

► 1CD-ROM ◀

Delphi.NET

程序设计

周兴华 张国平 来骏 编著



- 详尽讲解 Delphi .NET 的基本语法
- 从基础知识到高级特性及应用，应有尽有
- 结合语言的内在技术、技巧和近百个示例
- 章后精选习题，配书光盘附部分习题答案

 科学出版社
北京科海电子出版社

Delphi.NET 程序设计

周兴华 张国平 来骏 编著

科学出版社
北京科海电子出版社

内 容 提 要

本书由浅入深地讲述了 Delphi.NET 集成开发环境、面向对象程序设计基础、Delphi.NET 语法基础知识、基本流程控制、过程与函数、数据结构、报表的创建、文件管理、DBExpress 应用、VCL 窗体应用、调用 Win32 代码、多线程应用程序、Delphi 数据库基础、数据访问和 ASP 应用程序开发。在讲述这些知识的过程中，作者尽量采用浅显易懂的语言并试图将相关知识融于示例，使读者能够将理论与实践应用相对照，从而在最短时间内领会 Delphi.NET 的思想精髓，并利用它进行快速应用程序的开发。

本书语言简明，其中伴有大量实用性较强的示例，适用于 Delphi.NET 语言的初级用户，对于有一定工作经验的开发人员也有较好的参考价值。

图书在版编目 (CIP) 数据

Delphi.NET 程序设计/周兴华编著. —北京:

科学出版社, 2004

ISBN 7-03-014698-0

I. D… II. 周… III. 软件工具—程序设计

IV. TP311.56

中国版本图书馆 CIP 数据核字 (2004) 第 127543 号

责任编辑: 何武 / 责任校对: 科海

责任印刷: 科海 / 封面设计: 林陶

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

北京市艺辉印刷有限公司印刷装订

科学出版社发行 各地新华书店经销

*

2004 年 12 月第一版

开本: 16 开

2004 年 12 月第一次印刷

印张: 27.75

印数: 1-4 000

字数: 675 千字

定价: 39.00 元 (1CD)

(如有印装质量问题, 我社负责调换)

前 言

2003年11月, Borland软件公司推出Borland Delphi 8 for the Microsoft .NET Framework (简称Delphi.NET)。新版本的Delphi的设计目的是让开发者能够用现有的Delphi语言与.NET Framework技巧来建造基于.NET Framework的应用。而且, 开发者能够应用现有的在Delphi应用源代码与技巧方面的投资, 简化将Windows应用移植到.NET Framework的过程。

Delphi .NET针对健壮Web开发的微软ASP.NET Web Forms与XML Web Services技术, 采用了Windows窗体和可视组件库(VCL)控件及其包含的几百个类和控件; 针对高性能数据库应用的微软ADO.NET与Borland数据供应者(BDP), 使用Borland 企业核心对象(ECO)来加速设计驱动的开发。以上措施使我们能够快速开发功能丰富的可靠的应用程序。

对于Delphi开发者来讲, 使用Delphi .NET开发平台的优势是: 可以使用熟悉的Delphi语言语法在Delphi .NET开发平台上开发程序; 可以将现存的Delphi源代码迁移到.NET框架中; 采用现有的数据库技术, 简便地移植数据库应用到.NET框架上。

对于开发团队来讲, 由于在Delphi.NET开发平台可以共享能够运行在.NET框架上的超过20种语言的组件与源代码, 因此把用Delphi开发的产品移植到.NET框架, 可以增强产品的可靠性、安全性、可互操作性及性能。在今天就可以针对下一代Windows平台开发应用, 增强产品的竞争优势。

本书由浅入深地讲述了Delphi.NET集成开发环境、面向对象程序设计基础、Delphi.NET语法基础知识、基本流程控制、过程与函数、数据结构、报表的创建、文件管理、DBExpress应用、VCL窗体应用、调用Win32代码、多线程应用程序、Delphi数据库基础、数据访问和ASP应用程序开发。在讲述这些知识的过程中, 本书尽量采用浅显易懂的语言并试图将相关知识融于示例, 使读者能够将学习与实践应用对照结合起来, 从而在最短时间内领会Delphi.NET的思想精髓, 并利用它进行快速应用程序的开发。

本书由邵谦谦和许小荣总策划, 由周兴华、张国平、来骏和刘磊共同执笔编写。此外, 管永东、张英、杨旭、何晶晶、施少鹏、蔡宇、刘峰、周小杰、徐红、高林宇、施伟伟、张爱华、缪珩珩、黄瑜、张一琳、冒小飞、张蓓、张英、朱勇、冯志刚、潘华、金伟、缪辉、戴旭东、许宝建、蔡东军、梁小军和刘小松等同志在整理材料方面给予了编者很大的帮助, 在此, 对他们表示衷心的感谢。

由于时间仓促, 加之水平所限, 书中的缺点和不足之处在所难免, 欢迎读者提出宝贵的意见和建议, 编者的联系方式为: myworks2001@263.net。

编 者

2004年10月

目 录

第1章 Delphi .NET集成开发环境..... 1

1.1 .NET Framework 1.1简介	1
1.1.1 .NET Framework.....	1
1.1.2 .NET Framework 1.1 版本的 新功能.....	5
1.2 Delphi.NET简介	6
1.3 Delphi.NET可视化开发环境简介.....	8
1.3.1 Delphi.NET的基本形式	8
1.3.2 面向对象编程的概念.....	8
1.3.3 对象的属性、事件和方法.....	9
1.3.4 组件.....	10
1.3.5 进入Delphi.NET集成开发环境	15
1.4 开发环境的优化	20
1.4.1 自定义工具栏.....	20
1.4.2 编辑环境设置.....	21
1.4.3 工程设置.....	22
1.5 一个简单的应用程序	23
1.6 本章小结	28
1.7 习题	28

第2章 面向对象程序设计基础..... 29

2.1 面向对象的基本概念	29
2.2 类和对象	30
2.2.1 类的声明.....	30
2.2.2 对象和类变量.....	31
2.3 类的方法	32
2.3.1 方法的声明.....	32
2.3.2 构造函数与析构函数.....	34
2.3.3 方法指令符.....	35
2.3.4 抽象方法.....	38
2.3.5 重载方法与重定义方法.....	38
2.4 类的特性	40
2.4.1 声明特性.....	40

2.4.2 特性限定符.....	41
------------------	----

2.4.3 数组特性.....	43
-----------------	----

2.4.4 特性重载和重定义.....	44
---------------------	----

2.5 类成员的可见性	45
-------------------	----

2.6 类引用	47
---------------	----

2.7 继承	48
--------------	----

2.8 命名空间	50
----------------	----

2.8.1 工程默认命名空间和命名空间 声明.....	50
--------------------------------	----

2.8.2 搜索命名空间.....	51
-------------------	----

2.8.3 命名空间的使用.....	52
--------------------	----

2.9 本章小结	53
----------------	----

2.10 习题	53
---------------	----

第3章 Delphi.NET语法基础..... 54

3.1 保留字与标识符	54
3.1.1 保留字.....	54
3.1.2 标识符.....	55
3.1.3 指令符.....	56
3.2 数据类型	57
3.2.1 数值型数据.....	58
3.2.2 字符数据.....	59
3.2.3 布尔型数据.....	60
3.3 常量与变量	60
3.3.1 常量.....	60
3.3.2 变量.....	62
3.4 运算符与表达式	63
3.4.1 算术运算符.....	63
3.4.2 位运算符.....	65
3.4.3 字符串运算符.....	65
3.5 常用函数与过程	65
3.5.1 数学函数.....	66
3.5.2 字符处理函数.....	67
3.5.3 日期时间函数.....	71

3.5.4 顺序类型函数.....	73	5.2.2 参数传递方式.....	118
3.5.5 TObject.ToString	74	5.2.3 使用默认参数.....	119
3.6 语句	76	5.2.4 函数与过程的重载.....	121
3.6.1 语句的书写规则.....	76	5.3 子程序的嵌套与递归	121
3.6.2 语句格式中的符号约定.....	76	5.3.1 子程序的嵌套.....	122
3.6.3 赋值语句.....	77	5.3.2 子程序的递归.....	123
3.6.4 注释语句.....	77	5.4 变量的作用域	125
3.7 本章小结	78	5.4.1 公有变量与私有变量.....	125
3.8 习题	78	5.4.2 全局变量与局部变量.....	127
第4章 基本流程控制	80	5.4.3 全局变量的声明.....	128
4.1 选择结构	80	5.5 Delphi.NET的程序结构.....	132
4.1.1 条件表达式.....	81	5.5.1 主程序.....	132
4.1.2 If语句	82	5.5.2 程序单元.....	132
4.1.3 Case语句	88	5.5.3 程序段.....	134
4.2 循环结构	90	5.6 本章小结	134
4.2.1 For语句	90	5.7 习题	134
4.2.2 While语句	93	第6章 数据结构	135
4.2.3 Repeat-Until语句	95	6.1 数组类型	135
4.2.4 综合结构.....	96	6.1.1 静态数组.....	135
4.3 跳转语句	97	6.1.2 动态数组.....	140
4.3.1 Break语句和Continue语句.....	97	6.2 指针类型	145
4.3.2 Label语句和Goto语句.....	98	6.2.1 指针类型的声明.....	146
4.4 结构化异常处理	99	6.2.2 指针运算符.....	147
4.4.1 Raise语句	100	6.2.3 指针运算.....	148
4.4.2 Try...Except语句	101	6.2.4 无类型指针.....	151
4.4.3 Try...Finally语句.....	103	6.2.5 字符指针类型.....	151
4.4.4 异常类.....	104	6.3 枚举类型	152
4.5 方法与技巧	106	6.3.1 枚举类型定义与变量的声明.....	152
4.5.1 命名规则.....	106	6.3.2 枚举类型的运算.....	153
4.5.2 设置代码模板.....	107	6.4 子界类型	154
4.6 本章小结	108	6.4.1 子界类型定义与变量的声明.....	154
4.7 习题	109	6.4.2 子界类型变量的运算.....	155
第5章 过程与函数.....	110	6.5 集合类型	155
5.1 过程与函数	110	6.5.1 集合与集合类型.....	155
5.1.1 过程.....	111	6.5.2 集合类型定义与变量声明.....	156
5.1.2 函数.....	115	6.5.3 集合类型数据的运算.....	157
5.2 向子程序传递参数	117	6.6 记录类型	160
5.2.1 形式参数与实际参数.....	118	6.6.1 记录类型的声明.....	160

6.6.2 访问记录字段与with语句.....	161	第9章 DBExpress应用	214
6.6.3 记录的变体部分	162	9.1 DBExpress简介	214
6.7 本章小结	164	9.1.1 DBExpress核心接口介绍.....	214
6.8 习题	165	9.1.2 DBExpress的应用开发.....	215
第7章 报表	168	9.1.3 DBExpress的特点.....	215
7.1 初识Rave报表	168	9.2 DBExpress组件	216
7.1.1 Rave报表的特点.....	168	9.2.1 TSQLConnection组件	216
7.1.2 Rave开发环境简介.....	169	9.2.2 TSQLDataSet组件	220
7.2 Rave报表组件	169	9.2.3 TSQLQuery组件	222
7.2.1 TRvProject组件	170	9.2.4 TSQLStoredProc组件	224
7.2.2 TRvSystem组件	172	9.2.5 TSQLTable组件	226
7.2.3 其他组件.....	173	9.2.6 TSQLMonitor组件.....	228
7.3 报表设计器	173	9.3 DBExpress程序开发	229
7.3.1 启动报表设计器.....	174	9.3.1 DBExpress程序开发方法.....	229
7.3.2 报表设计器概述.....	174	9.3.2 DBExpress程序实例分析.....	231
7.3.3 报表设计器应用	176	9.4 本章小结	245
7.4 简易报表系统的开发	182	9.5 习题	245
7.4.1 程序界面设计	183	第10章 VCL窗体应用	246
7.4.2 报表设计	184	10.1 认识VCL	246
7.4.3 编写代码.....	186	10.1.1 VCL简介	247
7.4.4 运行结果.....	186	10.1.2 VCL.NET与.NET Framework 的关系.....	248
7.5 本章小结	187	10.2 VCL.NET组件	248
7.6 习题	187	10.2.1 窗体.....	249
第8章 文件管理	189	10.2.2 组件	249
8.1 文件类型和标准过程	189	10.2.3 VCL的命名空间.....	251
8.1.1 文本文件	189	10.3 VCL窗体应用	251
8.1.2 类型文件.....	192	10.3.1 VCL窗体开发.....	251
8.1.3 无类型文件.....	193	10.3.2 VCL组件导入.....	254
8.1.4 文件对话框组件.....	195	10.3.3 VCL窗体的图形应用.....	259
8.1.5 Win3.1相关组件	198	10.3.4 VCL窗体的多媒体应用.....	273
8.1.6 Win3.2相关组件	201	10.4 本章小结	278
8.2 文件管理常用函数和过程	201	10.5 习题	279
8.2.1 文件操作常用函数和过程.....	201	第11章 调用Win32代码.....	280
8.2.2 目录操作常用函数和过程.....	203	11.1 托管与非托管代码	280
8.2.3 驱动器操作常用函数.....	208	11.1.1 托管代码.....	280
8.2.4 文件名操作常用函数.....	209	11.1.2 非托管代码.....	281
8.3 本章小结	212	11.2 COM Interop概述.....	282
8.4 习题	213		

11.2.1	COM Interop术语	283	13.2.4	本地数据库和远程数据库	338
11.2.2	.NET Framework SDK的 COM Interop工具	286	13.3	Delphi.NET数据库应用程序的体系 结构	340
11.2.3	COM Interop程序集在IDE 中的应用	298	13.3.1	单层数据库应用程序	340
11.3	COM Interop实现对非托管代码的 调用	299	13.3.2	两层数据库应用程序	340
11.3.1	向COM Server中添加引用	299	13.3.3	多层数据库应用程序	341
11.3.2	向Tool Palette中添加ActiveX 控件	301	13.3.4	选择合适的体系结构	341
11.4	本章小结	304	13.3.5	可伸缩性	342
11.5	习题	304	13.4	Delphi.NET的BDE管理器介绍	343
第12章	多线程应用程序	305	13.4.1	BDE管理器介绍	343
12.1	进程和线程	305	13.4.2	配置BDE数据库并建立数据库 别名	344
12.1.1	进程和线程的概念	305	13.5	Delphi.NET数据库应用程序的 开发	346
12.1.2	线程调度	306	13.5.1	Client/Server的基本概念和 体系结构	346
12.2	TThread类	307	13.5.2	Delphi.NET数据库应用程序的 开发方法及步骤	347
12.2.1	TThread类的属性	307	13.5.3	交付数据库应用程序	349
12.2.2	TThread类的方法	308	13.6	Delphi.NET数据库综合应用程序 实例	349
12.2.3	创建线程类	309	13.6.1	应用程序的功能划分	350
12.2.4	线程的初始化操作	310	13.6.2	应用程序的窗体实现	350
12.2.5	实现线程对象的功能	310	13.7	本章小结	354
12.3	线程的同步	313	13.8	习题	355
12.4	线程的优先级	318	第14章	数据访问	356
12.5	多线程的示范程序	319	14.1	ADO.NET概述	356
12.6	本章小结	322	14.1.1	ADO.NET结构	356
12.7	习题	322	14.1.2	ADO.NET的设计目标	357
第13章	Delphi.NET数据库的基本 概念	323	14.1.3	SQL语言概述	358
13.1	数据库系统概述	323	14.1.4	BDP.NET概述	360
13.1.1	使用数据库	323	14.2	BDP.NET数据访问组件	361
13.1.2	数据库管理系统	324	14.2.1	数据访问组件简介	361
13.1.3	数据库应用程序	325	14.2.2	数据集组件 (DataSet)	363
13.2	Delphi.NET的数据库特性及 数据源	326	14.2.3	视图组件 (DataView)	365
13.2.1	Delphi.NET的数据库特性	326	14.2.4	数据适配器组件 (DataAdapter)	366
13.2.2	数据库组件应用实例	334	14.3	BDP.NET数据访问技术	368
13.2.3	Delphi可以访问的数据源	337			

14.3.1 BDP.NET数据库连接	369	15.3.2 Web服务的必要条件	402
14.3.2 BDP.NET数据库显示	373	15.3.3 Web服务方案	403
14.3.3 BDP.NET数据库编程	374	15.3.4 ASP.NET的Web服务文件	404
14.4 本章小结	376	15.3.5 Web服务与安全	404
14.5 习题	377	15.3.6 Web服务应用	406
第15章 ASP.NET应用程序开发.....	378	15.4 本章小结	412
15.1 ASP.NET概述	378	15.5 习题	412
15.1.1 运行环境配置	379	附录 各章习题答案.....	413
15.1.2 ASP.NET结构	379	第1章 Delphi.NET集成开发环境.....	413
15.1.3 Web窗体组件	380	第2章 面向对象程序设计基础	413
15.1.4 Web窗体数据存取	383	第3章 Delphi.NET语法基础.....	416
15.1.5 ASP.NET的命名空间	383	第4章 基本流程控制	419
15.1.6 ASP.NET应用程序示例	385	第5章 过程与函数	419
15.2 ASP.NET组件	387	第6章 数据结构	420
15.2.1 Components页组件.....	387	第7章 报表	423
15.2.2 HTML Elements组件页	387	第8章 文件管理	424
15.2.3 Data Components组件页	389	第9章 DBExpress应用	425
15.2.4 DB Web组件页	390	第10章 VCL窗体应用	426
15.2.5 Borland Data Provider组件页...392		第11章 调用Win32代码.....	427
15.2.6 Enterprise Core Objects组件页 392		第12章 多线程应用程序	428
15.2.7 ASP.NET的数据库应用	392	第13章 Delphi.NET数据库的基本概念...428	
15.3 Web Service	401	第14章 数据访问	429
15.3.1 Web服务结构	402	第15章 ASP.NET应用程序开发	430

第1章 Delphi .NET 集成开发环境

Borland Delphi 8 For the Microsoft .NET Framework（简称Delphi.NET）集成开发环境是.NET变革的演进。Delphi.NET集成开发环境得到了业内人士的高度认可和赞同。使用为Microsoft.NET Framework准备的Delphi.NET集成开发环境，开发者可以立即使用Pascal语言、Framework技巧以及应用程序源代码开发.NET应用程序。

本章将介绍面向对象编程的概念、Delphi.NET集成开发环境的基本情况以及利用Delphi.NET集成开发环境开发程序的基本步骤和主要方法。通过本章的学习，读者将了解Delphi.NET的产品情况，熟悉Delphi.NET集成开发环境。

1.1 .NET Framework 1.1简介

Microsoft .NET Framework 1.1是用于.NET 平台的编程模型，它的关键组件是公共语言运行库和.NET Framework类库。.NET Framework 提供了托管执行环境、简化的开发和部署并集成了各种编程语言。.NET Framework 软件开发工具包包含示例、编译器和命令行工具，旨在帮助你生成基于.NET Framework 技术的应用程序。

1.1.1 .NET Framework

.NET Framework是一种新的计算平台，它简化了在高度分布式Internet环境中的应用程序开发。.NET Framework旨在实现下列目标：

- 提供一个一致的面向对象编程环境，而无论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，或者是远程执行；
- 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；
- 提供一个保证代码安全执行的代码执行环境；
- 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。

为了确保令人满意的性能，.NET Framework 必须具备如表1-1、表1-2所示的最低客户端和服务端硬件要求和操作系统要求。若要使用附加功能，如ASP.NET、COM+服务和SQL Server .NET数据提供程序，则需要具备如表1-3所示的软件要求。

表1-1 硬件要求

方案	处理器	RAM
客户端	Pentium 90 MHz 或更快	96 MB 或更高
服务器	Pentium 133 MHz 或更快	256 MB 或更高

表1-2 操作系统要求

方案	操作系统
客户端	Microsoft Windows 98
	Microsoft Windows 98 Second Edition
	Microsoft Windows Millenium Edition
	Microsoft Windows NT 4.0 Workstation With Service Pack 6.0a
	Microsoft Windows NT 4.0 Server With Service Pack 6.0a
	Microsoft Windows 2000 Professional
	Microsoft Windows 2000 Server
	Microsoft Windows 2000 Advanced Server
	Microsoft Windows 2000 Datacenter Server
	Microsoft Windows XP Home Edition
	Microsoft Windows XP Professional
	Microsoft Windows Server 2003
	Microsoft Windows Server 2003 R2
服务器	Microsoft Windows 2000 Professional With Service Pack 2.0
	Microsoft Windows 2000 Server With Service Pack 2.0
	Microsoft Windows 2000 Advanced Server With Service Pack 2.0
	Microsoft Windows 2000 Datacenter Server With Service Pack 2.0
	Microsoft Windows XP Professional
	Microsoft Windows Server 2003

表1-3 软件要求

方案	功能	软件
客户端	SQL Server .NET 数据提供程序	数据访问组件 (MDAC) 2.6或更高版本
	对系统管理信息的访问	Windows Management Instrumentation
服务器	SQL Server .NET 数据提供程序	Microsoft 数据访问组件 (MDAC) 2.7
	ASP.NET	Microsoft Internet 信息服务 (IIS) 5.0

.NET Framework具有两个主要组件：公共语言运行库和.NET Framework类库。

公共语言运行库（Common Language Runtime，CLR）是.NET Framework的基础。你可以将CLR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。事实上，代码管理的概念是CLR的基本原则。以CLR为目标的代码称为托管代码，而不以CLR为目标的代码称为非托管代码。

.NET Framework 的另一个主要组件是类库，它是一个综合性的面向对象的可重用类型集合，你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（Graphical User Interface, GUI）应用程序，也包括基于ASP.NET的最新应用程序。

1. 公共语言运行库的功能

公共语言运行库可以管理内存、线程执行、代码执行、代码安全验证、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。

CLR强制实施代码访问安全。例如，编程人员可以让嵌入在Web页中的可执行文件在屏幕上播放动画或唱歌，但不能访问他们的个人数据、文件系统或网络。这样，CLR的安全功能就使通过Internet部署的合法软件能够具有特别丰富的功能。运行库通过实现称为通用类型系统（Common Type System, CTS）的严格类型验证和代码验证体系结构来加强代码可靠性。CTS确保所有托管代码都可以自我描述。Microsoft和各种第三方语言编译器生成符合CTS的托管代码，这意味着托管代码可在严格实施类型保真和类型安全的同时使用其他托管类型和实例。

CLR提高了开发人员的工作效率。例如，程序员可以用他们选择的开发语言编写应用程序，却仍能充分利用其他开发人员用其他语言编写的运行库、类库和组件。任何选择以CLR为目标的编译器供应商都可以这样做。以 .NET Framework为目标的语言编译器使得用该语言编写的现有代码可以使用.NET Framework的功能，这大大减轻了现有应用程序移植过程的工作负担。

CLR旨在增强性能。尽管公共语言运行库提供许多标准运行库服务，但是它从不解释托管代码。一种称为实时（Just-In-Time, JIT）编译的功能使所有托管代码能够以执行系统的本机语言运行。同时，内存管理器排除了出现零碎内存的可能性，并增大了内存引用区域以进一步提高性能。

2. .NET Framework 类库

.NET Framework类库是一个与公共语言运行库紧密集成的可重用类型集合。该类库是面向对象的，并提供托管代码可从中导出功能的类型，这不但使.NET Framework类库易于使用，而且减少了学习.NET Framework 的新功能所需要的时间。此外，第三方组件可与.NET Framework中的类无缝集成。

例如，.NET Framework集合类实现了一组可用于开发你自己的集合类的接口。你的集合类将与.NET Framework中的类无缝地混合。

作为一种面向对象的类库，.NET Framework类型使你能够完成一系列常见编程任务，包括字符串管理、数据收集、数据库连接以及文件访问等任务。除这些常见任务之外，类库还支持多种专用开发方案。例如，可使用.NET Framework开发下列类型的应用程序和服务：控制台应用程序、Windows GUI应用程序（Windows窗体）、ASP.NET应用程序、XML Web Services及Windows服务等。

3. 客户端应用程序开发

客户端应用程序在基于Windows的编程中最接近于传统风格的应用程序。这些是在桌

面上显示窗口或窗体从而使用户能够执行任务的应用程序类型。客户端应用程序包括诸如字处理程序和电子表格等应用程序，还包括自定义的业务应用程序（如数据输入工具、报告工具等）。客户端应用程序通常使用窗口、菜单、按钮和其他GUI元素，并且它们可以访问本地资源（如文件系统）和外围设备（如打印机）。

另一种客户端应用程序是作为Web页通过Internet部署的传统ActiveX控件（现在被托管Windows窗体组件替代），此应用程序在本机执行，可以访问本地资源，并包含图形元素。

过去，开发人员将C/C++与Microsoft基础类（MFC）或应用程序快速开发（RAD）环境（如Microsoft Visual Basic）一起使用来创建这样的应用程序。.NET Framework 将这些现有产品的特点合并到单个且一致的开发环境中，该环境大大简化了客户端应用程序的开发。

4. 服务器应用程序开发

在托管领域中，服务器端应用程序是通过运行库宿主实现的。非托管应用程序承载公共语言运行库，后者使你的自定义托管代码可以控制服务器的行为。此模型在获得主服务器的性能和可伸缩性的同时提供公共语言运行库和类库的所有功能。

图1-1显示在不同服务器环境中运行托管代码的基本网络架构。在应用程序逻辑通过托管代码执行时，服务器（如IIS和SQL Server）可执行标准操作。

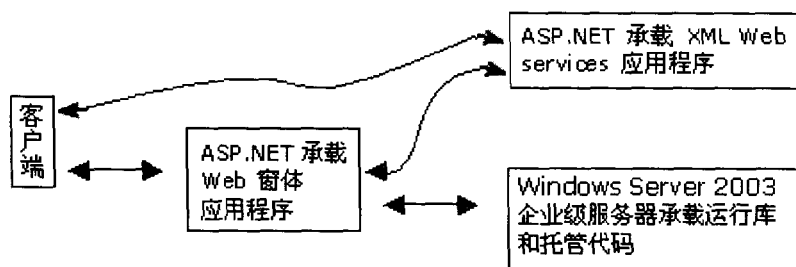


图1-1 服务器端托管代码

ASP.NET是使开发人员能够使用.NET Framework开发基于Web的应用程序的宿主环境。但是，ASP.NET不止是一个运行库宿主，它是使用托管代码开发Web站点和通过Internet分布的对象的完整结构。Web窗体和XML Web Services都将IIS和 ASP.NET用作应用程序的发布机制，并且两者在 .NET Framework 中都具有支持类集合。XML Web Services作为基于Web技术的重要发展，是类似于常见Web站点的分布式服务器端应用程序组件。但与基于Web的应用程序不同，XML Web Services组件不具有UI并且不以浏览器（如Internet Explorer和Netscape Navigator）为目标。XML Web Services由旨在供其他应用程序使用的可重用软件组件组成，所谓的其他应用程序包括传统的客户端应用程序、基于Web的应用程序以及其他XML Web Services。因此，XML Web Services技术正迅速地将应用程序开发和部署推向高度分布式Internet环境。

1.1.2 .NET Framework 1.1 版本的新功能

在.NET Framework 1.0 版的基础上, .NET Framework 1.1 版进行了扩展, 不仅对现有的功能做了改进, 还新增了一些功能, 并对文档进行了增强。本部分内容描述以前独立安装而现在包括在.NET Framework 1.1版中的技术以及对现有技术所做的主要更改。

1. ASP.NET 移动组件

ASP.NET移动组件扩展了.NET Framework和Visual Studio .NET, 提供了对移动电话和个人数字助理(PDA)等移动(无线)设备的支持。.NET Framework 1.1版将移动组件合并到.NET Framework 和Visual Studio .NET分发软件包中。

由于移动组件现在是.NET Framework的一部分, 因此术语已经做了相应的更改以便与.NET文档中使用的约定相匹配, 移动组件文档也已合并为更大的.NET Framework文档集。ASP.NET移动组件替换了Mobile Internet Toolkit这个名称。

ASP.NET移动组件扩展了ASP.NET服务器组件, 以便适应呈现Web应用程序的移动设备。通过浏览器检测, 移动组件能够适应各种个人设备的功能, 从全功能的PDA浏览器到小的5行×20字符的移动电话显示。这种自适应呈现功能可以处理很多单调的设备特定呈现决定, 使你可以将注意力集中在Web应用程序逻辑上。

2. ADO.NET 的改动

ODBC .NET Framework现在随.NET Framework在命名空间 System.Data.Odbc下提供。使用1.0版的开发人员需从<http://www.microsoft.com/china/msdownload/visual>下载 ODBC .NET Framework数据提供程序(命名空间= Microsoft.Data.Odbc)。Oracle .NET Framework数据提供程序现在随.NET Framework在命名空间System.Data.OracleClient下提供。使用1.0版的开发人员可以从<http://www.microsoft.com/china/msdownload/visual>下载 Oracle .NET Framework数据提供程序。

ADO.NET现在包括以下功能:

- DataReader对象现在公开HasRows属性, 可以确定是否无需调用Read就返回行;
- Connection 对象现在具有EnlistDistributedTransaction方法, 可以在分布式事务中启用手动登记。

3. 并行(Side-by-Side)执行

.NET Framework 1.1版支持并行执行。并行执行是在同一台计算机上存储和执行应用程序或组件的多个版本的能力。这意味着在同一台计算机上可以同时存在运行库的多个版本, 并且允许同时存在使用其中某个运行库版本的应用程序和组件的多个版本。

并行执行并不意味着托管应用程序与运行库或组件的其他版本兼容。相反, 它意味着托管应用程序可以选择它执行时所使用的运行库和组件, 而且运行库、应用程序和组件的多个版本可以在同一台计算机上共存。

4. .NET Framework 安全性的改动

在1.0 和1.1 版中,从运行库代码访问安全性系统接收不完全受信任代码的应用程序不能调用共享托管库,除非库编写器通过使用AllowPartiallyTrustedCallersAttribute属性特别允许它们调用。如果打算通过部分受信任的代码使用库,你需要知道有哪些库不能用于你的代码。在.NET Framework 1.1 版中, System.Web.dll、System.Web.Mobile.dll 和 System.Web.RegularExpressions.dll 都包含在具有AllowPartiallyTrustedCallersAttribute且可以通过部分受信任的代码调用的程序集列表中。

5. 宿主环境中的 ASP.NET 安全性

现在,ASP.NET在基于Web的应用程序中支持部分信任,能够更好地保证单个Web 服务器上承载的多个应用程序的安全。尽管应用程序运行所使用的操作系统帐户对应用程序施加了安全限制,公共语言运行库的代码访问安全性系统仍然能够根据你指定的策略,为选定的应用程序强制额外的限制。

ASP.NET提供<trust>配置指令,使你能为应用程序配置代码访问安全级别。如果部分受信任的ASP.NET应用程序要调用共享托管库,则这些库必须包含允许通过部分受信任的代码调用的AllowPartiallyTrustedCallersAttribute属性。

6. .NET Framework 中的 IPv6 支持

.NET Framework 1.1版支持最新的Internet协议(通常称为IP第6版或简称为 IPv6)。设计该协议的目的是为了大幅增加用于在Internet中标识通信终结点的地址空间,以便与它的持续增长相适应。System.NET命名空间、ASP.NET和XML Web Services均支持IPv6。

7. .NET Framework 支持向后和向前兼容性。

在.NET Framework的上下文中,向后兼容性意味着使用.NET Framework的较早版本创建的应用程序可以在更高的版本上运行。相反,向前兼容性意味着使用.NET Framework的更高版本创建的应用程序可以在较早的版本上运行。

1.2 Delphi.NET简介

Borland Delphi 8 for the Microsoft .NET Framework (也称Delphi.NET)是.NET 变革的演进,使用为Microsoft .NET Framework准备的Delphi RAD开发环境,马上就能简化向.NET 的过渡过程。这些开发包括VCL Forms、Windows Forms、ASP.NET、Web Forms、Web业务、ADO.NET等等。开发者可以立即使用Delphi语言与Framework技巧以及许多在Delphi应用程序源代码的现有投资,马上开发Windows .NET应用程序。

Delphi .NET包括下列解决方案:用于设计驱动开发的Borland Enterprise Core Objects (ECO),用于Microsoft .NET Framework的Borland Optimizait Profiler,以便帮助开发者识别并克服性能危害。

Borland的Microsoft .NET Framework战略是提供领先的开发解决方案,这些解决方案能

够无缝地集成应用程序生命周期组件,以便形成一个完整的企业.NET应用程序生命周期管理(ALM)解决方案。Borland的 Microsoft .NET战略完全支持Microsoft技术,也包括了对与主要企业供应商技术集成的全部特性支持,这些主要的企业供应商包括Oracle、IBM与Sun(Java)。Borland的.NET战略构筑在对Microsoft .NET Framework的完全支持上,包括ADO.NET、ASP.NET、Windows Forms、Web Forms与通过XML Web 业务或与J2EE的本地、可管理的连接实施的应用程序互操作性。

Borland Delphi 8 for the Microsoft .NET Framework为构建整个企业的ADO.NET数据库应用提供了灵活性,可用于企业数据库的建设。用于ADO.NET的Borland DataProvider可为大量的主要企业级数据库提供高性能支持,更快、更方便且更灵活地进行ADO.NET开发。用于ADO.NET的Borland DataProvider可灵活生成用于一个供应商的数据库原型,并可以分发到另一个供应商的数据库,自动.NET数据类型映射降低了学习曲线。

Delphi 8 for the Microsoft .NET Framework有3个版本:针对模型驱动的应用设计与开发的Borland Delphi架构版、针对利用Microsoft ADO.NET建造企业级数据库应用的团队的Borland Delphi企业版以及针对在Microsoft .NET Framework上建造Web与GUI应用的个人的Borland Delphi专业版。

Delphi 8 for the Microsoft .NET Framework的特性与优点如下所述:

- 使用Borland Delphi 8与Microsoft .NET Framework,改善了应用过程中的互操作性、安全性与可靠性;
- 使用现有的Delphi 语言与Framework技巧,可以马上开始开发.NET;
- 重复利用现有的Delphi投资,使得Windows应用向.NET 的过渡变得轻松;
- 使用Borland Enterprise Core Objects (ECO)模型驱动的 框架可以加速开发生命周期;
- 使用Delphi 8与ASP.NET可以开发高性能的Web解决方案;
- 用于ADO.NET 的高性能企业数据库开发;
- 统一的Delphi IDE可带来更加丰富的开发体验;
- 加速.NET 应用程序生命开发周期;
- 提升分布式远端开发组的表现,改善分布式远端开发组之间的沟通;
- 使用内含的、功能强大的第三方解决方案,提高应用程序的性能。

Borland Delphi 8 for the Microsoft .NET Framework的3个版本都有软件保证书。Borland是一个全球性的组织机构,在欧洲、南美、亚太与美国设有办事处。欲寻找当地的Borland办事处,请登录http://www.borland.com/company/borland_worldwide.html。欲了解更多的信息,请登录Borland的网址: <http://www.borland.com/delphi>。Borland开发者网址有新闻组、开发者社区信息, <http://bdn.borland.com/>上面也有更多的信息。

1.3 Delphi.NET可视化开发环境简介

Delphi.NET是全新的可视化编程环境，它使用了Microsoft Windows图形用户界面的许多先进特性和设计思想，采用了当今世界上最快的编辑器、弹性可重复利用的面向对象程序语言（Object-Oriented Language）和最为领先的数据库技术。对于广大的程序开发人员来讲，使用Delphi.NET开发应用软件，无疑会大大地提高编程效率。

1.3.1 Delphi.NET的基本形式

Delphi.NET应用程序的应用程序框架是其“骨架”。在骨架上即使没有附着任何东西，仍可以严格地按照设计运行。应用程序框架通过提供所有应用程序共有的东西，为用户应用程序的开发打下了良好的基础，为你做好了一切基础工作，只是不处理任何事情。在空白窗口的背后，应用程序框架正在等待用户的输入。Delphi.NET把Windows编程的回调、句柄处理等繁复过程都放在一个不可见的覆盖物下面，这样你可以轻松从容地对可视组件进行编程。

Delphi.NET是基于Object Pascal语言的可视化集成开发工具。利用Delphi.NET编程，可以快速、高效地开发出各类程序，尤其在数据库和网络方面，Delphi.NET更是一个十分理想的软件开发平台。可视化开发环境通常分为3个组成部分：编辑器、调试器和窗体设计器，和大多数现代RAD（Rapid Application Development，快速应用开发）工具一样，这3部分是协同工作的。当用户在窗体设计器中工作时，Delphi.NET在后台自动为正在窗体中操纵的组件生成代码，用户还可以自己在编辑器中加入代码来定义应用程序的行为，同时可以在同一个编辑器中通过设置断点和监控点来调试程序。

简单来说，“可视化”就是使用Delphi.NET的Object Pascal语言，利用它所提供的“组件”来创建对象。

1.3.2 面向对象编程的概念

面向对象程序设计（Object-Oriented Programming，OOP）是Delphi.NET诞生的基础。OOP立意于创建软件重用代码，具备更好地模拟现实世界环境的能力，这使它被公认为是自上而下编程的优胜者。它通过给程序中加入扩展语句，把函数“封装”进Windows编程所必需的“对象”中。面向对象编程语言使得复杂的工作条理清晰、编写容易。Delphi.NET是完全面向对象的，这就使得Delphi.NET成为一种触手可及的促进软件重用的开发工具，从而具有强大的吸引力。

Delphi.NET使用“可视化”的编程方法。程序员不必自己建立对象，利用Delphi.NET提供的可视“组件”，只要在提供的程序框架中加入完成一定功能的代码，不必考虑按精细次序执行的步骤就可以建立由若干小程序组成的大程序，这些小程序由用户启动事件激发，这样就可以快速创建强大的用户程序。Delphi.NET允许在一个具有真正OOP扩展的可视化编程环境中使用Object Pascal语言，这种组合使得可视化编程与面向对象的开发 框架