NET 的这些新特性是利用 CLS 来实现的,它使得数据类和对象被公开调用,即任何符合 CLS 规范的编程语言都可以使用在 Visual Basic.NET 中定义的类、对象和组件,反之,作为 VB 程序员,你也可以使用任何符合 CLS 编程语言编写的类、对象和组件,而根本不用担心不同语言的数据类型。在