

Visual C++ .NET

开发实例 完全剖析

刘廷 白晓春 编著

★
学生信息管理系统

★
房屋信息查询系统

★
卡拉 OK 点歌系统

★
图书管理与查询系统

★
飞机售票系统

从基础知识到
案例讲解的全
面突破!

新手学习信息
管理系统开发
的理想教程!

附赠超值 CD，内含整个系列图书各种语言的程序设计实例
大量的数据库结构，涵盖企业日常管理、客户信息管理、



源代码、数据库和可执行文件等，并额外赠送了视频教程和
图书馆信息管理、聊天室设计等方面。



中国电力出版社
www.infopower.com.cn

企业

信息管理系统

开发实践系列

Visual C++ .NET

开发实例 完全剖析

刘廷 白晓春 编著



中国电力出版社

www.infopower.com.cn

内 容 简 介

本书按照由浅入深、先理论后实践的原则来组织内容,详细介绍了如何使用 Visual C++ .NET 进行数据访问编程,对 ODBC、DAO、OLE DB 和 ADO 等具体的数据库访问技术进行了全面而又细致的讲解。全书共有 5 个案例,每个案例在编排和讲解上都独具特色,相信读者通过理论与实践的结合,能够快速掌握相关的数据库开发知识。

本书结构清晰,语言通俗易懂,适合于 Visual C++ .NET 初、中级用户进行学习,同时也可作为高等院校的教学用书和相关培训机构的培训教材。

图书在版编目(CIP)数据

Visual C++ .NET 开发实例完全剖析 / 刘廷, 白晓春编著. —北京: 中国电力出版社, 2005.11
(企业信息管理系统开发实践系列)
ISBN 7-5083-3873-1

I.V... II.①刘...②白... III.C 语言—程序设计IV.TP312

中国版本图书馆 CIP 数据核字(2005)第 130430 号

版权声明

本书由中国电力出版社独家出版。未经出版者书面许可,任何单位和个人不得以任何形式复制或传播本书的部分或全部内容。

本书内容所提及的公司及个人名称、产品名称、优秀作品及其名称,均为所属公司或者个人所有,本书引用仅为宣传之用,绝无侵权之意,特此声明。

责任编辑:李富颖
责任校对:崔燕菊
责任印制:李志强

丛 书 名: 企业信息管理系统开发实践系列
书 名: Visual C++ .NET 开发实例完全剖析
编 著: 刘廷 白晓春
出版发行: 中国电力出版社
地址: 北京市三里河路 6 号 邮政编码: 100044
电话: (010) 88515918 传真: (010) 88518169
印 刷: 北京市同江印刷厂印刷
开本尺寸: 185 × 260 印 张: 23.5
书 号: ISBN 7-5083-3873-1
版 次: 2006 年 1 月北京第 1 版
印 次: 2006 年 1 月第 1 次印刷
印 数: 0001—4000
定 价: 38.00 元(含 1CD)

前 言

本书详细介绍了微软新一代程序开发工具 .NET 的使用方法和特性,并结合 Visual C++ 的最新版本 Visual C++ .NET,介绍了数据库编程设计等方面的知识。Visual C++ .NET 提供了一个动态的开发环境,从而使开发人员可以利用 C++ 语言来创建基于 Microsoft Windows 和基于 Microsoft .NET 的应用程序、动态 Web 应用程序和 XML Web Services。Visual C++ .NET 包括具有工业标准的活动模板库(ATL)和 Microsoft 基础类库(MFC)、高级语言扩展以及强大的集成开发环境(IDE)功能,这些功能使开发人员能够高效地编辑和调试源代码。Visual C++ .NET 为开发者提供了面向对象的可靠语言,它可以生成功能强大同时注重性能的应用程序。通过高级模板功能、低级别平台访问和优化编译器,Visual C++ .NET 提供应用程序和组件的出众功能,使开发人员能够生成各种解决方案。C++ 是世界上最流行的系统级语言之一,而 Visual C++ .NET 为开发人员提供了生成软件的世界级工具。

本书按照由浅入深、先理论后实践的原则来组织内容。前一部分介绍了编程基础,后一部分按照由简单到复杂的顺序给出了几个具体实例,并进行了详细的讲解。具体来说,本书前面三分之二的篇幅用来介绍数据库开发的基础知识,着重于讲解如何使用 Visual C++ .NET 进行数据访问编程,对 ODBC、DAO、OLE DB、ADO 等具体的数据库访问技术进行了全面而又细致的讲解,并针对不同的技术来划分章节,使很多初级读者通过学习可以掌握相关的知识。书的后三分之一的篇幅用来介绍数据库应用的案例。全书共有 5 个案例,按照由浅入深的顺序给出。每个案例的内容在编排和讲解上都独具特色,具体包括:提出明确需求、给出解决方案、详细具体地讲解相关技术和手段。相信读者通过理论与实践的结合,尤其是通过后面几个具体、贴切实例的实践,可以快速地掌握相关的数据库开发知识。

全书分为两大部分,主要内容如下:

第一部分介绍了 Visual C++ .NET 数据库应用基础,主要包括:Microsoft .NET 简介、Windows 编程模型及 Visual Studio .NET 简介、Visual C++ .NET 应用基础、逐步认识 SQL、ODBC 基础、使用 DAO 技术开发数据库程序、OLE DB 模板以及利用 ADO 在 Visual C++ .NET 中进行数据库编程。

第二部分通过几个实例对有关数据库编程技术进行了更加深入的探讨。主要内容包括:学生信息管理系统、房屋信息查询系统、卡拉 OK 点歌系统、图书管理与查询系统和飞机售票系统。

本书适合初、中级读者和大专院校学生学习使用,也可作为刚使用 .NET 环境编程的人员的参考书。本书在整体结构上充分考虑了初级读者从零开始、按部就班进行学习的需要,做到了讲解流畅、示例充足,能够使读者轻松掌握 Visual C++ .NET 中各项数据库的操作技术。通过本书的学习,读者能够快速、高效地掌握数据库开发的常用技术,并可胜任使用 Visual C++ .NET 开发较简单但又具实战性的数据库系统的任务。

全书由刘廷、白晓春、陈明霞、崔燕、王金辉等人主持编写，由于计算机技术的迅速发展，加上作者的水平所限，时间仓促，本书中不足之处在所难免，欢迎读者批评指正。对本书的意见和建议请发电子邮件到：pcbook@263.net，我们会在第一时间给您回复。

作 者

2005 年 10 月

目 录

前 言

第 1 章	Microsoft .NET 简介.....	1
1.1	Microsoft .NET 概况.....	1
1.2	对 Microsoft .NET 的进一步了解.....	3
1.3	什么东西没有变.....	10
1.4	本章小结.....	11
第 2 章	Windows 编程模型及 Visual Studio .NET 简介.....	12
2.1	Windows 编程模型.....	12
2.2	.NET 框架.....	19
2.3	Visual Studio .NET 简介.....	22
2.4	初步了解 Visual C++ .NET.....	24
2.5	使用 Visual Studio .NET 进行开发.....	25
2.6	Visual Studio .NET 的安装.....	29
2.7	本章小结.....	39
第 3 章	Visual C++ .NET 应用基础.....	40
3.1	Visual Studio .NET 集成开发环境使用简介.....	40
3.2	怎样用 Visual C++ .NET 定制 MFC 应用程序.....	48
3.3	Visual C++ .NET 开发数据库技术的特点.....	59
3.4	本章小结.....	60
第 4 章	逐步认识 SQL.....	61
4.1	SQL 概述.....	61
4.2	SQL 数据定义.....	62
4.3	SQL 数据更新.....	65
4.4	SQL 数据查询.....	67
4.5	视图.....	75
4.6	存储过程.....	78
4.7	触发器.....	81
4.8	权限管理.....	83
4.9	本章小结.....	85
第 5 章	ODBC 基础.....	86
5.1	传统的数据库编程方式与 ODBC 的推出.....	86
5.2	了解 ODBC.....	87
5.3	使用 ODBC 数据源管理器.....	90
5.4	应用 ODBC API 函数建立应用程序.....	91
5.5	了解 ODBC 异常.....	98
5.6	MFC 访问 ODBC 数据库类.....	99
5.7	本章小结.....	104

第 6 章 使用 DAO 技术开发数据库程序	106
6.1 DAO 的内部结构	106
6.2 了解 DAO 数据库类	124
6.3 DAO 数据库应用程序的开发	130
6.4 本章小结	137
第 7 章 OLE DB 模板	138
7.1 OLE DB 基础	138
7.2 OLE DB 模板、属性	144
7.3 OLE DB 使用者模板	145
7.4 使用 OLE DB 使用者模板	158
7.5 OLE DB 提供程序模板	181
7.6 本章小结	198
第 8 章 利用 ADO 在 Visual C++ .NET 中进行数据库编程	199
8.1 了解 ADO 编程	199
8.2 使用 ADO 编程的几种方法	209
8.3 利用 ADO 实现一个数据库编程实例	213
8.4 本章小结	217
第 9 章 学生信息管理系统	218
9.1 系统概述	218
9.2 数据库设计	219
9.3 系统模块的设计与开发	221
9.4 本章小结	239
第 10 章 房屋信息查询系统	240
10.1 系统概述	240
10.2 数据库设计	241
10.3 系统模块的设计与开发	242
10.4 本章小结	261
第 11 章 卡拉 OK 点歌系统	262
11.1 系统概述	262
11.2 数据库设计	263
11.3 系统模块的设计与开发	264
11.4 本章小结	282
第 12 章 图书管理与查询系统	283
12.1 系统概述	283
12.2 数据库设计	285
12.3 系统模块的设计与开发	288
12.4 本章小结	312
第 13 章 飞机售票系统	313
13.1 系统设计概述	313
13.2 数据库设计	316
13.3 系统模块设计与开发	319
13.4 本章小结	369

第 1 章 Microsoft .NET 简介

进入 21 世纪,人们普遍使用通过 Internet 连接的多个应用程序的 PC 机以及各种各样可以互联的设备,而抛弃了那些没有联网的单台 PC,这样的变化源于 Internet 带给我们的梦想。在这样一个网络普及的年代,我们无论身处何地,无论使用何种设备,都能够随时通过 Internet 获得信息。我们希望能把从多个来源获得的数据合成为可以直接使用的、有意义的信息;期望能够从容地、长时间地使用同一种软件而不必忙于不断地安装和升级;还期望这一切远远比过去的 PC 机稳定而且可靠,而不再需要无休止的维护。

由于大多数 Internet 应用程序都是以数据库为中心的,所以方便地访问多种数据源的能力比以往任何时候都得更加重要。但是在软件开发过程中很多时候都存在“不兼容”的现象,比如:用不同的语言编写的模块很难集成,为不同操作系统开发的应用程序各自基于互不相容的 API,使得移植非常困难等等。现在软件行业的焦点已经从客户端——服务器方式的应用程序转向了运行在 Web 上的应用程序,于是新的不兼容性产生了:基于 Web 的应用程序混和使用了 HTML、DHTML、XML(可扩展标记语言)、COM、ActiveX 等彼此互有差别的技术。

现在,微软公司已经开始着手解决这些令人头疼的技术问题,解决的方法具体体现在其 .NET 计划中。

1.1 Microsoft .NET 概况

2000 年 6 月,Microsoft 公司推出了“Microsoft .NET 下一代互联网软件和服务战略”;同年 9 月 Microsoft 公司发布了 Enterprise 2000;11 月 Microsoft 公司又发布了 Visual Studio .NET 软件(包括 C#、VC++ .NET、VB .NET、ASP .NET、VBScript .NET、JScript .NET 等)。.NET 是一种全新的平台,它将对现有的所有代码产生影响。比尔·盖茨曾说过:“微软所有的产品都会被涉及。我们的整个战略是围绕这个平台展开的。”

现在很多人因为没有听说过或是不了解微软的 .NET 战略而有一种被时代抛弃的感觉,仿佛一夜之间自己又与别人拉开了很大的距离,被远远地抛在了后面。根据聊天或阅读中得来的信息,有人心目中的 .NET 或许是一个针对整个基于分布式结构的 Internet 的宏伟建筑,有人或许会认为它仅仅是 Web 应用程序的别名而已。但是有一件事倒是越发清楚了,那就是大家对什么是 .NET,还有 .NET 对 Web 和应用程序开发意味着什么都急切希望有一个深入的了解。

Microsoft .NET 是 Microsoft XML Web services 平台。XML Web services 允许应用程序通过 Internet 进行通信和共享数据,而不管所采用的是哪种操作系统、设备或编程语言。Microsoft .NET 平台提供创建 XML Web services 服务并将这些服务集成在一起。

.NET 带给我们的好处之一是,它不仅允许应用程序直接访问和利用基于 Web 的远程服务,而且使得 Web 服务器可以很容易地向外部的程序提供这种服务。但这并不意味着客户端

要利用 Web 服务都非得要用 .NET 不可,只不过是利用 .NET 平台来进行这些开发更加便利而已。

尽管现在要准确地指出什么是 .NET 还很困难,但是如果通过历史和整个软件业的发展历程来看待 .NET,我们或许会更容易了解 .NET。就如同 20 世纪 80 年代初计算机的普及率还不是很高、人们所用的操作系统大多是当时流行的 MS-DOS 一样,互联网一段时间内大部分的作用还只是给浏览者提供静态的信息,让浏览者搜寻自己需要的东西。MS-DOS 被淘汰是因为它只是个严格的命令行的、基于文本的操作系统,它的方便快捷远不如图形化的环境,同样下一代互联网肯定会淘汰现在互联网的这种结构,因为人们有了整体服务的需求。在交互式的服务下,用户提供信息给 Web 站点,这些信息在 Web 站点中被交给应用程序逻辑来处理然后产生一个结果。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。

从最底层开始,.NET 的设计就是以适应 Internet 的需要为目标的,它融合了各种 Internet 标准。它对 Web 服务的支持,使程序可以方便地从多个站点上获取数据;广泛运用 XML 语言,使得不同计算机系统之间的数据互换变得简单易行。另外,.NET 还能使用户更容易地访问数据,更容易地编写出能在众多不同浏览器和设备上良好运行的 Web 应用程序,而不必再为每种设备或浏览器分别编写程序。

.NET Runtime 支持简化和可靠的安装和升级,使用更简便,可靠性更好。它和 .NET Runtime 的安全性及版本管理功能的结合也提高了可靠性,并使软件可以作为一种服务而不必产品来出售。

正如上面所提到的那样,Microsoft .NET 解决了下面这些当今软件开发中的一些核心问题:

(1) 互操作性(Interoperability)、集成性(Integration)和应用程序的可扩展性(extensibility)太难实现而且代价很高,Microsoft .NET 依靠 XML(一个由 World Wide Web Consortium(W3C)管理的开放标准)消除了数据共享和软件集成的障碍。

(2) 无数具有相当竞争力的私有软件技术使得软件的集成变得非常复杂,而 Microsoft .NET 建立在一个开放的标准上,它包含了所有编程语言。

(3) 当终端用户使用软件时,他们总觉得不够简便,有时甚至感到很沮丧,因为他们无法在程序之间方便地共享数据或是对能访问的数据进行操作,XML 使数据交换变得容易了,并且 .NET 软件可以使用户只要一得到数据就可以对它们进行操作。

(4) 终端用户们在使用 Web 的时候,无法对自己的个人信息和数据进行操作,这导致了个人隐私和安全泄漏问题,而 Microsoft .NET 提供了一套服务,使用户可以管理他们的个人信息,并且控制对这些信息的访问。

(5) .COM 公司和 Web 站点开发者们很难为用户们提供足够的有价值的信息,至少有一部分原因是由于他们的应用程序和服务无法很好地和其他程序和服务合作,只是一个不和外界连接的信息孤岛,而 Microsoft .NET 的设计宗旨就是为了使来自于多个站点和公司的数据或服务能够整合起来。

Microsoft .NET 将从根本上改变我们思考和使用电脑的方式。目前“服务器”和“桌面电脑”这两种概念占据了计算领域的统治地位,然而 Microsoft .NET 是一种分布式计算范例,它消除了传统上服务器和桌面电脑的区别,取而代之的是计算的处理被放在最合适的地方进



行，它可能是服务器、PC，也有可能是手提电脑以及其他智能设备。

.NET 的计算模型对商务和终端用户都产生了重要影响，但方法不同。对终端用户来说，这个新计算模式更具个性化、综合程度更高，会给他们带来一种史无前例的新体验。对商务来说，这个模式改变了制造和销售软件的方法，使 IT 成为一个公司成功的重要贡献者，并建立起新的商务模型。

当然，.NET 的设计不只是停留在 Internet 接口这个层面上。访问远程服务的这种编程模型同样也被应用在本地服务的利用与内部事务逻辑上。换句话说，这个编程模式将遍布在程序设计中。其目的在于提供一种统一的、面向组件的编程模型。无论你想显示一个远程的翻译服务还是本机的文件系统或者是一个对话框，所有的支持都将是一个一般化的基于组件的接口：属性、方法和事件。支持 .NET 的编程语言可以容易而且融洽地使用这些组件，而不管这些组件是用什么样的语言开发的，而且协作者们再也不会意识到自己的伙伴或许正在使用另外一种编程语言。.NET 其实是在引出一种全新的开发应用程序的思考方法。

1.2 对 Microsoft .NET 的进一步了解

看了上一节，现在你可能开始对 .NET 是什么有了一个想法。Internet 的应用正在不断地扩大，只需要有一个 IP 地址、一台服务器和一些有意义的内容，我们就可以建设一个网络站点，在允许范围内的任何人都可以通过 Internet 来访问这些内容，这样 Internet 就成了一个卓有成效的新闻传播媒介。除此之外，它也是一个极具创造力的研究工具，因为，网站不仅提供可以浏览的内容，还可以更进一步地提供更加丰富的用户界面，也就是说，站点将是可编程的。.NET 将远程服务器所提供的计算能力和允许用户交互操作所必需的通信结合了起来。

为了充分认识 Microsoft .NET 平台的重要性，很有必要研究这种体系结构变化、它的起因以及对计算机界所造成的持续影响。

1. 结构变化

当 Internet 技术特别是 Web 在 20 世纪 90 年代中期成为计算主流时，业务计算的模型发生了巨大变化。这一变化的焦点是客户端/服务器计算的行业观念，直到那时它仍然是非常复杂、昂贵和专有的。

可以认为 Web 模型是由各种信息和应用程序集合所组成的层松散地连接而成的，而这些信息和应用程序又驻留在种类繁多的硬件平台上。记住，Internet 背后的驱动力从一开始便是希望提供一个可伸缩、可扩展和高度可用的公共信息传送平台的。这一平台设计上非常灵活，而且不局限于一个或两个计算层。Internet 世界中应用程序开发的唯一真正的限制是计算机的容量和应用程序设计者的想像力。

随着 Web 浏览器迅速变得无所不在和 Web 服务器在大小公司中的激增，非常明显的一点是：尽管客户端/服务器软件生产商一直在尽最大努力使其产品具有 Web 功能，但是开发人员仍需要采用一种截然不同的方式来思考应用程序模型。开发人员首先想到的是对业务计算采用“最低公共点”方法。显然，这要求有新的技术和工具的出现来迎接技术变化和开发人员所面临的挑战。

2. 技术变化和开发人员面临的挑战

一旦 Internet 革命成功并且出现了新的技术, 开发人员即面临现有设计模型和工具无法充分解决的若干个挑战。以下是使开发人员处于困境的核心问题:

- (1) 异类环境。
- (2) 可伸缩性。
- (3) 快速应用程序开发和部署。
- (4) 平台操纵和管理。
- (5) 具有网络功能的应用程序。

3. 异类环境

最先也可能是最大的挑战之一是生成能够轻松适合异类环境的应用程序的需要。大多数的大型机构都具有各种各样的终端、胖客户端和瘦 (Web) 客户端。除了适应客户端基之外, 新的应用程序将不得不和大型机和中型机上承载的旧式数据和应用程序进行交互, 而这些计算机通常来自不同的硬件供应商。

4. 可伸缩性

在引入 Internet 技术之前, 可伸缩性是一个相对容易管理的问题。在那时, 计算环境基本上是一个封闭系统, 原因是人员、客户或业务伙伴所进行的远程访问的数量受到限制。这意味着对于给定的应用程序和服务, 用户的数目和他们的使用模式是人们所熟知的。进行战略规划的人员有着丰富的历史数据, 他们能够据此进行计算环境伸缩设计以满足用户的需求。

另外, 应用程序开发周期一般要跨越若干年。因而, 规划者有充足的时间对系统和应用程序的伸缩进行规划。

最后, 个人计算机仍然没有充分发挥它们的潜力, 所有公司刚开始在他们的整个企业里部署个人计算机。实际上, 许多人把个人计算机视为某种比终端聪明一些的东西。随着时间的推移, 人们期望个人计算机会成为任何给定应用程序的组成部分。

尽管个人计算机重新定义了人们的工作方式, Internet 技术, 尤其是 Web 却改变着企业的心态。最初, 企业把这种新技术视为一种在组织中实现信息共享的理想而成本低廉的方法。它不但便宜, 而且还使用户能够很容易地进行他们自己的开发, 同时, 内部 Web 站点 (Intranet) 也迅速出现在计算领域中。

可伸缩性规划的传统基础开始受到破坏, 并在公司向外部世界开放的时候彻底崩溃了。

5. 快速应用程序开发和部署

Intranet 和 Internet 现象突出显示了快速应用程序部署的可能性和需要。公司 Intranet 的经历清楚地表明你可以快速生成业务应用程序。增加的附带优点是基于 URL 部署的简单性。结果是业务经理和用户们开始怀疑整个的传统开发平台和过程。他们不再准备在能够使用应用程序之前等待几个年头。从投资角度来看, 业界对在一完成即会成为旧式系统的应用程序中所进行的任何投资均持怀疑态度。

当业界在将他们应用程序的范围从 Intranet 扩展到 Internet 时, 进一步改善了快速应用程序开发的观念。为了获得竞争力, 开发人员需要创建几乎能够满足即时使用需要的应用程序, 即实时 (JIT) 开发。为了实现这一点, 它们需要彻底修改他们的应用程序并使其具有新的活力。



6. 平台操纵和管理

和计算机技术的任何方面一样，在 Internet/Web 世界中事物永远不是完美的。接受这一新应用程序模型的信息技术（IT）专家发现，他们刚获得的自由和灵活带来了一组全新的操纵和管理问题。这些问题围绕着客户端、应用程序和主机出现。

从无到有的浏览器没有给大多数机构提供浏览器标准（日常支持和升级问题本身对于后勤来说就经常是一个噩梦）。从开发的角度看，缺乏标准化意味着应用程序设计人员必须适应每个浏览器版本的核心和扩展的 HTML 呈现能力。

应用程序部署甚至更加难以管理，原因是系统管理员不得不与大量的出版商进行争论。由于业务接受了提供数据驱动的动态内容的思想，基于 Web 计算的这个方面的管理变得愈加困难起来。包含各种数据存储和适应若干种不同脚本撰写语言的需要进一步扩大了 Web 编程模型的范围。

任何使用过早期基于 Internet 的业务应用程序的站点管理员都可以证明，即使是使中等大小的站点保持正确而又连续的运行也需要大量辛苦的手工工作，因为 Internet 现象的另一个方面是用户希望能够 24 h 进行访问，并且每天都能访问。而添加服务器以适应 Web 站点上增加的通信量会增加支持的需要。

不幸的是，Web 的设计人员和支持者没有将管理平台的工具集包括进来，却让 IT 界来开发解决方案。

7. 具有网络功能的应用程序

开发人员要面对的最后一个问题是：必须适应便携式计算机技术的进步和便携式计算机（如膝上型、笔记本和掌上型计算机）成本的降低所带来的变化。随着 Internet 将全球访问变为可能，移动计算已经以近似于 Web 的速度发展起来。最近的数字表明膝上型电脑的销售已经超过了桌面电脑的销售。

脱机或者断开使用不再是特例。用户社区期望既能够以联机方式也能够以脱机方式使用应用程序和服务。应用程序开发人员必须能够在应用程序中提供这种功能。

.NET 允许我们在服务的层次上而不是在发布的层次上来共享信息并交互。Microsoft 所有支持 .NET 的产品和许多来自第三方的组件都具有 .NET 的内部结构。那么，是什么使 .NET 能够像我们所描述的那样工作呢？

这些都要从 XML 开始。

1.2.1 用 XML 实现数据编码

在过去的几年里，两个最关键的进展就是引入了 XML 及其在 SOAP 中的应用。这些技术都是 .NET 的核心。要成为 .NET 开发世界中的佼佼者，就必须理解 XML 和 SOAP。像我们在自己的交流中所说和写的语言一样，XML（eXtensible Markup Language）是一种可扩展标记语言，它由 SGML 语言演化而来。XML 的设计思想是用来描述数据的，它的标签不是事先定死的，而是根据实际需要定义的。XML 可以自我描述并用文档类型定义的方式来正式地定义、约束 XML 文档。

在 .NET 结构中，XML 是多种语言、多种应用无缝结合的一种手段，是 .NET 的基础。基于 XML 的共同语言是 .NET 的一大亮点。也许我们有很巧妙的思想和重要的信息可以分享，但如果我们不能以一种别人可以理解的方式来表达我们的思想和信息，即使付出艰苦的

工作也只能事倍功半。XML 是 .NET 的灵魂，是所有 .NET 的基础。数据库将通过 XML 中的记录集来读写，Web 浏览器将接受 XML 并将其和伴随它的样式表一起显示，Visual Studio 甚至会产生 XML 代码。不理解 XML 和与之相关的技术，就不可能同支持 .NET 的资源进行交流。

HTML 的出现无疑是对 Internet 技术和 Web 技术的一次突破，它第一次使人们能够在 Web 上浏览和显示多种格式的数据，为推动 Internet 和 Web 技术的发展发挥了巨大的作用。可以说，如果没有 HTML，Web 技术就不可能发展到今天。

然而，随着 Web 技术的飞速发展，Internet 上的 Web 信息越来越多，内容越来越复杂，数据格式也越来越多，传统的 HTML 有限的标记功能已经无法满足表达日益丰富的数据形式的需要。在这种背景下，XML 技术应运而生。

1. HTML 的问题

由于 Web 技术的飞速发展，传统的 HTML 存在着如下的问题和不足：

(1) HTML 的标记 (tag) 集合是固定的。随着 Web 技术的飞速发展，新的数据格式不断产生并需要在网上展示，这就要求有一种比较灵活的标记机制来满足不断发展的 Web 内容的要求。但标准的 HTML 语法规则过于简单，又不允许用户根据自己的需要来创建新的标记，所以它无法支持那些专门的页面格式，例如：数学公式、化学方程式、音乐乐谱、财务报表以及工程应用等。

(2) DHTML 带来的问题。由于标准 HTML 已经无法满足用户的需求，人们在其基础上增加了动态的成分，如脚本程序等。但是，这些非标准技术制作的网页在不同的浏览器之间互不兼容。

(3) HTML 只是一种表现技术，它并不能揭示 HTML 标签所标记的信息的任何具体含义。例如：语句<h1> Sun </h1>表示在 Web 浏览器中用标题 1 显示文本“Sun”，但 HTML 标记却没有表明“Apple”究竟代表什么意思，它可能是指恒星太阳，也可能是一个公司的名字，或者是一个别的什么东西。HTML 当初在制定规范时并没有考虑到这方面的功能。

另外，随着 Web 文件变得越来越大、越来越复杂，Web 内容提供商已经开始感受到普通的 HTML 已经无法提供用于大规模的商业出版所需要的扩展性、结构化和数据检查功能。

2. XML 的产生

在 1996 年 8 月，那些关心 SGML 的专家聚集在美国西雅图，成立了一个名为 GCA (Graphic Communications Association，图形通信协会) 的组织。在这次会议上，GCA 探讨了 SGML 对 Web 技术的影响。

讨论主要集中在如何开发 SGML 以便使它适应和促进 Web 技术的发展上。但在 1996 年 8 月，在 Internet 上流行的是 HTML 技术而不是 SGML 技术。虽然 SGML 是一个设计严格的完整的系统，但是 SGML 标准并没有考虑其在软件应用中是否易于使用。为了使 SGML 被更多的人理解和接受，就必须对 SGML 过于复杂、难于被理解和实现的方面进行简化。专家们迅速着手对 SGML 进行精简，去掉其语法定义部分，适当简化 DTD 部分，并增加了部分互联网的特殊成分。为了体现它与 HTML 的不同，工作组将其命名为 XML (eXtensible Markup Language)，同时也将自身更名为 XML 工作组。1998 年 2 月 10 日，XML 工作组正式向 W3C 提交了 XML 的最终推荐标准，这就是 XML 1.0 标准。

经过几年的发展，XML 已经被广大编程人员、数据库开发者和文档编写人员所接受，



W3C 所支持的另外两种语言：SMIL（Synchronized Multimedia Integration Language，同步多媒体接口定义语言）用于更好地将音频和图像同步化以便于进行 Web 传输；MathXL（数学标记语言）用于显示和处理数学公式。它们都是以 XML 为基础的。

今天，XML 已经逐渐成为整个 Web 的基本结构和未来各种发展的基础，由于 XML 能针对特定的应用来定义自己的标记语言，这一特征使得 XML 可以在电子商务、政府文档、报表、司法、出版、联合、CAD/CAM、保险机构、厂商中提供各具特色的独立解决方案。

3. XML 的优缺点

XML 的优势之一是它允许各个组织、个人建立适合自己需要的标记集合，并且这些标记可以迅速投入使用。

XML 的最大优点在于它的数据存储格式不受显示格式的制约。一般来说，一篇文档包括 3 个要素：数据、结构以及显示方式。对于 HTML 来说，显示方式内嵌在数据中，这样在创建文本时，要时时考虑输出格式，如果因为需求不同而需要对同样的内容进行不同风格的显示时，要从头创建一个全新的文档，重复工作量很大。此外，HTML 缺乏对数据结构的描述，对于应用程序理解文档内容、抽取语义信息都有诸多不便。

XML 把文档的三要素独立开来，分别处理。首先把显示格式从数据内容中独立出来，保存在样式单文件（Style Sheet）中，这样如果需要改变文档的显示方式，只要修改样式单文件就行了。XML 的自我描述性质能够很好地表现许多复杂的数据关系，使得基于 XML 的应用程序可以在 XML 文件中准确高效地搜索相关的数据内容，忽略其他不相关部分。XML 还有其他许多优点，比如它有利于不同系统之间的信息交流，完全可以充当网际语言，并希望成为数据和文档交换的标准机制。

当然，XML 作为一个新建立的标准，还有许多不足之处：它在强调了数据结构的同时，却在语义表达能力上略显不足，例如，定义了<地址>这样一个标记，如果不是在文档中实际定义内容，我们就无法知道是要表达家庭住址还是 E-mail 地址。

另外，XML 的有些技术尚未形成统一的标准，充分支持 XML 的应用处理程序很少，甚至浏览器对 XML 的支持也是有限的。

所以，XML 还并不能完全取代 HTML，毕竟 HTML 是最为方便、快捷的网上信息发布方式。况且 HTML 是描述数据显示的语言，而 XML 是描述数据及其结构的语言，二者在功能上也是截然不同的。

对今天的 Web 服务器而言，XML 差不多无所不在。几乎所有的计算平台都能分析 XML，因而也就能获得 XML 文档中的内容。所以，如果我们能找到一种方式从远程系统获得 XML 文档，问题就剩下如何理解文档中包含的数据了。这时，我们就需要通过 SOAP 来实现数据通信。

1.2.2 用 SOAP 实现数据通信

长期以来我们使用超文本传输协议 HTTP 来提供 Web 页面以及往来的内容。但当我们将来将 HTTP 或一些其他 Internet 传输协议同 XML 结合起来，并指定 XML 文档自身的格式时，就得到了简单对象访问协议 SOAP。至少在开始构想它时，SOAP 是被设计为从本地系统向远程系统传递远端方法调用的。基于 SOAP 的结构与同时代的其他远程结构比如 DCOM、CORBA 和 RMI 等等所不同的是，SOAP 协议可以穿越任何团体的防火墙，并且 SOAP 数据

包中包含着以 XML 编码的数据，它们比较易于分析和使用。SOAP 还有很好的伸缩性，这使得我们能同时为非常多的用户服务。

SOAP 模型最初的构想是使用请求—响应模型，同我们今天所用的 Internet 计算模型很相似。随后，SOAP 发展到包含了消息模型。两者的不同之处是 SOAP 在对远端系统上的方法参数进行编码时，有获得结果的特殊目的。它并不请求 Web 站点提供一个感兴趣的数据表格，相反，比如说，在同样的系统上用户可以调用一个假想称为 CalculatePayment() 的远程调用，并收到一个个人付款数值。虽然今天你能用一个表单做到这些，但关键是在调用服务和提交表单之间存在着差别。服务调用是功能更强的概念。

SOAP 技术与现实中的 B2B 应用有着很密切的关系。以下是对于 SOAP 技术各个组成部分的一些总结。

(1) SOAP Message (SOAP 消息) 是用于传输应用入口调用信息的承载体。从根本上来看，SOAP Message 是从发送方到接受方的一种传输方法。在常见的 B2B 应用中，SOAP 消息是 B2B 应用调用信息的基本承载体。

(2) SOAP Envelope (SOAP 信封) 用于描述 SOAP Message 的结构，定义了一个以 SOAP Header、SOAP Body 为子主体的描述结构。SOAP Envelope 是 SOAP Message 的信息框架。在 B2B 应用中，SOAP Envelope 为 B2B 应用调用信息提供了一个基本的信息描述框架。

(3) SOAP Header 提供了一个可伸缩的机制用于在分散的模块化中扩展 SOAP 消息，而通信双方并不需要预先的约定知识。在 B2B 应用中典型的扩展例子可以是实现一些诸如认证、事务管理以及支付的 Header 条目。

(4) SOAP Body 元素提供一个简单的用于与消息的最终接收者交换必须处理的信息的机制。在 SOAP Body 中可以描述应用入口调用和响应的各种数据信息。在 B2B 应用中，Body 元素的应用可以是 B2B 应用调用请求信息（如产品 ID、产品数量等）、响应信息（如订单号、订单履行日期）及错误信息（如产品库存不够等）。

(5) SOAP Encoding (SOAP 遍序方法) 是基于一个简单类型的系统，而这个系统是程序语言、数据库和半结构数据中类型系统的公共特性的泛化。默认的 SOAP Encoding 定义和 XML Schema 是相容的、基本一致的。SOAP Encoding 使类似整数 (integer)、字符串 (string)、数组 (array)、结构 (struct) 等数据类型得以在 SOAP 消息中表示。SOAP Encoding 为 SOAP Header 和 SOAP Body 中的数据表示提供了默认的方式，当然你也可以使用其他任意基于 XML Schema 的型/值描述系统。

(6) SOAP RPC Presentation (SOAP RPC 表示) 是一个用于表示远端过程调用和响应的约定，利用 XML 的可扩展性和可伸缩性来包装和交换 RPC 调用。目前通过绑定各种已有的 Internet 协议，譬如 HTTP、SMTP、POP3 等，利用这些协议的调用响应机制可完成 SOAP 的调用和响应。具体地说，SOAP 消息会作为这些协议的正文被发送。也就是说将 SOAP 协议包上一个 HTTP 的外壳，在 HTTP Network 里面传输，当然 SMTP、POP3 也是一样。SOAP RPC Presentation 是最终面向传输的绑定机制。

1.2.3 .NET：数据共享平台

由于 Web 技术正在迅速成为支持企业范围应用程序的优选平台，开发和承载应用程序所需的基础结构在规模和复杂性方面都已经增长。.NET 平台是计算机应用程序结构在 20 世纪



90 年代发生主要变化的直接结果。

.NET 用 XML 语言和 SOAP 的通信方式,就能建立允许交互和合作的系统。Microsoft 建立的这种具有强大功能的结构将改变我们使用计算机和编程的习惯,就像在过去的 20 年里 Windows 取代 DOS 那样。所以未来的 Internet 是可以提供更多增强服务的 Internet。

微软 .NET 计划的体系结构其实是一个可以作为平台支持下一代 Internet 的可编程结构。作为一个理想化的未来互联网环境,.NET 不再关注单个网站、单个设备与因特网相连的网络环境,而是要让所有的计算机群、相关设备和服务商协同工作在一个大的网络计算环境里。.NET 的最终目的,是要让用户在任何地方、任何时间利用任何设备都能访问到他们所需要的信息、文件和程序。用户不必关心这些东西存在什么地方、如何获取,他们只需发出请求,即可得到结果,所有后台的复杂性是完全屏蔽起来的。

1.2.4 .NET 的主要组成部分

以下是 .NET 的主要组成部分:

1. Windows .NET

Windows .NET 融入了 .NET 技术,整合了 .NET 的一系列构造模块,为数字媒体及应用间协同工作提供了强有力的支持。

2. .NET 框架 (.NET Framework): Web 服务引擎

.NET 框架 (.NET Framework) 提供对开发人员和用户透明化的基本结构。从 .NET 框架角度看,所有组件都可以是 Web 服务,而 Web 服务也仅是一种组件。实际上,.NET 框架提取出微软组件对象模型 (COM) 的精华,将它们与松散联接计算的精华有机地结合在一起,生成了强大、高效的 Web 组件系统:简化了程序员的操作,深入集成了安全性,引进了基于互联网的操作系统,极大地改善了应用程序的可靠性和可扩展性。

3. .NET 企业服务器

.NET 企业服务器的构成异常庞大,它提供并保证了电子商务的高可靠、高性能、可伸缩及易管理等性能。

4. 模块构建服务

模块构建服务是 .NET 平台中的核心网络服务集合。

5. Orchestration 软件集成和自动化处理技术

Orchestration 的目的是尽量不受时间、应用以及个人的限制,最大程度地完成软件集成和自动化处理技术的结合,以使系统更好地服务于商务应用。

6. Visual Studio .NET

Visual Studio .NET 是基于 XML 的编程工具,可以开发基于 .NET 平台的应用软件。它是一套完整的工具,用于生成桌面和基于团队的企业级 Web 应用程序。除了生成高性能的桌面应用程序外,还可以使用 Visual Studio 基于组件的强大开发工具和其他技术,简化基于团队的企业级解决方案的设计、开发和部署。

1.2.5 .NET 的一些其他相关内容

在这里简要介绍一下 .NET 的一些其他相关内容,其中有些内容将在下一章中详细讨论。

(1) .NET 中的普通语言运行: CLR (Common Language Runtime) 普通语言运行时是整

个 .NET framework 构建的基础,是实现 .NET 跨平台、跨语言、代码安全等核心特性的关键。它实际上是驻留在内存里的一段代理代码,负责在整个 .NET 执行期间的代码管理工作,比较典型的有内存管理、线程管理、远程管理和代码强制安全类型等。这些都可称得上 .NET framework 的生命线。实际上,普通语言运行时代理了一部分操作系统的管理功能。

(2) .NET 中的中间语言(IL)及其在 .NET 平台中所起的作用: .NET 中的中间语言是指可以在普通语言规范下,执行时通过即时(Just In Time)编译器转化为本地代码的、与 CPU 独立的一簇指令集合。它在 .NET 平台中是实现语言互操作的一个核心环节,所有 .NET 平台的语言都要先被编译成中间语言(IL)。程序员也可以直接利用中间语言(IL)来编程,但不推荐那样做。中间语言对理解 .NET 对象某些行为和程序的调试很有帮助。

(3) .NET 构架类库: .NET 构架类库是基于运行时面向对象的特性而建造的、和普通语言运行时紧密集成的一簇可重用类(Class)的集合。它大大降低了软件开发的难度,而且很容易与第三方组件无缝集成。

.NET 架构类库主要包括对以下编程模型的支持:

- ① 控制台应用程序。
- ② 脚本和宿主应用程序。
- ③ Windows Forms 应用程序(Windows 桌面 GUI 应用程序)。
- ④ ASP.NET 应用程序。
- ⑤ Web Services 应用程序。
- ⑥ Windows 服务程序。

(4) .NET 中的自动垃圾收集: .NET 垃圾收集器负责整个 .NET 运行时受管代码的内存分配与释放任务。它通过一定的优化算法来选择收集的对象和时间。程序员只有在释放大量受管资源时才可以进行立即强制垃圾收集,在释放非受管资源时采用终止化(Finalize)操作来处理,需要在程序运行中释放的应该实现 Dispose()方法,并明确调用。其他时间将资源的回收交由 .NET 垃圾收集器来做。

(5) .NET 与 COM 的关系: COM 定义了组件对象间通信的二进制标准, .NET 通过组件内置的类型系统提供了类似的机制,它为此提供了另一个途经。 .NET 并不意味着 COM 的终结,但必将大大减少开发人员对 COM 的依赖。 .NET 支持 COM 和 .NET 组件之间的互操作性,实际上可通过 Runtime Callable Wrapper(RCW)来实现在 .NET 中获取 COM 组件,通过 COM Callable Wrapper(CCW)来实现在 COM 中获取 .NET 组件。

1.3 什么东西没有变

尽管 Microsoft .NET 给计算带来了一些翻天覆地的变化,但还有很多东西依然没有改变。

(1) 终端用户将依然使用熟悉的界面,这可以减少再培训的开支,也意味着用户可以马上开始使用 .NET 软件。

(2) 硬件上运行的还是像 Windows、Unix、Windows CE 和 Palm OS 一样的操作系统,实际上, .NET 增加了软件的运行场所,但同时减少了开发的负担。由于 XML Web 服务只使用 XML 与设备通信,所以任何智能设备都可以享用 XML Web 服务。

(3) 对程序员来说,他们依然可以使用原先熟悉的编程语言。 .NET 平台借助于 .NET 框