



Visual C++ .NET

数据库开发技术与实践

刘生平 编著



清华大学出版社

Visual C++ .NET

数据库开发技术与实践

刘生平 编著



清华大学出版社

北京

内 容 简 介

本书以如何建立一个 Visual C++ .NET 数据库应用程序为主线,从 Microsoft 通用数据库访问体系出发,将 Visual C++ .NET 支持的数据库访问技术 ODBC、DAO、OLE DB、ADO 和 ADO.NET 逐一进行了详细的介绍。以文字和片段代码相结合的方式详细地说明了使用各个技术建立数据库应用程序的基本步骤,并且提供了完整的示例工程以便于读者举一反三,全面掌握数据库访问技术。

本书条理清楚,内容全面、深入,叙述通俗易懂,适合有一定基础的大专院校师生、企业技术开发人员学习参考,也适合各类培训班学员学习 Visual C++ .NET 数据库开发技术。

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

Visual C++ .NET 数据库开发技术与实践/刘生平编著. —北京:清华大学出版社, 2005.1

ISBN 7-302-10028-4

I. V… II. 刘… III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 124663 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

组稿编辑: 欧振旭

文稿编辑: 鲁秀敏

封面设计: 姜凌娜

版式设计: 杨 洋

印 装 者: 北京鑫海金澳胶印有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印 张: 31 字 数: 710 千字

版 次: 2005 年 1 月第 1 版 2005 年 1 月第 1 次印刷

书 号: ISBN 7-302-10028-4/TP·6886

印 数: 1~5000

定 价: 49.00 元(附光盘 1 张)

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或 (010)62795704

前 言

在现代软件程序设计中，数据库程序设计是一个重要的组成部分。Microsoft 的 Windows 平台提供了多种数据库访问技术，每一种技术都具有自己的特点，在不同的应用中使用它们，在性能和效率上会有所差别。

Microsoft 提供了基于 Visual Studio .NET 集成开发环境的多种开发工具开发数据库应用程序，Visual C++ .NET 是其中的一种。由于 Visual C++ .NET 相对地接近于系统底层，因此使用它开发的数据库应用程序代码比较短小、精练，执行效率高。在某些特殊的应用中，必须使用它来开发数据库应用程序，以提高整个系统的性能和效率。

本书根据 Microsoft 通用数据库访问体系分为 6 个部分，第 1 部分讲述 Microsoft.NET 平台和 Microsoft 数据库开发技术的一些基本概念，其他 5 个部分分别讲述如何使用 Visual C++ .NET 开发工具建立基于 ODBC、DAO、OLE DB、ADO 和 ADO.NET 数据库访问技术的应用程序。每一部分以如何使用该技术建立一个 Visual C++ .NET 数据库应用程序为主线，以文字和片段代码相结合的方式详细地说明了使用该技术建立数据库应用程序的基本过程和步骤，并且提供了完整的示例工程来全面地展示该技术。

本书还将 5 种数据库开发技术进行了对比，并将它们之间的关联和数据转化进行了说明，从而使得本书在条理清楚的基础上，不但内容通俗易懂，而且融会贯通成一体，完整地读完本书，读者会有一种高屋建瓴的感觉。

尽管作者本人具有多年的开发经验，但是 Microsoft 平台上的 Visual C++ .NET 数据库开发是一个庞大的体系，需要讲解的知识点比较多，参考的资料也主要是微软的最新英文 MSDN，加上本人水平有限，难免出现一些错误，请读者不吝指正。

作 者
2004 年 6 月

目 录

第 1 部分 Visual C++ .NET 数据库开发技术概述

第 1 章 .NET 平台概述	2
1.1 .NET 平台	2
1.2 .NET 框架	3
1.2.1 公共语言运行库	3
1.2.2 .NET 框架类库	8
第 2 章 Visual C++ .NET 数据库访问技术概述	10
2.1 Visual C++ .NET 数据库访问技术	10
2.1.1 ODBC	10
2.1.2 DAO	12
2.1.3 OLE DB	12
2.1.4 ADO	13
2.1.5 ADO.NET	14
2.2 通用数据库访问架构体系	15
2.2.1 ADO 和 OLE DB	16
2.2.2 RDS 概述	17
2.2.3 MDAC SDK 新版本特点	19
2.3 Visual C++ .NET 数据库开发技术的特点	19
2.4 选择 Visual C++ .NET 数据库开发技术	20
2.4.1 选择 DAO 和 ODBC	20
2.4.2 选择 OLE DB	21
2.4.3 选择 ADO.NET 和 ADO	21
2.4.4 总结	23

第 2 部分 ODBC 数据库访问技术

第 3 章 ODBC 数据库访问技术概述	26
3.1 ODBC 概述	26
3.2 ODBC 的体系结构	26
3.3 ODBC 的数据操作类型	27
3.4 ODBC 的跟踪和诊断技术	28
3.4.1 返回代码	28
3.4.2 诊断记录	29

3.5	使用 ODBC 需要包含的头文件和库文件.....	31
3.6	配置 ODBC DSN	31
3.6.1	静态地配置 ODBC DSN.....	31
3.6.2	动态地配置 ODBC DSN.....	34
第 4 章	ODBC API 编程技术	39
4.1	ODBC API 概述	39
4.2	句柄	39
4.3	ODBC API 应用程序开发基本步骤	40
4.3.1	建立应用程序的 ODBC 环境.....	40
4.3.2	分配连接句柄和设置连接的属性	40
4.3.3	连接到数据源.....	42
4.3.4	构造和执行 SQL 语句	46
4.3.5	记录的添加、删除和更新.....	51
4.3.6	取回查询结果.....	53
4.3.7	断开同数据源的连接.....	55
4.3.8	释放 ODBC 环境	56
4.3.9	示例程序	56
4.4	典型 ODBC API 程序示例	58
4.4.1	定义全局变量和宏.....	58
4.4.2	定义统一的错误处理程序.....	59
4.4.3	定义主控程序.....	60
第 5 章	MFC ODBC 编程技术	63
5.1	MFC ODBC 概述	63
5.2	MFC ODBC 基本类概述	64
5.2.1	CDatabase	64
5.2.2	CRecordSet	65
5.2.3	CRecordView.....	68
5.3	MFC 封装 ODBC API 的类库技术.....	69
5.3.1	RFX.....	70
5.3.2	DDX.....	71
5.3.3	DDX 和 RFX 的关系及比较	73
5.4	MFC ODBC 应用程序框架和开发步骤	73
5.4.1	建立数据库连接.....	73
5.4.2	创建记录集.....	74
5.4.3	操作记录集.....	75
5.4.4	执行 SQL 语句.....	80
5.4.5	断开数据库连接.....	81

5.4.6 事务处理	81
5.4.7 示例程序	83

第 3 部分 DAO 数据库访问技术

第 6 章 DAO 数据库访问技术概述	86
6.1 DAO 概述	86
6.2 DAO 对象	86
6.3 DAO 的环境支持	87
6.4 MFC DAO 概述	87
6.5 DAO 对象到 MFC 类的映射	88
6.5.1 MFC 类和对应的 DAO 对象	88
6.5.2 MFC 如何管理未映射到类的 DAO 对象	88
6.5.3 MFC 中未公开的 DAO 对象	89
6.5.4 MFC 如何访问数据库引擎	89
6.5.5 MFC 和 DAO 的安全性	89
6.6 MFC DAO 基本类概述	89
6.7 MFC DAO 应用程序开发的基本步骤	91
6.7.1 建立数据库连接	91
6.7.2 创建记录集	91
6.7.3 操作记录集	93
6.7.4 事务处理	98
6.7.5 示例程序	99

第 4 部分 OLE DB 数据库访问技术

第 7 章 OLE DB API 编程技术	104
7.1 OLE DB 技术概述	104
7.1.1 通用数据访问技术	104
7.1.2 OLE DB 技术	104
7.2 OLE DB 对象	106
7.2.1 数据源对象	106
7.2.2 会话对象	106
7.2.3 命令对象	107
7.2.4 行集对象	107
7.2.5 事务对象	108
7.2.6 枚举器对象	108
7.2.7 错误对象	108
7.3 OLE DB API 应用程序的基本框架和示例	108

7.3.1	初始化环境.....	110
7.3.2	初始化数据源对象.....	110
7.3.3	获取会话对象和执行一个命令.....	114
7.3.4	从行集获取数据.....	116
7.3.5	示例程序.....	123
7.4	OLE DB API 的高级技术.....	124
7.4.1	行集的增强功能.....	124
7.4.2	处理大数据类型.....	126
7.4.3	数据操作.....	131
7.4.4	事务.....	138
第 8 章	OLE DB 使用者.....	140
8.1	OLE DB 使用者概述.....	140
8.2	OLE DB 使用者模板.....	140
8.2.1	数据源类和会话类.....	141
8.2.2	访问器类和行集合类.....	141
8.2.3	命令类和表类.....	143
8.2.4	用户记录类.....	150
8.2.5	模式行集合.....	152
8.3	OLE DB 使用者属性.....	152
8.3.1	db_source 属性.....	152
8.3.2	db_table 属性.....	154
8.3.3	db_command 属性.....	155
8.3.4	db_column 属性.....	158
8.3.5	db_param 属性.....	160
8.3.6	db_accessor 属性.....	161
8.4	建立 ATL OLE DB 使用者对象.....	162
8.4.1	使用向导建立 ATL OLE DB 使用者对象.....	162
8.4.2	手动建立 ATL OLE DB 使用者对象.....	165
8.5	建立 OLE DB 使用者应用程序.....	166
8.5.1	创建属性化 OLE DB 使用者 Win32 控制台应用程序.....	166
8.5.2	创建模板类 OLE DB 使用者 Win32 控制台应用程序.....	171
8.5.3	创建属性化 OLE DB 使用者 .NET 控制台应用程序.....	177
8.5.4	创建属性化 OLE DB 使用者 MFC 应用程序.....	179
第 9 章	OLE DB 使用者的增强功能.....	184
9.1	使用数据库属性简化数据访问.....	184
9.1.1	属性插入的用户记录类.....	184
9.1.2	设置行集合属性.....	185

9.1.3 示例程序	185
9.2 向导生成的访问器中的字段状态数据成员	188
9.3 获取数据的方式	190
9.4 使用模式行集合获取元数据或模式信息	191
9.4.1 目录/模式模型	193
9.4.2 查询模式信息时使用限制	193
9.4.3 示例程序	194
9.5 使用书签	194
9.5.1 实例化书签	194
9.5.2 从提供程序中请求书签列	195
9.5.3 将书签项添加到列映射	196
9.5.4 示例程序	196
9.6 更新行集合	197
9.6.1 支持更新操作	198
9.6.2 在行中设置数据	198
9.6.3 向行集合中插入行	198
9.6.4 从行集合中删除行	199
9.6.5 立即更新和推迟更新	199
9.6.6 示例程序	200
9.7 遍历简单行集合	201
9.8 使用存储过程	203
9.8.1 定义存储过程	204
9.8.2 存储过程中的输出参数	205
9.8.3 存储过程中的行计数	205
9.8.4 扩展存储过程	206
9.8.5 创建扩展存储过程	206
9.8.6 向数据库中注册扩展存储过程	207
9.8.7 调用扩展存储过程	209
9.8.8 向扩展存储过程中添加功能	209
9.8.9 使用一个存储过程返回多个结果集	210
9.8.10 调试扩展存储过程	210
9.9 使用访问器	211
9.9.1 确定需要使用的访问器类型	211
9.9.2 在一个行集合上使用多个访问器	212
9.9.3 使用 CDynamicAccessor 访问器	213
9.9.4 使用 CDynamicStringAccessor 访问器	214
9.9.5 使用 CDynamicParameterAccessor 访问器	215
9.9.6 重写动态访问器	217

9.9.7 使用手动访问器.....	218
9.10 检索 XML 数据.....	221
9.10.1 使用 CStreamRowset 检索 XML 数据.....	222
9.10.2 使用 CXMLAccessor 检索 XML 数据.....	222
9.10.3 示例程序.....	224
9.11 使用 OLE DB 记录视图.....	228
9.12 将接收通知功能添加到 OLE DB 使用者.....	229
9.13 在 OLE DB 中使用现有的 ADO 记录集.....	230
9.14 OLE DB 行集的持久性操作.....	230
9.14.1 将行集持久化.....	230
9.14.2 将已持久化的行集加载到 ADO 记录集.....	231
9.14.3 示例程序.....	231
9.15 在行集合中检索 BLOB 对象.....	232
第 10 章 OLE DB 提供者.....	234
10.1 OLE DB 提供者概述.....	234
10.2 OLE DB 规范级别支持.....	234
10.3 为何要创建 OLE DB 提供程序.....	234
10.4 OLE DB 提供程序的类型.....	235
10.5 OLE DB 提供者体系结构.....	235
10.5.1 数据源和会话.....	235
10.5.2 强制接口和可选接口.....	236
10.5.3 属性映射.....	239
10.5.4 用户记录.....	241
10.6 创建 OLE DB 提供程序.....	242
10.6.1 创建 OLE DB 提供程序实例.....	242
10.6.2 创建简单的只读提供程序示例.....	251
10.6.3 创建可更新的提供程序.....	259
10.7 增强 OLE DB 提供程序功能.....	267
10.7.1 添加接口到提供程序.....	267
10.7.2 在提供程序中引用属性.....	268
10.7.3 在提供程序中设置属性.....	269
10.7.4 在提供程序中动态绑定列.....	269
10.7.5 在提供程序中支持自由线程处理.....	271
10.7.6 转换提供程序不支持的数据.....	271
10.7.7 提供程序支持通知消息的功能.....	271
10.7.8 提供程序支持模式行集合.....	273
10.7.9 在提供程序中实现对书签的支持.....	278
10.8 调试提供程序.....	283

10.9 测试提供程序	283
10.10 一致性测试提供程序.....	284
10.11 OLE DB 提供程序资源池和服务	285
10.11.1 OLE DB 应用程序中的资源池	285
10.11.2 启用和禁用 OLE DB 服务	286

第 5 部分 ADO 数据库访问技术

第 11 章 ADO 技术概述	290
11.1 ADO 概述.....	290
11.1.1 ADO 体系结构.....	290
11.1.2 数据提供者.....	291
11.1.3 服务提供者.....	291
11.1.4 服务组件	292
11.1.5 ADO 技术特点.....	292
11.2 ADO 对象模型.....	292
11.3 ADO 编程模型.....	294
11.3.1 ADO 编程模型概述.....	294
11.3.2 ADO 编程模型细节	294
11.3.3 ADO 的特有数据类型	299
11.3.4 ADO 编程接口	305
11.4 ADO 事件模型.....	307
11.4.1 事件类型	307
11.4.2 捕获 ADO 事件.....	308
11.4.3 示例程序	308
第 12 章 ADO 开发应用程序技术	312
12.1 ADO 应用程序框架和开发流程.....	312
12.1.1 在 Visual C++中使用 ADO	312
12.1.2 建立数据库连接.....	321
12.1.3 建立 ADO 记录集.....	324
12.1.4 操作记录集.....	332
12.1.5 错误处理	341
12.1.6 事务处理	342
12.1.7 一个完整的示例项目.....	345
12.2 ADO 记录集的持久性操作.....	349
12.2.1 ADO 记录集与内存之间的转化操作	349
12.2.2 ADO 记录集与 XML 文件之间的转化操作	350
12.2.3 示例程序	350

12.3	ADO 读写 BLOB 技术	356
12.3.1	读取 BLOB 数据	356
12.3.2	写 BLOB 数据	356
12.3.3	示例程序	357
12.4	ADO 记录集与 OLE DB 行集	358
12.4.1	ADORecordsetConstruction 接口	358
12.4.2	示例程序	359
12.5	VC++对 ADO 的扩展	360
12.5.1	IADORecordBinding 接口	360
12.5.2	支持 ADO 扩展所需的头文件	361
12.5.3	绑定 Recordset 的字段	361
12.5.4	绑定条目和绑定条目宏	361
12.5.5	示例程序	363

第 6 部分 ADO.NET 数据库访问技术

第 13 章	ADO.NET 技术概述	368
13.1	ADO.NET 概述	368
13.2	ADO.NET 的设计目标	369
13.3	ADO.NET 结构	370
13.3.1	XML 和 ADO.NET	370
13.3.2	ADO.NET 组件	370
13.3.3	选择 DataReader 与 DataSet	371
13.4	ADO.NET 与 ADO	372
13.4.1	读取数据	372
13.4.2	DataSet、DataTable 和 Recordset 对象	374
13.4.3	转换现有代码	375
13.4.4	数据更新	376
13.4.5	对 XML 的扩展支持	377
13.5	.NET Framework 数据提供程序概述	379
13.5.1	.NET Framework 数据提供程序	379
13.5.2	名称空间组织	383
13.5.3	选择 .NET Framework 数据提供程序	384
13.6	使用 .NET Framework 数据提供程序访问数据	385
13.6.1	连接到数据源	385
13.6.2	执行命令	391
13.6.3	执行数据库操作和修改数据	393
13.6.4	使用 DataReader 检索数据	396
13.6.5	从 DataAdapter 填充 DataSet	398

13.6.6	使用 DataAdapter 和 DataSet 更新数据库.....	401
13.6.7	自动生成的命令.....	403
13.6.8	将参数用于 DataAdapter	410
13.6.9	将存储过程用于命令.....	416
13.7	ADO.NET DataSet	423
13.7.1	DataTableCollection	424
13.7.2	DataRelationCollection	425
13.7.3	ExtendedProperties	425
13.7.4	创建 DataSet.....	425
13.7.5	向 DataSet 添加 DataTable	425
13.7.6	添加表间关系和导航表间关系.....	426
13.7.7	DataSet 事件.....	426
13.7.8	示例程序	427
第 14 章	ADO.NET 开发应用程序技术.....	429
14.1	获取数据库架构信息.....	429
14.1.1	从 DataReader 中获取架构信息.....	429
14.1.2	示例程序	430
14.2	获取错误信息	432
14.2.1	为 DataSet 对象获取扩展错误信息	432
14.2.2	获取数据提供程序提供的异常信息.....	433
14.2.3	示例程序	433
14.3	调用参数化存储过程.....	437
14.3.1	用 DataReader 对象获取结果记录集、返回值和输出参数	437
14.3.2	用命令对象的 ExecuteScalar 成员方法检索参数值	437
14.3.3	用命令对象的 ExecuteNonQuery 成员方法检索参数值	437
14.3.4	示例程序	438
14.4	执行事务	447
14.4.1	使用 ADO.NET 执行事务	448
14.4.2	在分布式事务中登记.....	448
14.4.3	示例程序	449
14.5	在 OleDbConnection 对象中使用 Data Link Files.....	450
14.5.1	创建 UDL 文件	451
14.5.2	使用 UDL 文件的优缺点	451
14.5.3	示例程序	452
14.6	ADO.NET 操作 XML 技术	453
14.6.1	将 DataSet 对象中的数据持续化到 XML	453
14.6.2	将 XML 数据读到 DataSet 对象	454
14.6.3	利用 OpenXML 方法执行批 (Bulk) 更新和批插入.....	455

14.6.4	示例程序	455
14.7	ADO.NET 读写 BLOB 技术	463
14.7.1	从数据库中获取 BLOB 值	463
14.7.2	将 BLOB 值写入数据库	463
14.7.3	使用 Byte 数组一次读写完整的 BLOB 数据	464
14.7.4	以 Chunking 方式读写 BLOB 数据	464
14.7.5	示例程序	465
14.8	访问 ADO 记录集	476
14.8.1	使用 ADO 记录集填充 DataSet	477
14.8.2	示例程序	477

第 1 部分

Visual C++ .NET 数据库开发技术概述

第 1 章 .NET 平台概述

1.1 .NET 平台

.NET 平台就是 Microsoft 的 XML Web 服务平台。不论操作系统或编程语言有何差别, XML Web 服务能使应用程序在 Internet 上传输和共享数据。

Microsoft®.NET 平台包含广泛的产品系列,它们都是基于 XML 和 Internet 行业标准构建,提供从开发、管理、使用到体验 XML Web 服务的每一方面。XML Web 服务将成为今天正在使用的 Microsoft 的应用程序、工具和服务器的一部分,并且将打造出全新的产品以满足所有业务需求。

Microsoft 正在 5 个方面创建 .NET 平台,即工具、服务器、XML Web 服务、客户端和 .NET 体验。

Microsoft®.NET 平台主要包含下面几个部分。

(1) 开发工具,它主要包括 4 个部分。

- ① 一系列开发语言,包括 C#、VB.NET、J# 和 VC++ .NET 等;
- ② 一套与语言无关的开发工具,包括 Visual Studio.NET 等;
- ③ .NET 框架,主要建立 Web、Windows 应用和 Web 服务;
- ④ 公共语言运行库,它是 .NET 框架应用程序执行引擎。

(2) .NET 企业服务器,它是一套电子商业应用设施,在它上面运行 XML Web Services、SQL Server 2000、Exchange Server 2000、BizTalk Server 2000 和其他的商业构件服务器,这些服务器主要用来进行关系数据存储、数据的交换和 B2B (business-to-business) 交易。

(3) Web Services,它是一系列的商业 Web Services,主要包括客户服务 (.NET MyServices)、验证和识别服务 (.NET Passport) 及通信服务 (.NET Alerts)。开发人员可以使用这些服务建立需要知道用户身份信息的应用程序。

(4) .NET 设备软件,包括 Windows XP、Windows Me、Windows CE、Windows Embedded、.NET 框架和 .NET 简洁框架。

.NET 平台的组成体系如图 1-1 所示。

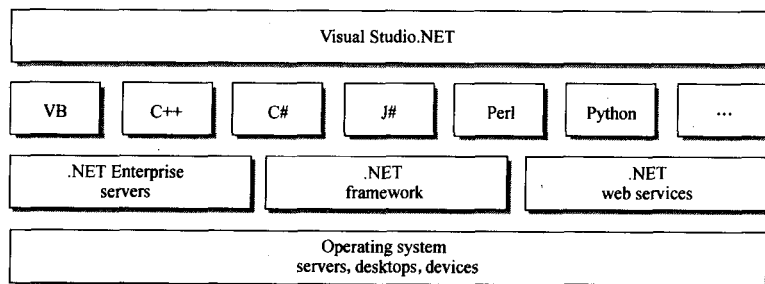


图 1-1 .NET 平台的组成体系

1.2 .NET 框架

.NET 框架是.NET 平台的编程模型,它是一种新的计算平台,简化了在高度分布式 Internet 环境中的应用程序开发。.NET 框架旨在实现下列目标:

(1) 提供一个一致的面向对象的编程环境。编成人员不用考虑对象代码是在本地存储和执行,还是在本地执行但在 Internet 上分布,或者是在远程执行的。

(2) 提供一个将软件部署和版本控制冲突最小化的代码执行环境。

(3) 提供一个保证代码(包括由未知的或不完全受信任的第三方创建的代码)安全执行的代码执行环境。

(4) 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。

(5) 使开发人员的经验在面对类型大不相同的应用程序(如基于 Windows 的应用程序和基于 Web 的应用程序)时保持一致。

(6) 按照工业标准生成所有通信,以确保基于.NET 框架的代码可与任何其他代码集成。

.NET 框架具有两个主要组件:公共语言运行库和.NET 框架类库。公共语言运行库是.NET 框架的基础。可以将公共语言运行库看作一个在执行时管理代码的代理,它提供核心服务(如内存管理、线程管理和远程处理),强制实施严格的类型安全以及可确保安全性、可靠性和其他形式的代码准确性。事实上,代码管理的概念是运行库的基本原则。以运行库为目标的代码称为托管代码,而不以运行库为目标的代码称为非托管代码。.NET 框架的另一个主要组件是类库(有时被称为 Base Framework),它是一个综合性的、面向对象的可重用类型集合,可以使用它开发多种应用程序,这些应用程序包括传统的命令行或图形用户界面(GUI)应用程序,也包括基于 ASP.NET 所提供的最新创新的应用程序(如 Web 窗体和 XML Web Services)。

.NET 框架可由非托管组件承载,这些组件将公共语言运行库加载到它们的进程中,并启动托管代码的执行,从而创建一个可以同时利用托管和非托管功能的软件环境。.NET 框架不但提供若干个运行库宿主,而且还支持第三方运行库宿主的开发。例如:ASP.NET 承载运行库为托管代码提供可伸缩的服务器端环境,它直接使用运行库以启用 ASP.NET 应用程序和 XML Web Services;Internet Explorer 是承载运行库(以 MIME 类型扩展的形式)的非托管应用程序的一个示例,使用 Internet Explorer 承载运行库使得能够在 HTML 文档中嵌入托管组件或 Window 窗体控件,以这种方式承载运行库使得托管移动代码(类似于 Microsoft® ActiveX® 控件)成为可能,但是它具有只有托管代码才能提供的重大改进,例如,不完全受信任的执行和安全的独立文件存储等。

.NET 框架的组成体系如图 1-2 所示。

1.2.1 公共语言运行库

公共语言运行库管理内存、线程执行、代码执行、代码安全验证、代码编译以及其他系统服务。这些功能是在公共语言运行库上运行的托管代码所固有的。若要使运行库能够向托管