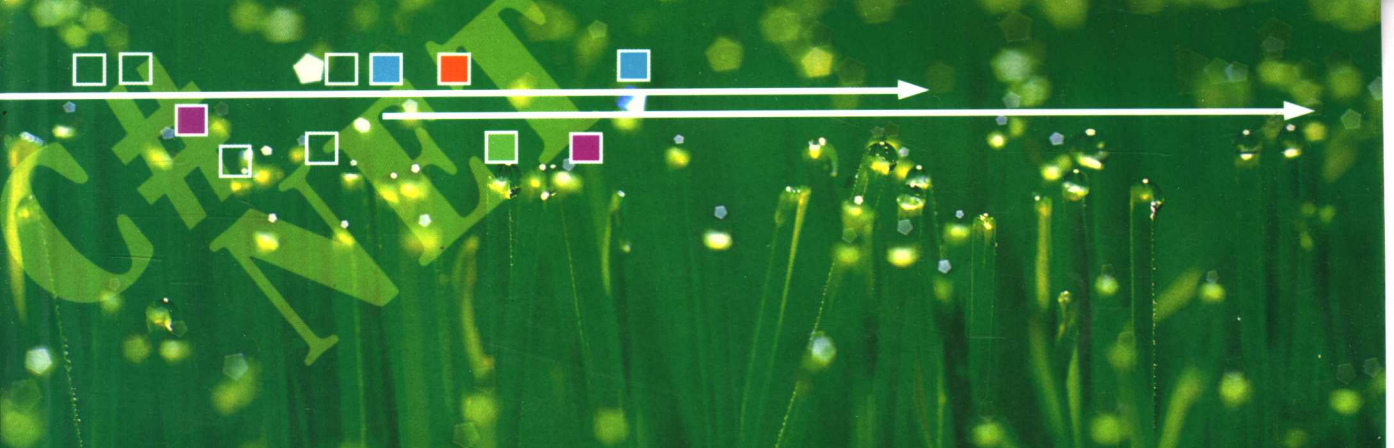




Web Service 编程

——用C#.NET 开发网络服务

北京希望电子出版社 总策划
陆昆仑 李旭东 吴媛静 编 著



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

■ ■ C# ■ NET

TP312
L862

Web Service 编程

——用C#.NET 开发网络服务

北京希望电子出版社 总策划
陆昆仑 李旭东 吴媛静 编 著



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

C# .NET

内 容 简 介

本书是一本介绍.NET 平台及使用.NET 平台开发 Web Service 的图书。.NET 是微软在 2000 年推出的新一代计算计划,它是一种面向网络、支持各种用户终端的开发平台环境。.NET 的内容核心之一就是要搭建第三代因特网平台,这个网络平台将解决网站之间的协同合作问题,从而最大限度地获取信息。

全书由 9 章组成。第 1 章介绍.NET 平台基础;第 2~4 章介绍微软新推出的 C#语言,重点讲解了 C#和 C++及 Java 的不同点;第 5 章介绍了 Web Service 的基础,并用两个实例将 C#语言及.NET 平台同后面的 Web Service 开发联系起来;第 6~8 章介绍.NET 平台上开发 Web Service 的工具 ASP.NET、ADO.NET 和 XML,以及开发.NET Web Service 必须掌握的 SOAP、WSDL 和 UDDI;第 9 章是一个旅游网的综合实例,应用了前面各章的所有知识点。

本书知识与实例结合,每个知识点都配有精心设计的实例,讲解详细,实用性强。本书不仅是具有网络开发及编程经验的程序设计人员的重要参考书,也是编程爱好者迅速了解并快速掌握.NET 的首选书籍。

本版 CD 内容为本版电子书。

盘 书 名 : Web Service 编程——用 C#.NET 开发网络服务

总 策 划 : 北京希望电子出版社

文 本 著 者 : 陆昆仑 李旭东 吴媛静

责 任 编 辑 : 周凤明

C D 制 作 者 : 希望多媒体开发中心

C D 测 试 者 : 希望多媒体测试部

出版、发行者 : 北京希望电子出版社

地 址 : 北京市海淀区知春路甲 63 号卫星大厦三层 100080

网址: www.bhp.com.cn E-mail: zwb@bhp.com.cn

电话: 010-62520290,62521724,62528991,62630301,62524940,62521921,82610344

(发行), 010-82675588-202 (门市), 010-82675588-501,82675588-201 (编辑部)

经 销 : 各地新华书店、软件连锁店

排 版 : 希望图书输出中心 杜海燕

CD 生 产 者 : 北京中新联光盘有限责任公司

文 本 印 刷 者 : 北京广益印刷有限公司

开本 / 规格 : 787 毫米×1092 毫米 1/16 20.125 印张 462 千字

版次 / 印次 : 2003 年 4 月第 1 版 2003 年 4 月第 1 次印刷

印 数 : 0001~3000 册

本 版 号 : ISBN 7-89498-124-9

定 价 : 28.00 元 (本版 CD)

说明: 凡我社产品如有残缺,可执相关凭证与本社调换。

前 言

不可否认，.NET 平台是多年来最重要的技术成果之一，是微软耗费数年才推出的重量级开发工具。而 .NET 平台开发 Web 服务是 .NET 平台的拿手好戏，本书的目标，就是教会读者如何使用 .NET 平台开发 Web 服务。

本书的重点是面向 Internet 的 Web 服务开发。在网络全球化的今天，掌握核心 Internet 编程以及成为高级网络程序员是很多程序员的梦想，本书也为此类人才提供了一个迅速提高自身认知水平的机遇。

本书不同于市面上一般的介绍 Web 服务开发的基础入门书，本书涉及到的内容是 Web 服务开发的核心内容。同时，本书也要求您已经有一定的网络开发经验，掌握 Java 或者 C++ 语言，了解一定的 Win Form 开发知识，最好能够熟练使用面向对象编程工具。但是，对于没有上述基础或者只掌握了上述一部分知识的读者，本书也可以作为您的 Web 服务开发参考用书。

也许您已经掌握了一定的网络开发技术，但是随着 .NET 平台的出现，网络开发技术又走上了一个新的台阶。而本书最及时讲解了涉及到 .NET 平台开发 Web 服务的诸多最新技术，并且力求知识性和实用性两大特点，但是更加偏重于实用性。本书的每个知识点在理论讲解之后，都会及时提供一些实例，在让您巩固所学知识的同时也让您掌握了具体开发 Web 服务的程序以及方法。这些实例由浅入深，同时附上了大量的图片，更加便于理解。通过本书的学习，在不知不觉中，您的编程水平将得到突飞猛进的提高。当您阅读完本书之后，您会在短时间之内成为一个开发 Web 服务的实战高手，就可以和其他高级编程员一样，用最新的、最强大的 .NET 工具，开发最流行的、功能完善的软件了。并且，本书也为您进一步学习 Internet 网络开发技术打下良好的基础。

Web 服务的开发繁杂而深奥，.NET 平台更是功能强大、内容丰富，在这一本书内面面俱到地讲述全部知识是不可能的，由于篇幅所限，只能重点介绍 Web 开发中的新技术以及常用技术，如微软新推出的 C# 语言，就使用了一定的篇幅给予了介绍。另外，像消息传递、内容发布、数据库连接等常用技术，也是介绍的重点！

本书由陆昆仑、李旭东、吴媛静执笔编写，宁静涛、张东、李欣、程远鹏、余元、丁燕、张翔在整理资料方面给予了作者极大帮助。在这里，我们特别感谢邵谦谦同志，他在本书的编稿以及出版工作中给予了作者极大的帮助！

由于时间仓促，加之编者的水平有限，缺点和错误在所难免，恳请专家和广大读者不吝赐教，批评指正。

目 录

第 1 章 .NET 和 .NET 框架	1	4.4 小结	100
1.1 .NET 简介	1	第 5 章 Web Service 基础	101
1.2 .NET 的产品集合和特点	5	5.1 Web Service	101
1.3 .NET 框架概述	8	5.2 Web Service 的发展前景及现有产品	109
1.4 通用语言运行时 (CLR)	12	5.3 Web Service 和相关协议规范的关系	111
1.5 装配集 (Assemblies)	15	5.4 实例: 身份验证文件传输 Web Service	111
1.6 应用域	19	5.5 实例: 创建接口相关的 Web Service 服务	121
1.7 通用类型系统	21	5.6 小结	128
1.8 元数据和自描述控件	27	第 6 章 ASP.NET 和 ADO.NET	129
1.9 跨语言互用能力和安全机制	29	6.1 ASP.NET	129
1.10 小结	31	6.2 ADO.NET	145
第 2 章 C# 基本语法	32	6.3 小结	159
2.1 如何编译 C# 语言	32	第 7 章 XML Web Service	160
2.2 值类型	39	7.1 XML Web 概述	160
2.3 引用类型	41	7.2 可扩展标记语言	164
2.4 装箱和拆箱	44	7.3 XML 开发 Web Service 实例: 上下文菜单	182
2.5 变量与常量	45	7.4 ASP.NET 和 XML 开发 Web Service 实例 (留言簿)	189
2.6 流程控制	46	7.5 小结	194
2.7 小结	52	第 8 章 SOAP、WSDL 和 UDDI	195
第 3 章 面向对象的 C#	53	8.1 简单对象访问协议 (SOAP)	195
3.1 类与对象	53	8.2 Web Service 描述语言 (WSDL)	225
3.2 构造函数和析构函数	56	8.3 通用描述、发现和集成服务 (UDDI)	248
3.3 方法	57	8.4 小结	263
3.4 域和属性	65	第 9 章 开发基于 .NET 平台的 Web 服务	265
3.5 代表、事件和索引	69	9.1 Web Service 的创建	265
3.6 继承	79	9.2 一个 Web Service 系统的实现过程	271
3.7 小结	82	9.3 小结	304
第 4 章 C# 高级编程	83	附录 A .NET 框架类库	305
4.1 名称空间	83		
4.2 错误和异常处理	89		
4.3 接口	96		

第1章 .NET 和.NET 框架

本章导读

Microsoft 为开发人员提供了一整套开发工具——.NET 和.NET 框架,使用这些工具可以快速而轻松地创建最先进的应用程序和 Web Service。

.NET 框架和 Visual Studio .NET 将为应用程序和 Web Service 提供更高的可靠性。而本章主要介绍了.NET 框架中的一些基本特征、技术特点,以区别于以往的开发工具。

本章要点

- 通用语言运行时 (Common Language Runtime): 通用语言运行时使跨语言之间设计和应用构件变得非常容易。用不同语言写的对象可以相互通讯,它们的行为可以被整合。
- 装配集 (Assemblies): 装配集用来简化应用程序的配置和解决在基于控件的应用中经常出现的版本问题。
- 应用域: 应用域提供了一个安全、通用、通用语言运行时可以使用的单元来解决这个问题。
- 通用类型系统: 通用类型系统定义了类型在运行时是怎样声明、使用、受控的。它也是运行时 (Runtime) 支持跨语言整合的重要组成部分。
- 元数据和自描述控件: .NET 框架通过允许编译器在所有的模块和装配集中声明额外的信息,使控件整合变得非常容易。这些信息就是元数据,帮助控件进行无缝整合。

1.1 .NET 简介

本节介绍.NET 的发展背景、概念、定义以及其核心技术。

1.1.1 发展背景

.NET 的产生和发展离不开 IT 业界数年来围绕着 Internet 的发展。Microsoft 在这一领域中曾经落后过,他一直在奋起直追。激烈的业界竞争和不断的官司缠身使其必须提出一个崭新的战略和概念来重新树立自己在软件业界的信心和地位,这个战略就是.NET。

众所周知,Microsoft 作为全球软件业的巨头,其地位是不可撼动的,Microsoft 从操作系统起家,在推出了里程碑式的产品 Windows 95 之后,迅速出击,开发出大量相关应用程序,一举占领了大部分软件市场。甚至有人惊呼,Microsoft 将垄断整个软件业,美国也开始对 Microsoft 的垄断行为进行调查。随后,Microsoft 继续巩固自己的阵地,并且稳步持续发展。但随着 Internet 的快速发展、成熟,Microsoft 这次被甩在了后面,甚至有悲观者认为:10 年之后,Microsoft 将成为历史,一些对 Internet 先知先觉的公司(如 SUN 等)将取代 Microsoft 的位置。

在这种情况下,Microsoft 为了重新夺取软件霸主的地位,推出了一系列新产品,重新



把 Internet 世界纳入囊中。

从 Microsoft.NET 计划的实施时间表就可以发现, 在 2002 至 2003 这两年的时段中, Microsoft 将连续发布 4 大产品, 个个都让世界动容。它们是: HailStorm, 一个包含 14 项基本用户应用的网络化解决方案; Windows.NET; Office.NET; bCentral (现在是 Microsoft 小型门户网站, 但是以后会逐步发展成为 Microsoft 的主力商业网站数据交换中心)。

这 4 个产品中, 中间 2 个宣传已久, 但是这 4 个产品之间的相辅相成、相生相克所形成的系统却是令人不得着力关注。

HailStorm 瞄准了个人的信息工具市场, 它把一个人的个人身份、日程表、通讯录、信用卡号码等与网络应用有关的项目信息, 通过 Passport 技术在各网站之间认证。就像银行信用卡一样, Microsoft 认证了个人的身份、信用和财务, 使用户能够直接从 Microsoft 的服务器中取出信息, 对银行下达付费给电子商务网站的命令。如果想在网络上方便地得到各种服务, HailStorm 是离不开的基础。

Windows.NET 和 Office.NET 两项技术的细节现在还不知道, 但是 eCompany Now 的文章说得很清楚。巴尔默说: Microsoft 将为推广 Microsoft.NET 耗资 270 亿美元。

在.NET 体系结构中, XML 是各个应用之间无缝接合的关键。与 Windows 系统捆绑 IE 浏览器而惹起官司的做法不同, Microsoft 这次全面进军 Internet, 其背景是 XML, 这是一个公开不专属于任何公司的可扩展描述语言。用美国评论界的话说: XML 就像是标准的铁轨, 任何公司都可以制造火车, 但是 Microsoft 的火车是最棒的。这是一个看似公平的竞争法则。大家在一个相对公平的平台上竞争实力, 完全符合美国基本的竞争生存理念。Microsoft 第一次放弃“具有 Microsoft 特点的标准”, 走向一个公开的标准。但是应该看到, Microsoft 正是借助这样一个开放竞争的形式, 把自己在多个领域的优势力量组合起来。

如果仔细看看最近发布的 Visual Studio.NET 和 BizTalk 就会发现, 像 Microsoft 利用自己的 VC++ 夺取了标准 C++ 市场一样, 开发工具的“Microsoft 化”奠定了 Microsoft 在市场上的领导地位。大量的用户、最大的门户网站 (MSN)、最权威的电子数据交换网站 (bCentral) 足以让 Microsoft 轻松地获得整个 Internet 市场的领导权。国内也有评论家说: 标准在谁手里并不重要, 关键是谁能够控制这些标准。就像 SUN 拥有 Java 一样, Java 虽然在 SUN 的手中, Microsoft 试图向这个标准中掺入“Windows 沙子”的计划也没有成功 (官司败诉), 但是 Microsoft 有足够的能力和金钱, 重建一个类似的系统, 实际上 C# 很快就诞生了。

从软件时代开始, 确切地说, 从 PC 时代到 Internet 时代, 每一步 Microsoft 都走得十分稳当。盖茨说过, 世界上还没有一个企业能够在两个时代辉煌, 如果有, Microsoft 是第一个。

1.1.2 什么是 .NET

很难用一句话把.NET 说清楚, Microsoft 是这样说的: “.NET 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讯。”其实简单地说, .NET 就是一个开发和运行软件的新环境, 只不过这个环境提供了许多基于 Web 的服务, 更加易于使用, 它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。这个概念仅仅指的是.NET 框架, 它是一个基础性的平台。Microsoft 以此还

派生了其他许多概念, 比如 Server.NET 等等。

Microsoft 首席执行官鲍尔默说: “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。”这句话基本上简单扼要地表述了.NET 的其他特性。

.NET 首先是一个环境。这是一个理想化的未来 Internet 环境, Microsoft 的构想是一个“不再关注单个网站、单个设备与 Internet 相连的 Internet 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, Internet 提供的服务, 要能够完成更高层次的自动化处理。未来的 Internet, 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。这是一种理想, 但的确是 Internet 的发展趋势所在。

.NET 核心上就是一大套构件库, 这套构件库既像 VCL, 又像 Java Class 也像 COM, 包罗万象, 可以找到市场上任何流行构件/类库的影子。所以说, 在 Windows 开发工具市场上, .NET 一出, 谁与争锋! 不是 VC 或 C#或 IDE 多么厉害, 厉害的是构件库, 以前得从各家的开发工具里找, 现在它已经成为操作系统的一部分, 而且是与语言无关的, 并且提供了开发所需要的所有功能, 具有良好扩展能力。因此, 开发工具再也不能通过构件体系结构获得竞争优势了。

.NET 彻底地把计算模式从单机、客户机/服务器和 Web 网站的方式转向分布式计算 (Distributed Computing)。CORBA 和 COM 是今天比较流行的部件对象计算模型。但是它们都存在着“局部计算”的局限性。也就是说, 这些模型都仅仅是本地计算或本网计算的模式, 而不能把整个 Internet 当作一个计算资源体系来加以利用。.NET 则通过一种称作 Web Service (这是.NET 的核心概念) 的技术把分布在 Internet 上的各种资源有效地通过编程手段整合在特定的应用界面中。Web Service 就相当于过去编程中常常调用的 API 函数和在面向对象编程中常用的部件接口, 只不过 API 一般存在于单个程序的不同模块中, 部件接口存在于相同机器的不同部件中, 而 Web Service 则将无所不在地分布在网络上。

.NET 还为 Microsoft 带来了一种全新的商业模式。Microsoft 预见了“服务”是数字经济的核心商业模式, 因此它将逐步转换今天依靠销售盒装软件的获利形式, 利用.NET 把“软件作为服务” (Software-As-Service)。Microsoft 已经先行一步, 把各种软件改造成为能够通过“订阅”方式使用的服务, 就像今天使用电话业务、水电和订阅杂志的方式一样, 按照“使用量”付费。为此, Microsoft 还在.NET 中构造了一系列辅助模块。这些模块包括用户认证、通知、网络存储等功能。一旦 Microsoft.NET 的各种基础构造模块开发完毕, Microsoft 将最先开始拿自己的产品做“服务”尝试。不久的将来, 软件用户将可以体验到在网络上注册使用软件并根据使用时间、存储要求或使用的功能模块多少来付费的方式。此外, Microsoft 还会把各种商业服务、内容服务与软件服务并列考虑, 这样用户在未来将不会感到“软件”与其他信息产品有任何不同, 因为人们真正需要的是快速地完成自己的工作目标, 而不是考虑“软件”等产品形式。

Java 作为一种与平台无关的语言, 配合庞大的类库, 是目前看来惟一可以与.NET 争锋的东西。但是, Java 目前主要面向服务器端应用和企业级的应用, 而这些领域并没有多少人使用 Windows 平台。而在广泛使用 Windows 平台的中低端应用上, Java 的应用则不是很广泛。Java 这样一个完全面向对象的语言, 通过小粒度划分对象、JNDI/RMI 等集成



在语言内的服务,使得 J2EE 体系特别适合描述商业对象 (Business Object) 和商业逻辑 (Business Rule)。而且由于 Java 的平台无关性,使用得它较少受硬件平台和操作系统供应商的制约,即使 Microsoft 不在 Windows 平台中提供 Java 虚拟机,对它也不构成太大影响,因为 SUN 等也可以提供。简而言之,现在一说到 Java,并不是指那种语言,而是整个 Java 体系,一个已经被实现了的优美的、完备的企业级架构。

简而言之,.NET 有三个优势:

- 语言无关
- 与操作系统紧密结合
- 体系完备,易于使用

在 COM/DCOM 时代,语言无关、与操作系统紧密结合都已具备,但是只能提供一些零散的孤立的服务,没有完备的体系。而且 COM/DCOM 的使用极其复杂,感觉就像摸到了粘着苍蝇的粘蝇纸,这显然也阻碍了 COM/DCOM 的流行。而目前业界最为完备强大规范的体系是 CORBA,可惜并没有人完全实现了该体系。Java 体系几乎与 CORBA 一样完备而规范,并且有众多厂商实现,目前被广泛应用于中高端解决方案。如今,Microsoft 在 DDE、OLE、OLE2、COM、DCOM、COM+这一系列解决方案中的努力终于达到了.NET 的境界,充分整合了竞争对手优点的体系架构,足以与 Java 体系抗衡。由于 Windows 与.NET 平台紧密结合,在 Windows 世界中 Java 与.NET 竞争没有优势,在 Windows 桌面应用市场,Java 完全无法与.NET 竞争,而 Java 和.NET/Windows 将会引发中低端应用市场的一场恶战。

Delphi 之所以有今天的成就,是源于其便利的对象模型,清晰、易用的 VCL 构件库,高度可扩展的插件体系结构以及大量的第三方控件。Delphi 主要应用在快速原型开发、桌面应用,也大量用于中低端企业应用的开发。相对于主要竞争对手 VB 而言,真正的面向对象编程语言是 Object Pascal,本地编译使它成为名符其实的 RAD 开发工具,被称为 VB Killer。由于在 Delphi 的对象模型及 VCL 架构设计时充分考虑了对 COM 的融合,使得 Delphi 能够以与使用 VCL 对象相同的方式使用 COM 对象,从本质来讲 VCL 对象就是 COM 对象。相对于 SDK 式的 COM 开发,Delphi 具有巨大的优势,相对于 VB 它能充分利用 COM 的全部能力,因此 COM 的强大与复杂进一步增加了 Delphi 的竞争力。

.NET 融合了 VCL 的优点,在发展 COM 及融合原来分散的各服务的同时,提供了清晰易用的构件库,该构件库具有与 VCL 相若的插件式结构和第三方扩展能力。并且提供既像 Java 又像 Object Pascal 又像 C++的 C#开发语言及配套开发环境,由此看来 C#及其 IDE 已经完全具备取代 Delphi 当前版本所需的特性。

.NET 作为 Microsoft 目前的核心战略,其意义并非只限于技术范畴。Microsoft 面临拆分的结局,而.NET 就是应对之策,它将担负起用技术形式将拆分后的若干小 Microsoft 捆绑成一个整体的重任。无论是应用软件还是开发工具,都将基于.NET 体系实现,通过.NET 与操作系统的紧密结合就把应用软件和开发工具的核心与操作系统捆绑在一起了。这样每一个领域里的竞争对手将要面对的并不是一个小 Microsoft,而是若干小 Microsoft 组成的混合舰队。.NET 的核心竞争力就是与操作系统的紧密结合,而.NET 战略的主要目标就是把 Microsoft 的核心竞争力由操作系统转移到.NET 上,这也就意味着.NET 完全可能被移植到其他操作系统上。通过这种方式,Microsoft 的软件可以覆盖到软件业的所有方向,从这里就可以看出 Microsoft 的一统江湖之意。

目前 Microsoft 的核心竞争力是操作系统, 与 Microsoft 的竞争实质就是操作系统的竞争, 大量应用软件领域的竞争结果已经证明这一点。想要抗衡 Microsoft, 就要从操作系统上下手。这也是当前业界巨头公司大力支持 Linux 操作系统的根本原因。Linux 操作系统抛开其开放源代码和自由软件的特性不谈, 它也是一个伟大的操作系统产品, 它既继承了 Unix 的强大与稳定, 又具备了与 Microsoft Windows 产品线一样全面的产品线, 在 Server 端极具优势。从目前来看, Linux 的弱点就是在桌面应用上, 应用软件还不够丰富, 特别是缺少像 Office、Lotus Notes、Macromedia Authorware/Director/Dreamweaver 这样广受欢迎的用户必不可少的桌面应用软件。另一个原因就是缺少像 VB、Delphi 这样强劲易用的快速开发环境。但是, 现在也有了 Kylix。所以, .NET 将是 Microsoft 核心竞争力, Linux 将是其主要竞争者。

.NET 谋求的是一种理想的 Internet 环境。而要搭建这样一种 Internet 环境, 首先需要解决的问题是针对现有 Internet 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、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。所以对于企业的 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。

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, 有望揉合许多根本性、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。.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通用语言运行时 (Common Language Runtime) 上的称为 MSIL 的中间语言, Microsoft 放弃了 Intel x86 代码。

此外, .NET 大量地运用了 XML 技术, 并赋予 SOAP 协议极其重要的地位。借助于 SOAP, Microsoft 希望将软件开发带入一个新的时代, 即不再依赖于装配组件和对象, 而是以重用服务作为软件开发的基础。SOAP 和 Web Service 共同构成了.NET 平台的基础。

1.2 .NET 的产品集合和特点

本节简要介绍.NET 的产品集合中的各个产品以及.NET 的技术特征。

1.2.1 .NET 主要组成部分及其作用

1. Windows .NET

Windows .NET 是融入.NET 技术的 Windows, 它紧密地整合了.NET 的一系列核心构造模块, 为数字媒体及应用间协同工作提供支持, 是 Microsoft 公司的下一代 Windows 桌

面平台。

2. NET Framework

.NET Framework 的目的是便于开发商更容易地建立网络应用程序和 Web Service, 它的关键特色是提供了一个支持多语言组件开发和执行的环境。

从层次结构来看, .NET Framework 又包括三个主要组成部分: 通用语言运行时 (Common Language Runtime)、服务框架 (Services Framework)、上层的两类应用模板——面向 Web 的网络应用程序模板 (Web Forms 或 Web Service) 和 Windows 应用程序模板 (Win Forms)。其中, 通用语言运行时在组件运行时, 负责管理内存分配、启动和中止线程和进程、强化安全系数, 同时还调整任何该组件涉及到的其他组件的附件配置。在通用语言运行时上是服务框架, 它为开发人员提供了一套能够被任何现代编程语言调用的、统一的面向对象、异步、层次结构的可扩展类库, 包括集合、输入/输出、字符串、图画、网络、线程、全球化、安全加密、数据库访问、调试相关服务等类库。在服务框架之上是两种应用类型的模板, 一类是传统的 Windows 应用程序模板, 另一类是基于 ASP.NET 的 Web 网络应用程序模板。其中 ASP.NET 以一组控件和体系结构的方式提供了一个 Web 应用模型, 由 .NET 框架提供的类库构建而成, 通过它可以简化 Web 应用的实现过程。

3. NET 企业服务器

在 Microsoft 宣称的“第三代 Internet”中, .NET 企业服务器是企业集成和管理所有基于 Web 的各种应用的基础, 它提供企业未来开展电子商务的高可靠性、高性能、高可伸缩性以及高可管理性。NET 企业服务器的构成异常庞大而复杂, 它共包括 8 个各司其职的服务器, .NET 企业服务器功能描述:

- Application Center 2000: 部署和管理基于 Windows 2000 之上的 Web 应用。
- Biztalk Server 2000: 用于企业间交换商务信息。
- Commerce Server 2000: 用于快速创建在线电子商务。
- Exchange 2000: 提供基于 Windows 2000 的通信和协作功能。
- Host Integration Server 2000: 为主机系统的组件集成提供方便。
- Internet Security & Acceleration Server 2000: 主要解决企业应用安全性和可管理性的问题。
- Mobile Information 2001 Server: 为移动解决方案提供可靠而具伸缩性的平台。
- SQL Server 2000: 提供完全的数据库和数据分析解决方案。

其中, 目前被关注最多的是 BizTalk Server 2000, 它提供了企业间进行电子商务所需的自动、集成的业务处理工具和架构。BizTalk Server 2000 利用 XML 作为企业内部及企业间文档传输的数据格式, 可以屏蔽平台、操作系统不同的差异, 使商业系统的集成成为可能。它主要完成的功能有: 企业间可靠的文档交换、ERP 应用集成、商业流程自动化定制、管理和监控等等。

4. 模块构建服务

模块构建服务 (Building Block Services) 是 .NET 平台中的核心网络服务集合, 它主要包括以下几个组成部分。



- Internet XML 通信：使 Web 站点转变成灵活的服务来交换和处理数据。
- Internet XML 数据空间：在 Web 上提供安全的和可编程的 XML 存储空间。
- Internet 动态更新：为快速开发和动态配置应用提供服务。
- Internet 日程安排：集成工作、社会和私人的日历。
- Internet 身份认证：提供从口令、钱包到生理数据等多级身份认证手段。
- Internet 目录服务：与 Internet 即时信息传递等服务。

5. Orchestration

Orchestration 是一种基于 XML 的面向应用的软件集成和自动化处理技术。它的目标是尽量不受时间、组织、应用及个人的限制，来最大程度和最好地把集成技术和自动处理技术结合起来，以便商业事务能够交互、动态、可靠地进行下去。Orchestration 有三个基本要求：处理与执行过程分离，即整个处理并不一定非要同执行的细节及途径绑定起来；动态处理，即随着数据不同及交换的变化，整个操作过程必须随时动态更新改变；Any to Any 集成，即整个处理过程不能对参与的平台、应用及协议等做出限制。.NET 的 BizTalk Orchestration 是上述技术的一个实现，它包括一个可视化的设计环境、一套捆绑的工具和一个 Orchestration 引擎，用于业务流程处理、管理和调试。

6. Visual Studio.NET

Visual Studio.NET 是基于 XML 的编程工具和环境，它便于快速开发符合 .NET 体系的软件服务，使其在独立设备、企业数据中心和 Internet 之间的传送更加容易。

1.2.2 .NET 的技术特征

.NET 包括 4 个重要特点：

- 软件变服务
- 基于 XML 的共同语言
- 融合多种设备和平台
- 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. 软件变服务

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 Internet 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 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 Service 的形式出现并在 Internet 发布的。Web Service 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 Service 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。

Microsoft 的 .NET 战略意味着: Microsoft 公司以及在 Microsoft 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, Microsoft 将陆续发布有关 .NET 的平台和工具, 用于在 Internet 上开发 Web Service。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 Internet 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 Service 描述语言 (WSDL)、Web Service 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 包括以下几个人机界面: 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。

1.3 .NET 框架概述

1.3.1 背景

.NET 框架 (Framework) 是能简化高度分布式网络开发的计算平台。

.NET 框架应当满足下列要求:

- 无论对象代码在本地存储并执行, 远程存储本地执行还是远程执行, .NET 框架都应当提供一个连续的基于对象的编程环境。
- 提供一个代码执行环境, 使软件配置和版本控制最方便。
- 提供一个代码执行环境, 保证代码安全执行, 包括那些由不确认和非完全信任的第三方提供的代码。
- 提供一个代码执行环境保证消除平台的差异。使开发者的经验广泛得以保存和传播, 无论是基于 Windows 的开发还是基于 Web 的开发。

- 建立符合工业标准的通讯体系，使基于.NET 框架的代码可以相互整合。

.NET 框架主要由两部分组成：通用语言运行时（CLR）和.NET 框架类库（.NET Framework）。

1.3.2 通用语言运行时

通用语言运行（CLR）时是.NET 框架的基础。可以把这个运行时（Runtime）想象为代码执行时管理代码的代理（Proxy），它提供一些核心的服务，例如内存管理、线程管理、远程调用、保证类型安全等。运行时管理的是受控代码（Managed Code），非受控代码（Unmanaged Code）不受运行时管理。.NET 框架的另一个主要部件是类库，类库是一个全面的、面向对象的可复用的类型集合，可以用它开发各种各样的程序，从传统的命令行应用（Command-Line）到图形用户界面应用（Graphical User Interface），一直到最新的由 ASP.NET 提供的新型应用，如 Web 窗体和基于 XML 的网络服务（XML Web Service）。

.NET 框架可以被非受控部件当作主机以便将通用语言运行时（CLR）调入进程并且初始化受控代码的执行，最终生成能衍生出受控和非受控应用的软件环境。.NET 框架不仅能够提供许多运行时（Runtime）主机，还支持第三方开发的运行时主机。

例如，ASP.NET 的运行时为受控代码提供了一个可升级的、服务器端的环境。ASP.NET 直接和运行时一起工作以使 Web 窗体应用和 XML Web Service 可以使用，这两部分将在这一章的后面进行讨论。

IE（Internet Explorer）是一个拥有运行时（以 MIME 扩展类型的形式）的非受控应用的例子。使用 IE 的运行时使得能够将受控部件和 Windows 窗体控制插入 HTML 文档中。通过这样使用运行时可以使受控代码可移动（类似于 Microsoft 的 ActiveX 组件）成为可能，但是有重大意义的改进是由受控代码提供的功能，例如不完全信任的执行（Semi-Trusted Execution）和安全的隔离文件存储。

图 1-1 显示了通用语言运行时、类库与应用以及整个系统之间的关系。这个图还显示了在大型的体系结构中受控代码是如何操作的。

下面是通用语言运行时特点。

1. 管理

通用语言运行时管理内存、线程执行、代码执行、代码安全验证、编辑和其他系统服务。这些特性是运行在通用语言运行时上的受控代码所固有的。

2. 安全性

为了保证安全性，受控部件被赋予信任等级验证，标准是它的出身特性（如 Internet、企业网、本地计算机）。这意味着一个受控部件可能或不能执行文件访问操作，注册表访问操作或其他敏感的功能，就算它被相同的应用所使用。

运行时强调了代码访问的敏感性。例如，用户可以信任一个嵌入网页的应用，可以在屏幕上执行动画或播放音乐，但不能访问他们的私人数据、文件系统和网络。因此，运行时的敏感特性可以使网络服务更加出色。

运行时也通过实现严格类型和构造验证（也被称为通用类型系统 CTS）保证代码的健壮性（Robust）。CTS 确保所有的受控代码是自描述的，不同的 Microsoft 的和第三方的语



言编译器产生的受控代码都符合 CTS。这意味着受控代码可以使用其他受控类型和实例，只要类型是安全的。

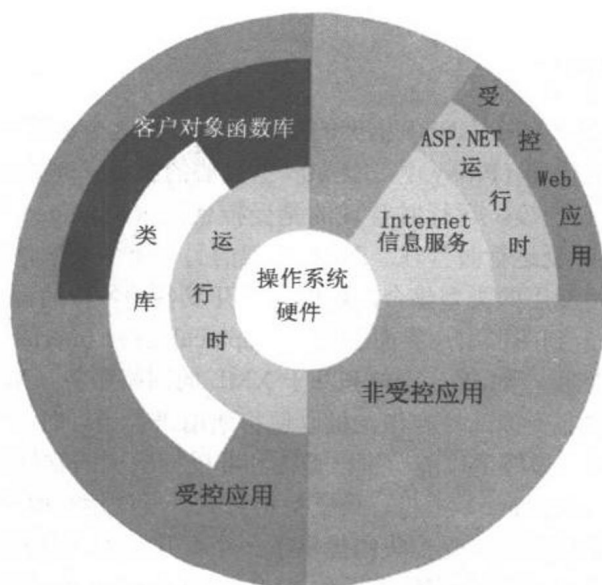


图 1-1 CLR、类库和整个系统的关系

3. 纠错性

运行时的管理环境消除了许多普通的软件问题。例如，运行时自动管理对象句柄和引用，并当它们不再被使用时自动释放，这种自动内存管理解决了两个常见的应用错误：内存漏洞和非法内存引用。

4. 快速

运行时同时也加速了开发。例如，程序员可以用他们熟悉的语言进行程序开发，并和其他使用不同语言开发的程序员共同享用运行时和类库的优势。任何选择运行时的编译器开发者都可以这样做。符合.NET 框架的编译器都支持。

尽管运行时是为了未来的软件而开发的，它也支持现在和以前的软件。受控代码和非受控代码的协同使开发者在需要时可以继续使用 COM 部件和 DLL。

运行时的设计是为了提高性能。尽管通用语言运行时提供了许多标准的运行时服务，受控代码从没有被提及。一个被称为 JIT (Just-In-Time) 的即时编译特性使所有的受控代码以本地系统的机器语言形式运行。同时，内存管理系统消除了可能的内存碎片并提高了系统的性能。

运行时可以被运行在高性能服务器端的应用中，例如 Microsoft 的 SQL Server 和网络信息服务系统 (IIS)。这些基础构造使得开发者能够用受控代码书写商业逻辑，同时享受着高性能的企业级服务。

1.3.3 .NET 框架类库

.NET 框架类库是一个和通用语言运行时紧密结合的可重用类型集合。类库是基于对

象的、提供了受控代码可以派生功能的类型集合。这不仅使 .NET 框架类型易于使用，并且减少了学习 .NET 框架新特性的时间。特别的，第三方控件可以和 .NET 框架的类轻易整合。

例如，.NET 框架 Collection 类实现了一套接口，可以使用它开发自己的 Collection 类。用户的 Collection 类可以和 .NET 框架的 Collection 类实现无缝连接。

若希望有一个基于对象的类库，.NET 框架类型可以完成一定的普通开发任务，包括字符串管理、数据收集、数据库连接和文件访问。除了这些普通任务，类库还包括了一些特殊的开发需求。例如，可以使用 .NET 框架开发下面类型的应用和服务：

- 控制台应用
- 脚本或主机应用
- Windows GUI 应用
- ASP.NET 应用
- XML Web Service
- Windows 服务

例如，Windows Forms 类是一个简化了 Windows GUI 开发的、全面的可重用的类型集合。如果要编写 ASP.NET Web Form 应用，可以使用 Web Forms 类。

1.3.4 客户端应用的开发

客户端应用是基于 Windows 编程中最接近传统形式的应用。它是这样一种应用，在桌面上显示窗体和表单，让用户执行某种任务。客户端应用包括字符处理、制表和其他一些办公处理应用，如数据登录、报告等。客户端应用一般包括窗体、菜单、按钮和其他一些 GUI 元素，并访问一些本地资源如文件系统和打印机。

另一种类型的客户端应用是 Internet 上使用的传统 ActiveX 控件(现在被受控 Windows 窗体控件所代替)。这种应用和另外的客户端应用很相似：它是在本地执行，访问本地资源，包括很多图形元素。

过去，开发者使用 C/C++ 结合 Microsoft 基础类库 (MFC) 或其他一些快速开发工具 (RAD) 环境如 VB 开发这些应用。.NET 框架整合了这些目前的产品成为一个单独的，完美的开发环境，使之更容易进行客户端应用的开发。

.NET 框架中的 Windows Forms 类被设计成被 GUI 开发使用。可以很容易地根据应用的需求创建普通的窗体、按钮、菜单、工具栏和其他一些屏幕元素。

例如，.NET 框架提供了简单的调整窗体特性的属性。有时候底层操作系统不支持直接改变这些属性，在这种情况下，.NET 框架自动重建窗体，这只是 .NET 框架的优点之一。

不像 ActiveX 控件，Windows 窗体对用户是非完全信任访问的。这就是说，二进制的或本地执行的代码可以访问用户系统的部分资源但不是全部。由于代码的访问安全性，许多以前必须安装在用户端的应用现在可以安全的通过 Web 使用。Web 应用可以像运行在本地系统上一样执行。

1.3.5 服务器端开发

服务器端的应用在受控世界里是通过运行时主机实现的。非受控应用调用通用语言运行时可以使受控代码控制服务器的行为。使用这个模型，可以得到通用语言运行时和类库



的所有特性,同时不影响服务器端的性能。

图 1-2 显示了一个基本的关于受控代码在不同服务器环境下运行的图例。服务器例如 IIS 和 SQL Server 可以在应用执行逻辑操作时提供标准操作。

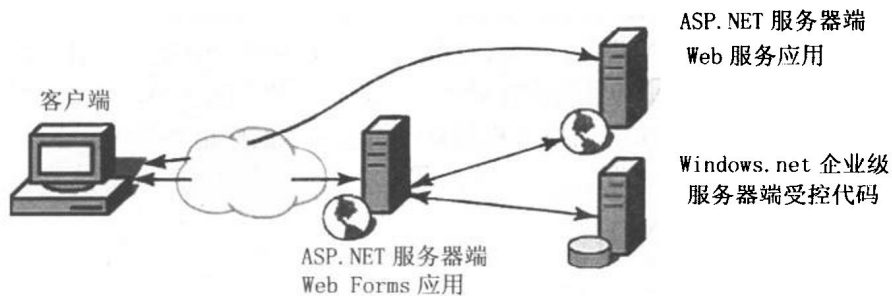


图 1-2 受控代码在不同服务器上运行

ASP.NET 是使开发者使用 .NET 框架整合 Web 应用的环境。然而,ASP.NET 不仅仅是一个运行时主机,它是一个使用受控代码开发 Web 站点和网络分布式组件的完整体系结构。Web 窗体和 XML Web Service 都使用 IIS 和 ASP.NET 作为应用的发布工具,并且都有一套 .NET 框架支持的类库。

如果使用过 ASP 的以前版本,也许很快发现 ASP.NET 的改进。例如,可以使用 .NET 框架支持的任何语言开发 Web 窗体页面。而且,代码不必和 HTML 代码放在一起,Web 窗体页面用本地机器语言执行。就像其他受控应用一样,它享有运行时的一切优势。ASP.NET 应用运行更快、功能更强、更容易被开发。

1.4 通用语言运行时 (CLR)

.NET 框架提供了一个叫做通用语言运行时的运行时环境来管理代码的执行,并提供一些使开发进程更加轻松的服务。

1.4.1 概述

编译器和工具集揭示了运行时的功能,从而可以写出符合受控执行环境的代码,用一种符合运行时规范的语言书写的代码就被称为受控代码 (Managed Code); 它有一些特殊的优点,如跨语言整合、跨语言异常处理、安全性提高、版本控制方便、控件交换方便等。

为了能够使运行时提供为受控代码的服务,语言编译器必须在代码中发布描述类型、成员、引用的元数据。元数据和代码一起存储; 每个可调用的通用语言运行时的简化执行 (PE) 文件都包含元数据。运行时运用元数据来定位类并进行调用,为实例分配内存、解析方法、生成本地代码、确保安全性、设置运行时上下文边界。

运行时自动处理对象规划、管理对象引用,当对象不再被使用时自动被释放。在生命周期内被这样管理的对象称为受控对象。自动内存管理消除了内存漏洞和其他一些常见的编程错误。如果代码是受控的,可以在 .NET 应用中使用受控的数据、非受控的数据或同时使用这两者。由于语言编译器支持它们自己的类型,例如简单类型,所以不必知道代码是不是受控的。

