

.NET构架技术与 Visual C++编程

.NET ARCHITECTURE and PROGRAMMING using VISUAL C++

[美] Peter Thorsteinson, Robert J. Oberg 著
赵学良 译

- 充分利用现有的Visual C++技能来构建先进的.NET应用程序和Web服务
- 涵盖托管代码、COM/Win32、互操作性、Windows® 窗体、ADO.NET、ASP.NET、SOAP、XML、安全性以及更多技术
- 贯穿全书的案例研究，使用Visual C++和.NET构建一个完整的系统



清华大学出版社

高级编程丛书

.NET 构架技术与 Visual C++ 编程

[美] Peter Thorsteinson 著
Robert J. Oberg
赵学良 译

清华大学出版社
北 京

内 容 简 介

.NET 并不是一次彻底的革命,而是多种技术的整合与发展。全书共分为五部分,论述了.NET 是什么、托管 C++ 编程、.NET 的一些最基本主题、用 ASP.NET 和 SOAP 进行 Web 编程以及.NET 中的一些其他主题,如安全性、跟踪与调试和互操作性。本书是 The Integrated .NET Series 丛书之一。本文的作者是从事培训工作多年的资深专家,内容组织简练清晰,毫无累赘,是一部不可多得的教程。

本书的特色为代码示例短小精悍、言简意赅。本书中所有程序界面操作都采用英(中)文注释,使得本书可以适用于中/英文版本 Visual Studio.NET 的使用。

本书适合所有希望了解并掌握.NET,并对 Visual C++ 有所涉猎的读者阅读。

Simplified Chinese edition copyright © 2003 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: .NET Architecture and Programming using Visual C++, 1st Edition
by Peter Thorsteinson, Robert J. Oberg, Copyright © 2002

EISBN: 0-13-065207-5

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc. publishing as Prentice-Hall PTR.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 Pearson Education 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2002-6528

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

.NET 构架技术与 Visual C++ 编程/[美]托尔斯坦森, [美]奥伯格著;赵半良译

—北京:清华大学出版社, 2003

(高级编程丛书)

书名原文: .NET Architecture and Programming using Visual C++

ISBN 7-302-06926-3

I. N… II. ①托…②奥…③赵… III ①计算机网络—程序设计②C 语言—程序设计
IV. ①TP393②TP312

中国版本图书馆 CIP 数据核字(2003)第 060009 号

出 版 者: 清华大学出版社

地 址: 北京清华大学学研大厦

http: www.tup.com.cn

邮 编: 100084

社 总 机: 010 62770175

客户服务: 010 62776969

文稿编辑: 汤涌涛

封面设计: 立日新设计公司

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: 185×260 印张: 31.75 字数: 773 千字

版 次: 2003 年 8 月第 1 版 2003 年 8 月第 1 次印刷

书 号: ISBN 7-302-06926-3/TP·5124

印 数: 1~3000

定 价: 59.00 元

读者对本书的赞誉

This book is proposed for the experienced C++ programmers. The book is very practical, with many examples, and includes a case study that is used to as a realistic demonstration that continues through many of the chapters. this book is intended for experienced C++ developers and provides all the practical insight they need to build Windows and Web Systems for Microsoft's .NET Platform using VC++. With this you can Leverage your Visual C++ skills in advanced .NET development! This book *.NET Architecture and Programming Using Visual C++* gives an innovative dimension of realistic approach to building Windows and Web system with the power of Visual C++.

——G. Gnana Arun Ganesh

这本书是专为有经验的 C++ 程序员准备的，其实践性很强，并且包含大量的示例，以及一个贯穿所有章节的案例研究，作为实际应用的演示。这本书适用于有经验的 C++ 开发人员，提供了使用 VC++ 创建 Microsoft .NET 平台上的 Windows 和 Web 系统所需的全面知识。通过这本书，能够充分地将现有的 Visual C++ 技能应用到先进的 .NET 开发中。本书从一个全新的角度，给出了应用 Visual C++ 构建 Windows 和 Web 系统的切实可行的方法。

——G.Gnana Arun Ganesh, 摘自亚马逊图书网站

This is a great book, perhaps the best on Visual C++ .Net I've ever read. Very clear, covers many interesting and new topics. Since there are not many books on VC++ .Net, and most of the .Net books cover C# and/or VB, this is really a must for any C++ developer who would like to move to C++ .Net.

——Israel Rosenberg

这是一本很好的书，也许是我读过的介绍 Visual C++ .NET 的最好的书。本书条理十分清晰，介绍了许多人们感兴趣的新技术。VC++ .NET 方面的书籍并不多，以前大多数 .NET 相关的图书都是介绍 C# 和/或 VB，这本书对于那些希望转移到 C++ .NET 的开发人员来说真得不可多得。

——Israel Rosenberg, 摘自亚马逊图书网站

译者序

如同各种各样的汽车都依靠发动机一样，计算机世界的基础就是编程语言。越基础的东西变动往往越少，就是因为变动少，所以我们才需要及时跟上，因为这种变化决定了以后几年甚至十几年的编程方式、平台的改变。编程语言的变动往往会引起计算机世界翻天覆地的变化。（在此，编程语言并不仅仅指某一种语言，而是指编程平台及用于该平台开发的语言。）

软件工程与编程语言是互相促进的，为了实现软件工程的要求，编程语言需要提供越来越多的特性；而同时由于编译技术的进步以及新观念的引入，编程语言能够为软件工程提供更好的解决问题的方法。

从微软公司不断发展的技术中就可以看出软件工程的要求在不断发展：MFC，OLE，ATL，COM，COM+，ASP 等等，而能够真正了解所有这些技术的人并不多，更不要说在实际编程中得益于这些技术进步了。软件工程对编程工具与平台提出了各种要求：面向对象、封装、组件、重用……，虽然这些技术都可以部分地解决相应的问题，但是要么太过复杂，难以实现，要么受到种种限制。

现实要求变革。.NET 是微软公司的解决方案。

老问题：.NET 是什么？

说句实话，在读本书以前，让我说.NET 是什么，我还说不上来。微软总是声称.NET 是计算机世界的将来，但.NET 也真的和我们的将来一样，虽然大家都有个模糊的认识，但却说不清。如果您拿到了这本书，祝贺你，您读完本书后就会知道.NET 是什么！

“距离产生美”，用到.NET 上可以说“距离产生神秘”。之所以大家不清楚.NET 是什么，只不过是熟悉而已，举个例子，Internet 是什么？这个问题看似简单，回答起来并不容易，但这并不妨碍大家享受由 Internet 所带来的全新体验。

两种不可买的书

作为编程人员，一般都会对两种书有切肤之痛。

1. 全书中大篇大篇的代码，我以前看到过别人在初学 VC 时买的一本书，里面 1/3 是代码，并且是将 VC 自动生成的项目中的每个文件都打印出来，如果添加按钮、响应鼠标事件的话，则又打印出来一次。难道写书的作者认为读者可能在没有计算机的情况下学习 VC 吗？真是令人费解。（如果说保护视力就更离谱了）

2. 照抄 SDK 的内容。以前这样的书更多些，因为微软中文版的资料出的不及时。“XX 大全”之类的书往往采用这种方式。其实我不否认 SDK 的内容不错，但不适合学习。古人云“明足以查秋毫之末，而不见舆薪”，SDK 就是“秋毫”，只有从大处着眼，才能对事物有整体的把握。

本书中也有代码，但特色就在于此，本书提供的示例短小精悍、言简意赅，用最少的代码来说明问题，真是“增之一分则太肥、减之一分则太瘦”。我在翻译本书的过程中，觉得这些示例十分精彩，它的精彩就在于它的简单，要让示例能够说明一个功能，但却没有一点枝叶，作者的苦心可见一斑。

本书内容

本书涉及了.NET 的许多方面，但是由于.NET 十分庞大、复杂，所以不可能在一本书中详细地论述。在阅读本书的过程中，结合本书提供的示例，可以很快地进入.NET 的编程世界并熟悉它，但要真正掌握.NET 编程则需要更多的实践，甚至实际的项目经验才行。在这一点上，书籍是做不到的。

版本说明

在对照本书学习时，最好将随书的示例重新编译一下，以避免版本问题。由于 Visual C++ 都是英文版的，所以本书中也用了英文版的 Visual Studio .NET。但本书中的所有程序界面操作都采用英文后面标注中文的做法。这样使得本书可以适用于中/英文版本 Visual Studio .NET 的使用。我的一点经验是：对于熟悉 Visual C++ 以前版本的开发人员，最好先学习英文版，原因有三：

1. 在 Visual Studio .NET 中，VB、VC、C# 都采用一个集成环境来开发，所以相对原来的 Visual C++ 集成开发环境来说还是有很大变化，使用英文版可以很快地将原来使用 Visual C++ 6.0 的经验移植到新的集成环境中。之后，再使用中文版也比较容易。而如果一开始就使用中文版的话，一些东西都得从头熟悉。而且，回到英文版可能对应不起来。

2. 中文版中一些出错信息也是中文的。使用过 Visual C++ 6.0 的用户要适应中文的编译错误，我想还是需要很长的一段时间。

3. 英文版更新会快一些。虽然现在处理软件多种语言版本的技术已经十分成熟，但是由于.NET 与 Visual Studio .NET 是全新的平台和工具，所以其更新会相对快些，在快速更新的过程中，英文版肯定还是有一些优势的。

致 谢

尤晓东先生对本书翻译作了大量指导性的工作，并给出了大量规范，使文字更为标准。感谢汤涌涛女士对译稿提出的许多意见，我们之间的交流使本书更臻完善。

赵学良
上海虹桥
2003 年 4 月

序

近几年来, Microsoft Visual C++已经成为了 Windows 软件开发中最强大的工具。虽然获得 Visual C++开发技能需要大量的投入,但是 Visual C++有一些功能是其他的编程语言所不能及的。现在.NET 世界向我们敞开,这真是令人兴奋:我们可以继续使用 Microsoft Visual C++,而且还可应用崭新的托管 C++扩展来进行开发,以期在开发中实现最大可能的强大功能与最高的性能。托管 C++可以用来开发.NET 程序集和创建令人惊奇的桌面程序、Web 应用程序和 Web 服务。非托管 C++可以用来构建基于 ATL Server 的 Web 站点和 Web 服务。

.NET 代表 Microsoft Windows 软件开发的一次飞跃。.NET 对 Microsoft Windows 程序员来说同样代表着巨大的知识挑战。新的平台包括新的 C++语言扩展,以及大量的类库——.NET 框架。

本书实践性很强,伴有大量示例以及一个案例研究,该案例研究作为一个现实应用的演示贯穿了许多章节。本书的目的是帮助读者使用 Visual C++和.NET 框架构建卓有成效的应用程序。本书属于 Object Innovations and Prentice Hall PTR 出版社 The Integrated .NET Series 丛书。

组 织

本书由 5 个主要部分构成,本书的结构使读者能够轻松地定位到需要学习的章节。第 I 部分由第 1 章和第 2 章组成,它提供一个概述,每位读者都应该阅读这一部分。它回答了一个重要问题:“Microsoft .NET 是什么?”,并简要介绍了.NET 框架的编程模型。

第 II 部分由第 3~5 章组成,这一部分介绍托管 C++的编程。即使读者熟悉传统 C++,也可能需要阅读一下这些章节。第 4 章介绍 C++的托管扩展。本书的案例研究(在整本书中详细阐述)也在第 4 章引入。第 5 章介绍重要的主题:接口、委托和事件,同时本章还描述了托管 C++与.NET 框架之间的重要交互。

第 III 部分是第 6~9 章,这一部分介绍.NET 框架中最基本的一些主题。第 6 章介绍使用 Windows 窗体类进行用户界面编程。第 7 章论述程序集和部署,它们在部署 Windows 应用程序方面具有简单和健壮的突出优点,终结了臭名昭著的“DLL 地狱(DLL hell)”噩梦。第 8 章介绍重要的.NET 框架类,包括元数据、序列化、线程、属性、异步编程、远程处理和内存管理等主题。第 9 章介绍 ADO.NET,ADO.NET 为访问关系型数据和 XML 数据提供了一系列一致的类。

本书第 IV 部分深入介绍了使用 ASP.NET 和 SOAP 进行 Web 编程。第 10 章介绍用 ASP.NET 来开发 Web 站点的基础知识,包括 Web 窗体的使用。第 11 章介绍 SOAP 和

Web 服务，它们为异构系统的互操作提供了一种易用和健壮的机制。第 12 章展示了怎样使用 ATL Server 模板库来创建 Web 服务器与 Web 服务。

最后一部分介绍 .NET 框架中其他的重要主题。第 13 章详细介绍安全性主题，包括代码访问安全性以及声明安全性。第 14 章介绍 .NET 提供的调试与跟踪类。第 15 章介绍 .NET 与遗留 COM 组件以及 Win32 应用程序之间的互操作性。

示例程序

切实掌握一个重要框架的惟一方法就是阅读与编写大量的程序，程序大小要适度。本书提供大量的小程序，独立说明 .NET 中相关的特性，便于理解。

从第 4~12 章，逐渐地开发一个主要的案例研究：Acme 旅行代理。它说明了托管 C++ 与 .NET 组合用于实际应用程序的许多特性。

示例程序在本书的 Web 站点上以自解包文件的形式提供。解开后，会创建一个目录结构，其默认的根目录是 C:\OI\NetCpp。示例程序（从第 2 章开始），分别在 Chap02、Chap03……中。给定章的每个示例都在该章目录下的单独文件夹中。文件夹的名称清楚地以文本标识出来。包含案例研究步骤的每一章中都有一个 CaseStudy 文件夹。如果需要，在每章的目录下还会有 readme.txt 文件，解释使示例工作所必需的用法说明。

本书属于 The Integrated .NET Series 丛书。本系列中其他书籍的示例程序在 \OI 目录中相应的目录下，本系列中所有书的所有 .NET 示例都可以在一个公共的区域（读者的安装目录）找到。

单独提供这些程序仅仅是为了指导学习，不应该将其嵌入到任何软件产品中。本书提供的软件（包括使用说明）不包括任何形式的任何担保。

警 告

本书与相关的代码是用 .NET 框架的 Beta 2 版本开发的。微软已经提示该版本的 .NET 接近于最终版本。虽然如此，在 .NET 发布前可能还会有修改。示例中的代码已经在 Windows 2000 中验证过，可以工作。数据库代码在 SQL Server 2000 上验证过。在数据库与安全性章节的几个示例中，在连接字符串或角色名中嵌入了计算机名。当尝试运行这些示例时，必须将这些名称替换成相应计算机的名称。为了使安装更为容易，数据库示例使用用户名“sa”并且没有密码。勿须说明，在真实的应用中，我们从不应该使任何登录 ID 没有密码，或让数据库应用程序使用 sa 来登录到数据库。

网 站

本系列图书的 Web 站点是：

www.objectinnovations.com/dotnet.htm

在该 Web 站点上，提供了一行链接用于下载本书的示例程序。关于 .NET 的其他信息可以在下面的 URL 得到：

www.mantasoft.com/dotnet.htm

本书的示例程序同样也可以在这个 Web 站点得到。

本书的 Web 站点还包括 .NET 学习资源的一个列表，并且会及时更新。

致 谢

感谢 Mike Meehan，当微软宣布 .NET 时，从在 PDC 的一个会议开始，他使得本项目得以实施。这次会谈促成了一系列关于 .NET 技术的图书，其中本书是第三部。还要感谢 Prentice Hall 的 Jill Harry，感谢她对本图书项目的支持。我们的编辑，Nick Radhuber，也为本书以及整个丛书提供了大量的帮助。

微软的几位人士在许多方面给予了帮助：Steven Pratschner, Jim Hogg, Michael Pizzo, Michael Day, Krzysztof Cwalina, Keith Ballinger 和 Eric Olsen。感谢他们在百忙之中抽出时间为我们提供了深入的见解和说明。Connie Sullivan 和 Stacey Giard 整理了技术会话，并帮助我们访问微软的资源。

Michael Stiefel 是本丛中另一本书的作者，他为本书的许多章节提供了资源。Will Provost 帮助澄清了关于 XML 的几个问题。我们还要感谢 .NET 丛书的所有其他作者，因为在同时编写图书的团队工作中，我们进行了许多协同工作，尽管在写作最激烈的时候，我们的协同并不总是如预期那样的紧密。感谢以下人员的努力工作：Eric Bell, Howard Feng, Michael Saltzman, Ed Soong, Dana Wyatt, David Zhang 和 Sam Zhu。

很多人对本书的编写提供了帮助，因此 Robert 很难一一对这些人表示感谢。我（Robert）要感谢我的妻子 Marianne，因为她在我写作的过程中提供了大量的支持和鼓励。由于这个项目的要求十分严格，所以我十分感谢她的支持。感谢所有的人：多年来帮助我的其他同事、朋友和学生——由于太多了就不在此单独列举出来。

Peter 希望感谢他最爱的妻子 Elizabeth 和女儿 Katherine，他永远爱着她们。

2001 年 11 月 7 日

目 录

第 1 章 Microsoft .NET 是什么	1
1.1 微软与万维网	1
1.2 Windows 桌面系统	3
1.3 新的编程平台	4
1.4 XML 的角色	7
1.5 小结	7
第 2 章 .NET 基本原理	8
2.1 Windows 开发缺陷	8
2.2 未来的应用程序	8
2.3 .NET 概述	9
2.4 小结	16
第 3 章 托管 C++ 编程	17
3.1 C++ 与 .NET 世界的契合	17
3.2 启用 C++ 托管扩展	18
3.3 初涉托管 C++ 程序	18
3.4 映射 C++ 到 CLS 和 .NET 框架	35
3.5 .NET 平台的 C++ 编程	37
3.6 异常处理	54
3.7 C++ 的属性	56
3.8 小结	61
第 4 章 托管 C++ 中面向对象的编程	62
4.1 面向对象概念的回顾	62
4.2 Acme 案例研究: 设计	65
4.3 托管 C++ 中的继承	72
4.4 Acme 案例研究: 实现	73
4.5 小结	79
第 5 章 .NET 框架下的托管 C++	80
5.1 System::Object	80
5.2 集合	85
5.3 接口	88
5.4 .NET 中的通用接口	101
5.5 委托	115
5.6 事件	123
5.7 小结	129

第 6 章 用户界面编程	130
6.1 Windows 窗体的体系	130
6.2 用 .NET SDK 编写的简单窗体	130
6.3 Windows 窗体的事件处理	136
6.4 菜单	140
6.5 控件	142
6.6 Visual Studio .NET 和窗体	144
6.7 对话框	156
6.8 列表框控件	162
6.9 Acme 案例研究步骤 3	164
6.10 小结	166
第 7 章 程序集与部署	167
7.1 程序集	167
7.2 私有程序集的部署	174
7.3 共享程序集的部署	175
7.4 程序集配置	183
7.5 多模块程序集	186
7.6 部署案例研究	189
7.7 安装和部署项目	192
7.8 小结	196
第 8 章 .NET 框架类	197
8.1 元数据和反射	197
8.2 .NET 中的输入输出	203
8.3 序列化	207
8.4 .NET 应用程序模型	213
8.5 上下文	226
8.6 应用程序隔离	228
8.7 异步编程	232
8.8 远程处理	238
8.9 自定义属性	243
8.10 垃圾回收	247
8.11 小结	255
第 9 章 ADO.NET 编程	256
9.1 数据提供程序	257
9.2 Visual Studio .NET 服务器资源管理器	258
9.3 连接数据库	259
9.4 数据读取器	259

9.5 参数集合	265
9.6 SqlDataAdapter 和 DataSet 类	267
9.7 DataSet 中的集合	268
9.8 数据集基本原理	270
9.9 数据库事务与更新	274
9.10 开放式与保守式锁定和 DataSet	275
9.11 DataSet 的使用	277
9.12 Acme 案例研究	292
9.13 XML 数据访问	292
9.14 AirlineBrokers 数据库	293
9.15 小结	299
第 10 章 ASP.NET 和 Web 窗体	300
10.1 ASP.NET 是什么?	300
10.2 Web 窗体构架	305
10.3 请求/响应编程	315
10.4 案例研究	322
10.5 ASP.NET 应用程序	332
10.6 ASP.NET 应用程序中的状态	334
10.7 ASP.NET 配置	336
10.8 ASP.NET 进阶	337
10.9 小结	337
第 11 章 Web 服务	338
11.1 协议	338
11.2 Web 服务的构架	341
11.3 SOAP 格式化的差别	352
11.4 WebService 类	356
11.5 Managed C++ Web Service 模板	356
11.6 Arithmetic Web 服务	360
11.7 Hotel Broker Web 服务	364
11.8 小结	368
第 12 章 ATL Server 站点与 Web 服务	369
12.1 动态 Web 内容技术的历史	369
12.2 ATL Server 应用程序	370
12.3 ATL Server 应用程序构架	371
12.4 创建 ATL Server 项目	372
12.5 创建 ATL Server Web 服务项目	395
12.6 小结	404

第 13 章 安全性	405
13.1 基于用户的安全性	405
13.2 代码访问安全性	406
13.3 Internet 安全性	407
13.4 .NET 中基于角色的安全性	408
13.5 代码访问权限	416
13.6 代码标识	427
13.7 安全策略	430
13.8 小结	440
第 14 章 .NET 跟踪与调试	441
14.1 TraceDemo 示例	441
14.2 部署 TraceDemo.exe.config	442
14.3 使用 Debug 和 Trace 类	442
14.4 使用开关来启动诊断	443
14.5 开启和禁止开关	444
14.6 TraceListener	445
14.7 侦听器集合	445
14.8 小结	446
第 15 章 托管与非托管代码的混合编程	447
15.1 对比托管与非托管代码	447
15.2 托管与非托管引用和值类型	449
15.3 C++ 中使用托管类型的限制	451
15.4 跨越托管与非托管边界的调用	453
15.5 对比 COM 和 .NET 中的 C++ 编程	455
15.6 从托管代码中访问 COM 组件	457
15.7 从 COM 客户程序中访问托管组件	465
15.8 平台调用服务	473
15.9 小结	475
附录 A Visual Studio .NET	476
A.1 Visual Studio .NET 概述	476
A.2 创建控制台应用程序	479
A.3 项目配置	483
A.4 调试	485
A.5 小结	490
术语表	491

第 1 章 Microsoft .NET 是什么

.NET 是微软对 Internet 时代应用程序的构想。基于开放的 Internet 标准，.NET 提供了增强的互操作性；同时，.NET 提高了传统 Windows 桌面应用的活力与健壮性；为开发者提供了一个全新的编程平台以及优秀的工具集，其中 XML 扮演一个基本的角色。

Microsoft .NET 是在操作系统之上构建的一个平台。.NET 是微软一项至关重要的投资，制作期长达三年。.NET 受到许多其他技术进步的影响，比如说 XML, Java™ 和 COM。

.NET 提供了：

- 公共语言运行库（Common Language Runtime, CLR），一个健壮的运行平台。
- 多语言开发。
- .NET 框架，一个可扩展的编程模型，为所有的 .NET 语言提供一个庞大的可重用代码类库。
- 支持在 Internet 标准上构建的网络应用基础结构，允许应用程序之间进行高层通信。
- 支持新工业标准——Web 服务（Web Services）。Web 服务提出一个应用程序交付的新机制，这套机制将基于组件的开发理念扩展到了 Internet。
- 一个安全模型，程序员可以很容易地将其应用到程序中。
- 强大的开发工具。

1.1 微软与万维网

万维网（World Wide Web, WWW 或 Web）已经成为微软最大的、需要迎头赶上的挑战。事实上，Web 与微软的传统势力范围——个人电脑，共存了相当长的时间。通过一个 PC 应用程序，即浏览器，用户就可以访问到整个信息世界。Web 依赖于 HTML、HTTP 和 XML 等标准，对于在使用各种计算机系统和设备的用户之间进行通信来说，这些标准是基本的。

虽然有其复杂性，但是 Windows PC 是十分标准化的。虽然 Web 是基于标准协议的，但是在这些协议之上运行的语言、数据库、开发环境以及设备却是一片混乱。随着技术复杂性呈爆炸性的增长，能够应用新技术构建所需系统的知识人才越来越短缺。有了 .NET 提供的基础结构，程序员就可以不必重复解决一般的编程问题，而将精力集中在为应用程序增值上。

1.1.1 Internet 时代的应用程序

最初，Web 是一个巨量的信息库。浏览器发出对信息页的请求，Web 服务器将请求的信息以静态 HTML 页面的形式传送给浏览器。即使在交互式 Web 应用程序引入后，

HTML, 这种将信息以及信息的查看格式组合在一起的语言, 仍然被使用。

XML 提供了一种标准的、与格式无关的数据传送方式: 这样, 就可能在公司间就文档与信息流达成标准, 比如: 订购单和发票等; 电子商务也得以在合作公司之间自动进行 (B-to-B)。XML 只描述数据, 而不支持对数据的操作, 需要对数据进行操作的话, 我们需要 Web 服务。

1.1.2 Web 服务

.NET 最重要的方面之一就是 Web 服务的支持。基于简单对象访问协议 (Simple Object Access Protocol, SOAP) 这一工业标准, Web 服务允许通过 Internet 来呈现应用程序的功能。对于 .NET 程序员来说, Web 服务与 .NET 语言中类实现的服务并没有什么不同。不管是在同一个应用程序内, 还是在独立的组件之间, 或者运行于不同机器上的 Web 服务之间, 调用一个函数的编程模型都是相同的。

与生俱来的简单性使得公司创建与运行应用程序变得十分容易, 如果需要的话, 整个应用程序可以完全外包, 不再需要开发、部署和维护; 我们也可以使用在设计时尚未存在的第三方 Web 服务。

1.1.3 ASP.NET

.NET 包含一个关于 ASP (Active Server Page, ASP) 技术的完全重写版本, 也就是 ASP.NET。虽然 ASP 依赖于解释性描述代码 (用有限功能的语言和一些页面格式命令书写而成), 但是 ASP.NET 代码可以用任何 .NET 语言来写, 包括 C#、VB.NET、JScript 和包含托管扩展的 C++。由于它是编译后的代码, 所以可以通过一个独立的“代码隐藏 (code behind)”文件将界面代码与商业逻辑分开。虽然 C#、VB.NET 和 JScript 可以作为脚本嵌入到 Web 页面中, 但托管 C++ 写的代码一定要放在代码隐藏文件中。

ASP.NET 提供 Web 窗体, 极大地简化了 Web 用户界面的创建。

Visual Studio .NET 中的拖放使窗体的设计变得极为容易, 同时还可以为窗体事件添加代码, 比如按钮选择事件。

ASP.NET 会自动检测浏览器的能力, 对于高端浏览器, 代码处理可以在客户端执行; 对于低端的浏览器, 服务器会完成处理并生成标准的 HTML 页面, 就开发者来说都是透明的, ASP.NET 自动完成了这些处理。

Web 服务与编译语言的结合, 如: C#、VB.NET 以及托管 C++, 使 Web 编程可以依照面向对象的模型来进行, 而对 ASP 描述性语言与 COM 组件来说, 这是不可能的。

1.1.4 开放式标准与互操作性

现代计算环境包含大量多样的硬件及软件系统; 计算机应用范围涵盖了从大型机、高端服务器, 到工作站、PC, 甚至小型的移动设备, 比如 PDA 或者移动电话; 操作系统包括传统的大型机操作系统、UNIX 的各种变体、Linux、各版本的 Windows、实时操作系统和特殊系统, 比如用于移动设备的 PalmOs; 另外, 众多的编程语言、数据库、应

用开发工具和中间件产品也都在应用。

在现代的计算环境中, 只有很少的应用程序是孤立的; 甚至在 PC 上运行的零售应用程序也可能通过 Internet 来注册或升级。不同应用程序之间互操作性的关键就是使用标准, 由于应用程序一般都在网络上运行, 所以关键的标准就是使用的通信协议。

1.1.5 通信协议

TCP/IP 套接字是高度标准化的, 并且被广泛支持, 但对于那些需要编写健壮的、分布式应用程序的程序员来说还是过于低层了; 可能更高层一点的是远程过程调用 (Remote Procedure Call, RPC), 但是 RPC 还是过于复杂, 并且 RPC 有许多变体; CORBA、RMI 和 DCOM 等是更高层的、应用广泛的协议, 它们依旧复杂, 并且需要特定的环境支持, 同时还有许多其他的缺点, 比如难以通过防火墙。

由于 HTTP 已经成为一个无所不在的协议, 因此微软与其他的提供商引入了一个叫作简单对象访问协议 (Simple Object Access Protocol, SOAP) 的新协议。SOAP 应用 XML 这种基于文本的语言, 将对对象方法的请求与相关的数据编码。SOAP 最大的优点就是其简单性, 从而易于在各种各样的设备上实现。SOAP 可以在任何协议上运行, 由于它可以在标准的 Internet 协议之上运行, 诸如: HTTP 和 SMTP, 所以通过防火墙时不会有任何连接方面的问题。

1.2 Windows 桌面系统

微软是从桌面系统开始的, 现代的 Windows 环境已经变得无处不在, 拥有无数的应用程序, 大多数用户至少会在家里使用 Windows。尽管微软已经尽量完善 Windows 系统, 但是依旧存在一些明显的问题。

1.2.1 Windows 的问题

维护一个 Windows PC 是一项繁琐的工作, 因为应用程序十分复杂: 它们往往是由许多文件、注册表项、快捷方式等组成; 不同的应用程序可能共享特定的动态链接库 (Dynamic Link Library, DLL), 安装一个新的应用程序可能会覆盖现有程序使用的一个 DLL, 可能会使旧的应用程序不能运行 (“DLL hell, DLL 地狱”), 卸载一个应用程序也同样复杂, 而且很难完全卸载。

PC 会渐渐地变得不稳定, 有时会需要重新格式化硬盘, 然后从零开始才能解决。由于标准的应用程序十分廉价, 并且功能强大, 硬件也十分便宜, 因此使用 PC 有巨大的经济利益, 但是维护费用在一定程度上抵消了所得。

在开发 Windows 时, 当时的个人电脑并没有进行网络互联, 也不需要考虑安全性。当 Windows NT 和 Windows 2000 内建了安全性后, 原来的编程模型就难以为继了 (提问: 你曾经用其他的参数, 而不是 NULL, 作为 Win32 的 LPSECURITY_ATTRIBUTES 参数吗?)。

1.2.2 “玻璃房”与瘦客户

旧的“玻璃房 (Glass House)”模型是这样的：一台中心计算机控制所有的应用程序，这种模型曾经有过需求，也曾存在某种形式的向“瘦客户”转移的意愿，但是预期的网络 PC 从未真正流传开来。

基于 PC 的应用程序对用户来说有太大的价值，同时用户喜欢拥有“自己的”PC 的感觉，他们的数据可以方便地存储在本地计算机中。在没有高带宽连接的情况下，一个基于服务器的字处理程序表现不会很好。安全性对于瘦客户来说也是一个十分难以解决的问题。个人电脑无疑将会继续存在。

1.2.3 健壮的 Windows

在所有关于 .NET 与 Internet 的宣传中，重要的是意识到 .NET 改变了编程模型，允许创建更加健壮的 Windows 应用程序。应用程序不再需要将大量的配置数据存储到脆弱的 Windows 注册表中。.NET 应用程序是自描述的，其程序执行文件自身内部包含了元数据。组件的不同版本可以并行地部署。应用程序可以通过全局程序集缓存 (Global Assembly Cache, GAC) 来共享程序集 (Assembly)。版本控制集成在部署模型中。.NET 还包含一个简明的安全性模型。

1.3 新的编程平台

现在，我们从新编程平台的角度，看看已经讨论了关于 .NET 的哪些方面：

- 可以完成类型安全以及安全性检查，提供更加健壮的操作。
- 比 Win32 API 或 COM 更易于编程。
- 部分或全部平台可以在多种计算机上实现（就像 Java 所做的那样）。
- 所有的 .NET 语言共享同一个类库。
- .NET 语言可以容易地互相集成。

.NET 平台有下面几个重要部分：

- .NET 框架
- 公共语言运行库
- 多语言开发
- 开发工具

1.3.1 .NET 框架

现代编程方法十分依赖于库提供的可重用代码。面向对象的语言便利了类库的创建，类库具有很好的柔韧性、抽象性和可扩展性，后者表现在通过基于已有的类增加新的类，可以继承现有的函数。

.NET 框架提供了超过 2 500 个可以重用的类，可以被所有的 .NET 语言调用。.NET 框架是可扩展的，新的类可以从现存的类继承而来，即使该类是在另外一种语言中实现