



软件工程师丛书



免费附赠实例光盘 1 张

# Visual C++ .NET

## 高级编程技术与范例

胡海璐 周涛武 等编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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



软件工程师丛书

# Visual C++ .NET 高级编程技术与范例

胡海璐 周涛武 等编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 提 要

本书全面详细地介绍了 Visual C++ .NET 的所有新特性，书中通过实例循序渐进地讲解了 Visual C++ .NET 的高级编程技术与技巧。通过阅读本书并实际练习，读者能够迅速提高自己的开发水平，从而开发出功能强大的 Web 应用程序和数据访问应用程序。

书中的每章都深入浅出地剖析了 Visual C++ .NET 中最新、最重要且最实用的技术，提供的数十个开发实例涵盖了使用 Visual C++ .NET 开发当前最为热门的应用，包括高级界面、GDI+、OpenGL、多媒体、网络、数据库、ATL、Web Service、XML 及 STL 等多个方面。这些实例都能够为读者的实际编程起到很好的借鉴和指导作用，从而使读者牢牢地把握技术发展趋势和 Visual C++ .NET 开发应用系统的关键所在，获得真正意义上的提高。

本书所附的光盘中包括所有实例的源代码，读者可直接或稍加修改后使用。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有，侵权必究。

### 图书在版编目 (CIP) 数据

Visual C++.NET 高级编程技术与范例 / 胡海璐等编著. —北京：电子工业出版社，2002.8

(软件工程师丛书)

ISBN 7-5053-7925-9

I. V... II. 胡... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 060935 号

责任编辑：寇国华

印 刷：北京市天竺颖华印刷厂

出版发行：电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销：各地新华书店

开 本：787×1092 1/16 印张：25.75 字数：605 千字 附光盘 1 张

版 次：2002 年 8 月第 1 版 2002 年 8 月第 1 次印刷

印 数：6000 册 定价：48.00 元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：(010) 68279077

# 出版说明

随着我国加入 WTO, 现代化建设也将以前所未有的步伐向前迈进。我们面临更大的挑战, 也面临更多的机遇。一个不争的事实是计算机的应用普及将更加深入, 将需要数量更多、水平更高的软件工程师。

我国的软件工程师队伍已有了长足的发展, 软件开发水平已有了长足的进步。作为中国人, 我们期盼的是中国软件业走自主创新之路, 在世界上的地位越来越高。作为出版工作者, 为发展我国的软件事业尽最大努力, 是我们义不容辞的责任, 这正是我们于 1999 年底推出《软件工程师》丛书的初衷。

目前这套丛书已出版了 40 多种。从市场销售和读者反馈的情况看, 这套丛书已经得到了读者的首肯和厚爱, 这也是对我们下一步工作的激励。

可以说, 计算机应用系统的多样化、规模化和复杂化对软件工程师提出了更高的要求, 同时也为软件工程师提供了更多的施展个人才华的机会。

针对这种形势, 我们正在扩充《软件工程师》丛书的选题范围, 进一步界定这套丛书的特色, 并把丛书按如下类型整合。

一是开发类, 通过大量实例说明如何使用各种流行的高级语言、工具类软件开发不同的应用系统, 说明开发思想、开发过程、难点及其解决方案。为了适应我国软件工程师开发综合软件系统的需求, 我们把包含编程功能在内的高级应用软件的开发应用也纳入到书中。

二是技巧类, 通过大量实例说明在不同应用系统开发过程中, 有关缩短开发周期、提高开发质量、解决开发中的疑难问题的各种技巧。

三是技术类, 介绍软件开发的有关理论和技术, 以及在实际中的应用, 如系统分析与系统设计、软件测试和系统安全等。

四是手册类, 即每个软件工程师必备的案头书。

在新的一年里开始之际, 这套丛书从内容、开本、印刷及装帧等方面都将以全新的面貌与广大读者见面, 目的在于使其更受读者的欢迎, 每本书能容纳更多的信息。

我们以为软件工程师提供图书信息服务为宗旨, 坚持以图书质量为生命。我们希望《软件工程师》丛书能对读者有所帮助, 希望读者提出更多的宝贵建议和意见, 包括工作中遇到的技术难点、疑点和问题。希望更多的专家加入我们的作者行列, 推介自己的实践经验和累累硕果。我们的网址是 [www.phei.com.cn](http://www.phei.com.cn), 请和我们联系。

为了我国软件业的更加美好的明天, 让我们共同努力。

电子工业出版社

# 前 言

Visual Studio.NET 在数百万软件开发人员使用的基础上，全面加强了开发应用系统的功能。Microsoft 又一次用最简捷的方式将其 .NET 网络计算理念通过 Visual Studio.NET 影响全世界的软件开发人员。现在，开发人员可以用最简单的方式写出最好的程序。

作为 Visual Studio.NET 家族中的一个成员，Visual C++.NET 并不是简单地在 Visual C++ 6.0 的基础上增加一些新特性。而是根据 Internet 的需要重新设计，使其能够满足更加方便快捷地开发 Web 和企业级多层次系统之类的分布式应用程序的需求。

Visual C++.NET 是目前最新和最先进的软件开发工具之一，它汇集了 Microsoft 公司的技术精华。不仅全面体现了面向对象技术，而且在编译优化技术方面较其他同类产品具有明显的优势。它也是一个彻底的程序员级的开发环境，可视化的设计大大减少了编程的工作量。利用 Visual C++.NET，几乎可以完成任何设计工作：小至普通的应用系统，大至复杂的应用开发工具。

对于有一定 Visual C++ 编程经验的开发人员，那些详细介绍 MFC 编程机制和类库的书籍已无法满足其实际应用中的需要。本书力图以新的视点阐述目前最为热门的 Visual C++.NET 编程的高级技术和技巧，从而为读者提供卓有成效的帮助。

本书内容全面深入，详细介绍了 Visual C++.NET 的几乎所有新特性。本书并不只是知识点的简单罗列，而是重点通过实例循序渐进地讲解使用 Visual C++.NET 的关键技术和编程技巧。使得读者能够理解并灵活运用这些知识点，迅速掌握这个开发语言，从而能够开发出功能强大的 Web 应用程序和数据访问应用程序。

本书提供的几十个实例涵盖了高级界面、系统应用、GDI+、OpenGL、多媒体、网络、数据库、ATL、Web Service XML 及 STL 等多个方面。通过这些实例，详细介绍了使用 Visual C++.NET 开发应用系统的高级技术与技巧。书中深入浅出地剖析了 Visual C++.NET 中最新、最重要且最实用的技术，这些实例都能够为读者的实际编程起到很好的借鉴和指导使用，从而使读者牢牢地把握技术发展趋势和关键所在，获得真正意义上的提高。

本书所附光盘中包括所涉及实例的所有源代码，并且所有实例均在 Visual C++.NET 环境下调试实现。

本书由胡海璐和周涛武合作编写，其中第 1 章和第 6 章~第 11 章为胡海璐编写，第 2 章和第 3 章~第 5 章由周涛武编写，

参加本书编写工作的还有彭接文、左震宇、付领军、胡水龙、吴玉云、郭威、麻尚春、单红、赵静、张春华、刘燕、曾崎、牟建军、王小锋、蔡立春和许波。

本书在写作中融合了作者多年的程序设计经验和对 Visual C++.NET 的理解，力求做到言之有物，更好地体现实用性和操作性。当然，由于水平和经验有限，书中不足和纰漏在所难免，恳请广大读者批评指正。

我们的电子邮箱地址是： 胡海璐： [perilla@163.com](mailto:perilla@163.com)

周涛武： [taurcn@msn.com](mailto:taurcn@msn.com)

欢迎读者与我们联系，共同探讨有关 Visual C++.NET 高级编程方面的有关问题。

作 者

2002年3月

# 目 录

<b>第 1 章 Visual C++ .NET 简介</b> .....	<b>1</b>
1.1 .NET 平台概述 .....	2
1.1.1 .NET 平台产生的背景 .....	2
1.1.2 .NET 平台的组成 .....	3
1.1.3 .NET 技术框架内核 .....	4
1.1.4 .NET 主要特性 .....	5
1.2 Visual C++ .NET 运行环境 .....	6
1.3 VS .NET 的 IDE 界面 .....	6
1.3.1 集成开发环境 .....	6
1.3.2 菜单与工具栏 .....	7
1.3.3 属性窗口 .....	7
1.3.4 代码编辑器 .....	10
1.3.5 动态帮助窗口 .....	11
1.3.6 文件区 .....	12
1.3.7 工具箱 .....	12
1.4 Visual C++ .NET 新特性 .....	13
1.4.1 项目建立 .....	13
1.4.2 多顶级文档 .....	14
1.4.3 标准兼容 .....	15
1.4.4 托管的 C++ .....	15
1.4.5 集成 MFC 和 ATL .....	18
1.4.6 新的 UI 特性和更新 .....	18
1.4.7 管理位图的 CImage .....	19
1.4.8 CLinkCtrl 公共控制 WRapper 类 .....	20
1.4.9 HTTP 客户 .....	20
1.4.10 SMTP 和 MIMI-encoded 消息支持 .....	20
1.4.11 带有 OLE DB 的高性能数据库 .....	20
1.4.12 STL 的更新 .....	21
<b>第 2 章 高级界面编程</b> .....	<b>23</b>
2.1 概述 .....	24
2.2 菜单 .....	24

2.2.1	菜单类 CMenu.....	24
2.2.2	菜单类的成员.....	24
2.2.3	菜单类的应用.....	26
2.2.4	实例：在应用程序中使用菜单类实现“我的收藏夹” .....	27
2.3	控件.....	31
2.3.1	概述.....	31
2.3.2	Windows 公共控件的信息.....	31
2.3.3	创建和使用控件.....	32
2.4	按钮.....	35
2.5	窗口.....	36
2.5.1	概述.....	36
2.5.2	创建.....	37
2.5.3	销毁.....	37
2.5.4	更改 MFC 创建的框架窗口的样式.....	38
2.5.5	Windows 消息函数.....	38
2.5.6	CWnd 类的派生应用 .....	39
2.5.7	具体示例 .....	39
2.6	对话框.....	42
2.6.1	概述.....	42
2.6.2	创建并显示对话框.....	42
2.6.3	初始化对话框.....	44
2.6.4	在对话框中处理 Windows 消息.....	44
2.6.5	在对话框对象中检索数据.....	44
2.6.6	关闭对话框.....	44
2.6.7	销毁对话框.....	44
2.6.8	用代码向导创建对话框类.....	45
2.6.9	应用实例 .....	45
2.7	基于 HTML 的对话框.....	49
2.7.1	Web 对话框的创建方法.....	49
2.7.2	使用对话框资源与 HTML 资源编辑器 .....	51
2.7.3	HTML 对话框编程.....	53
2.8	小结.....	55
<b>第 3 章</b>	<b>系统编程.....</b>	<b>57</b>
3.1	概述.....	58
3.2	创建与管理进程.....	58
3.2.1	进程概述.....	58
3.2.2	创建进程.....	58
3.2.3	进程的销毁.....	61



3.2.4	进程相关函数.....	62
3.2.5	进程实例 .....	64
3.3	线程以及同步 .....	68
3.3.1	线程概述 .....	68
3.3.2	创建线程 .....	69
3.3.3	结束线程 .....	72
3.3.4	线程同步 .....	72
3.4	文件 .....	75
3.4.1	文件操作 .....	75
3.4.2	目录操作 .....	77
3.4.3	文件实例 .....	78
3.5	注册表与系统信息 .....	81
3.5.1	注册表基础.....	81
3.5.2	Win32 中的注册表操作函数 .....	82
3.5.3	.NET 框架类库中对注册表的支持 .....	83
3.5.4	操作注册表的编程步骤 .....	84
3.5.5	实例 .....	85
3.5.6	获取系统的网络信息.....	86
<b>第 4 章</b>	<b>GDI+图像编程 .....</b>	<b>91</b>
4.1	概述 .....	92
4.1.1	GDI+的组成部分 .....	92
4.1.2	基于类的接口结构.....	93
4.1.3	新增加的功能.....	93
4.1.4	GDI+编程模块的变化 .....	96
4.2	应用 GDI+编程.....	99
4.2.1	GDI+编程步骤 .....	99
4.3	GDI+绘图基础.....	100
4.3.1	坐标空间与坐标变换.....	100
4.3.2	画笔 .....	102
4.3.3	画笔和渐变画笔.....	103
4.3.4	重新着色 .....	105
4.3.5	图形容器 .....	107
4.3.6	区域 .....	108
4.3.7	基本绘图函数.....	110
4.4	选择字体和绘制文本 .....	111
4.4.1	构造字样和字体.....	111
4.4.2	文本输出 .....	112
4.4.3	设置文本格式.....	113

4.4.4	文本质量输出控制.....	115
4.5	GDI+的图像处理.....	116
4.5.1	概述.....	116
4.5.2	图像文件格式.....	116
4.5.3	调用和显示图像文件.....	117
4.5.4	调用和显示矢量图像文件.....	120
4.6	小结.....	122
<b>第 5 章</b>	<b>OpenGL 编程.....</b>	<b>123</b>
5.1	概述.....	124
5.1.1	OpenGL 基本功能.....	124
5.1.2	OpenGL 函数库.....	126
5.2	绘制 OpenGL 基本几何对象.....	129
5.2.1	定义点.....	129
5.2.2	OpenGL 的线定义.....	130
5.2.3	定义与绘制多边形.....	130
5.2.4	设置点、线和多边形属性.....	131
5.3	生成 OpenGL 程序.....	132
5.4	实现三维动画.....	135
5.4.1	OpenGL 变换的基本概念.....	135
5.4.2	帧缓存和动画.....	140
5.4.3	实现三维动画的编程步骤.....	141
5.5	在 OpenGL 中显示汉字.....	143
5.6	OpenGL 纹理映射.....	145
5.6.1	纹理的基本概念.....	145
5.6.2	纹理定义.....	146
5.6.3	纹理控制与映射.....	147
5.6.4	纹理坐标.....	149
5.6.5	使用纹理的编程步骤.....	150
5.7	小结.....	154
<b>第 6 章</b>	<b>多媒体编程.....</b>	<b>155</b>
6.1	多媒体编程概述.....	156
6.2	使用 DirectX 8.0 开发多媒体应用.....	156
6.2.1	DirectX 的功能.....	156
6.2.2	DirectX 的发展历史.....	156
6.2.3	DirectX 8.0 的新特性.....	158
6.3	使用 DirectAudio 制作可调音效的声音播放器.....	160
6.3.1	DirectAudio 简介.....	160

6.3.2	DirectAudio 编程示例 .....	161
6.4	使用 DirectDraw 绘制屏幕动画 .....	169
6.4.1	DirectDraw 简介 .....	169
6.4.2	DirectDraw 编程示例 .....	171
6.5	使用 DirectInput 制作鼠标画笔 .....	177
6.5.1	DirectInput 简介 .....	177
6.5.2	DirectInput 编程示例 .....	178
6.6	使用 DirectShow 制作媒体播放器 .....	189
6.6.1	DirectShow 简介 .....	189
6.6.2	DirectShow 编程示例 .....	191
6.7	小结 .....	200
<b>第 7 章</b>	<b>网络编程 .....</b>	<b>201</b>
7.1	网络编程概述 .....	202
7.2	远程拨号上网 .....	202
7.2.1	RAS 简介 .....	202
7.2.2	编程示例 .....	202
7.3	Web 风格的文件浏览器 .....	213
7.3.1	CDHtmlDialog 简介 .....	214
7.3.2	编程示例 .....	215
7.4	FTP 访问示例 .....	223
7.4.1	FTP 服务简介 .....	223
7.4.2	编程示例 .....	223
7.5	使用 CAIHttpclient 类构造 http 客户端 .....	230
7.5.1	编程实例 .....	231
7.6	小结 .....	244
<b>第 8 章</b>	<b>数据库编程 .....</b>	<b>245</b>
8.1	概述 .....	246
8.2	SQL 数据库访问引擎 .....	248
8.2.1	使用 MFC ODBC 类访问 SQL 数据库 .....	248
8.2.2	编程实例 .....	258
8.3	使用 DAO 编写数据库管理程序 .....	270
8.3.1	DAO 概述 .....	270
8.3.2	编程实例 .....	271
8.4	.NET 的新特性: SafeArrays .....	290
8.5	小结 .....	290

<b>第 9 章</b>	<b>ATL 编程</b>	<b>291</b>
9.1	概述	292
9.1.1	简介	292
9.1.2	关键技术	292
9.2	使用 ATL 开发 COM 服务器	293
9.2.1	基本步骤	293
9.2.2	运行结果	293
9.2.3	实际编程	294
9.3	使用 ATL 开发播放 CD 的控件	302
9.3.1	基本步骤	302
9.3.2	运行结果	302
9.3.3	实际编程	302
9.4	小结	308
<b>第 10 章</b>	<b>XML Web Service 编程</b>	<b>309</b>
10.1	Web 体系结构：过去与未来	310
10.2	XML 语言简介	312
10.2.1	XML 的起源和目的	312
10.2.2	XML 的特点与优点	314
10.2.3	XML 与 HTML 及 SGML 之间的关系	315
10.2.4	XML 的应用领域	316
10.3	SOAP 协议规范	317
10.3.1	简介	317
10.3.2	设计目标	318
10.3.3	SOAP 消息交换模型	318
10.3.4	与 XML 的关系	319
10.3.5	SOAP 封装	319
10.3.6	SOAP 编码规则	321
10.3.7	在 HTTP 中使用 SOAP	323
10.3.8	SOAP HTTP 示例	325
10.3.9	在 RPC 中使用 SOAP	325
10.4	Web Service 结构	326
10.4.1	概述	326
10.4.2	Web Service 描述语言 WSDL	330
10.4.3	UDDI	333
10.5	实例：使用 ATL Server 创建 Web Service	334
10.5.1	基本步骤	334
10.5.2	运行结果	334

10.5.3	实际编程 .....	335
10.6	实例：获取并格式化显示 XML 文件中的数据.....	342
10.6.1	基本步骤 .....	343
10.6.2	程序运行结果.....	343
10.6.3	实际编程 .....	343
10.7	小结 .....	355
<b>第 11 章</b>	<b>STL 编程 .....</b>	<b>357</b>
11.1	概述 .....	358
11.2	STL 组成 .....	358
11.2.1	STL 算法.....	359
11.2.2	STL 容器.....	360
11.2.3	STL 迭代器.....	360
11.2.4	STL 函数对象.....	361
11.2.5	STL 的其他部分.....	362
11.3	STL 编程环境 .....	365
11.3.1	头文件.....	365
11.3.2	名字空间.....	365
11.4	容器：双队列编程实例.....	366
11.4.1	概念 .....	366
11.4.2	功能描述.....	367
11.4.3	程序实现.....	369
11.4.4	运行结果.....	372
11.5	常用算法测试实例.....	372
11.5.1	<algorithm>算法实例 .....	372
11.5.2	<numeric>算法实例 .....	381
11.5.3	<functional>算法实例 .....	384
11.6	迭代器编程实例.....	389
11.6.1	普通迭代器实例 .....	389
11.6.2	反转型迭代器实例 .....	389
11.6.3	插入迭代器实例 .....	392
11.6.4	流迭代器实例.....	394

# 第 1 章

## Visual C++ .NET 简介

## 1.1 .NET 平台概述

.NET 为构建安全、可靠、高扩展性和高可用性的 Web Service 提供了从创建到配置与维护等一系列的应用程序模型和第一手的开发支持。

正如比尔·盖茨所言，.NET 将和 MS-DOS 和 Windows 一样成为计算领域中的里程碑。“.NET 框架开启了通向软件开发复兴的大门。”“.NET 实现之后，Internet 将走出泡沫。”无论从商务还是软件技术创新的角度考虑，.NET 平台都使人感到振奋。

.NET 使 Microsoft 成为 Web Service 软件供应商的战略方向，它包括两个方面：公司重新编写其应用程序和服务以通过 Internet 通信和共享数据，以及为开发商和集成商提供用于网络化解决方案的工具。

在开发等级上，.NET 框架在一个环境中集成了多种流行的程序语言。在 .NET 中可以共享类型，如同这些语言都具有同一个编译器一样。同时，.NET 在 Web 方面还进行了很多革新。这些革新使得 Web Service 开发和利用变得非常容易，包括使用可扩展标记语言 (XML) 格式化数据、使用简单对象访问协议 (SOAP) 在应用程序间传递数据，以及使用通用描述搜寻集成协议 (UDDI) 为 Web Service 提供目录。

### 1.1.1 .NET 平台产生的背景

近 10 年来，各行各业的人们潮水般地涌向 Internet。然而人们应用和理解软件的方式却一直停留使用 WWW 浏览，E-mail 信函，以及 FTP 下载等基本服务之上，距离 5 个“Any”（任何人，任何地方，任何时间，任何设备及任何手段）的 Internet 梦想还十分遥远。如果沿着传统的客户机/服务器模式走下去，即使能够创造足够强大、安全的中间件，在现有网络结构下的互连集成也将是举步维艰，代价高昂。

2000 年 10 月，Microsoft 提出“.NET”战略的构想，旨在建立一个无缝连接各个系统并充分注重开发人员体验的 Internet 系统，为开发人员创造一种安全和无障碍地在 Internet 上连接商务的运作方式。.NET 的策略是将 Internet 本身作为构建新一代操作系统的基础，对 Internet 和操作系统的设计思想进行合理延伸。这样，开发人员将能够创建出摆脱设备硬件束缚的应用程序，轻松实现 Internet 连接。.NET 平台包括一个相当广泛的产品家族，它们构建于 XML 和 Internet 产业标准之上，为开发人员提供 XML Web Services 的开发、管理、应用和体验。

.NET 将从连接到 Internet 的单一网站或设备上，转移到计算机、设备和服务群组上，使其通力合作，提供更广泛更丰富的解决方案。开发人员能够控制信息的传送方式、时间和内容，计算机、设备和服务相辅相成，从而提供丰富的服务。而不是像信息孤岛那样，由开发人员提供惟一的集成。.NET 允许企业使用开放的 Internet 协议将不同的应用程序和系统，甚至是运行在不同平台上的应用程序和系统联系起来，将它们无缝地集成为一个完整的系统。这样，数据和软件功能就能够无缝地从大型计算机转移到蜂窝电话上，极大地增加了获得和生成数据的能力。

.NET 的出现是软件界的一大革命，它带来的新特性解决了当今软件开发中的如下核心问题。

(1) 互操作性(Interoperability)、集成性(Integration)和应用程序的可扩展性(Extensibility)利用原有的开发工具很难实现,而且即使实现,也要付出高昂的代价。.NET 依靠 XML 技术标准消除了数据共享和软件集成的障碍。

(2) 在现有的开发环境中,开发人员不能在程序之间方便地共享数据或无法操作可访问的数据。而 XML 使得数据交换变得简单易行, .NET 对 XML 提供了强大的支持。

(3) .COM 公司和 Web 站点很难为开发人员提供有价值 and 可操作的数据,一个个信息孤岛很难与其他程序和服务很好合作,而 .NET 技术可以整合整个 Web 上的程序和服务。

(4) 终端用户的信息和数据个人隐私和安全性问题成为在 Web 上活动的一大障碍, .NET My Service 提供了一整套服务,增加了用户信息的可管理性和安全性。

### 1.1.2 .NET 平台的组成

Visual Studio.NET(以下简称 VS .NET)平台由以下 5 个部分组成。

#### 1. .NET 服务框架和开发工具

一个坚实、可靠并强大的编程框架是必须的, Microsoft.NET 框架充当了这一角色,并且是一个方便易用的 RAD 开发工具。

#### 2. .NET 服务器

能够提供广泛聚合和继承 XML Web Services 的服务器是搭建 Microsoft.NET 平台的后端基础,广泛支持 XML 的数据库、操作系统和商务交易系统将为 Microsoft.NET 平台提供一个坚实的后端基础。

#### 3. .NET 基础服务

密码认证、日历、文件存储和用户信息等这些基础服务必不可少, Microsoft 正在着力建设的 Microsoft.NET My Services 等基础性服务平台将为用户提供安全可靠的基础服务。

#### 4. .NET 终端设备

广泛连接 Internet 并体验 XML Web Services 的终端设备是实现的前端基础, PC、PDA, 以及各种嵌入式设备将在这个广阔的天地里发挥作用。

#### 5. .NET 用户体验

快捷、高效、方便,以及易用,并且能够满足人们各种各样需求的用户体验是 Microsoft.NET 的终极目的,也是 Microsoft.NET 的价值实现。将通信和计算两大 IT 主题通过 XML Web Services 完美地演化为人们日益增长的经验需求是 Microsoft.NET 平台中最具创造性及最具价值的部分。

在这 5 个组成部分中,“.NET 服务框架和开发工具”是整个 .NET 平台赖以生存和发展的核心,其他 4 个部分则紧紧围绕着 .NET 框架组织整合。一个健壮、高效并且安全的框架和一个经过检验的坚实结构对于现代软件开发是非常关键的,开发人员无需从底层开始完全靠自己来实现规则和数据处理,所要做的只是按照自己的商业逻辑组织服务应用。Microsoft.NET 框架为基于下一代 Internet 的 XML Web Services 及应用程序提供了一个托管的、安全和高效的执行环境。



### 1.1.3 .NET 技术框架内核

.NET 技术框架实现了从语言开发、代码编译、组件配置、程序运行，以及对象交互等各个层面的创新，既是 Microsoft 对几十年来软件开发的经验整合，也是面向未来基于 Web Service 应用的创新开拓。.NET 技术框架从底层的系统服务开始，提供了一个托管的代码执行环境——通用语言运行时(Common Language Runtime，简称 CLR)。在此之上，是为组件开发和运行提供的从基础编程，数据访问，程序装配等各种服务的一个坚实、强大且全面可重用类的集合。.NET 框架以组件形式包装这些可重用类，为程序开发提供 3 种基本编程模板：基于 ASP.NET 的 Web 表单应用，基于 ASP.NET 的 Web Service 和基于传统 GUI 交互的 Windows 应用。图 1-1 所示是 .NET 技术框架的一个总体描述。

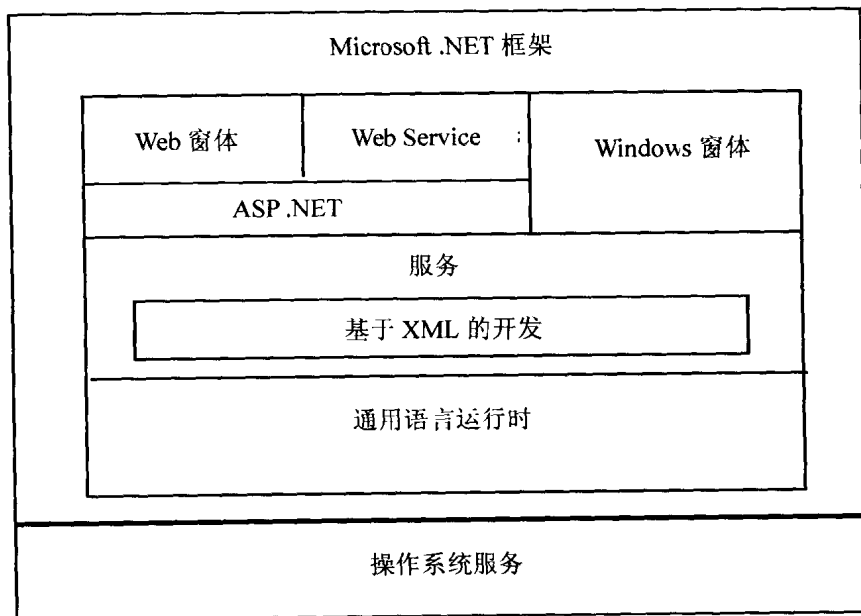


图 1-1 .NET 技术框架的总体描述

下面从通用语言运行时，.NET 框架类库，以及标准和语言支持 3 个方面来阐述 .NET 技术框架。

#### 1. 通用语言运行时

通用语言运行时为 .NET 应用程序提供了一个托管的代码执行环境，托管的意思是将原来由开发人员或操作系统做的工作分离出来交由通用语言运行时处理。这样的工作通常包括：内存管理、即时编译、组件自描述、安全管理和代码验证，以及其他一些系统服务。通用语言运行时是一个技术规范，无论程序使用什么语言编写，只要能编译成中间语言(IL)，即可在其支持下运行，这使得 .NET 应用程序得以独立于语言。通用语言运行时的设计目的是直接在应用程序运行环境中为现代基于组件的编程提供第一步的支持，它直接支持属性、事件、对象、继承性、多态性和接口。

自动垃圾收集器负责 .NET 应用程序运行时的内存的分配、对象布局及释放等内存管理