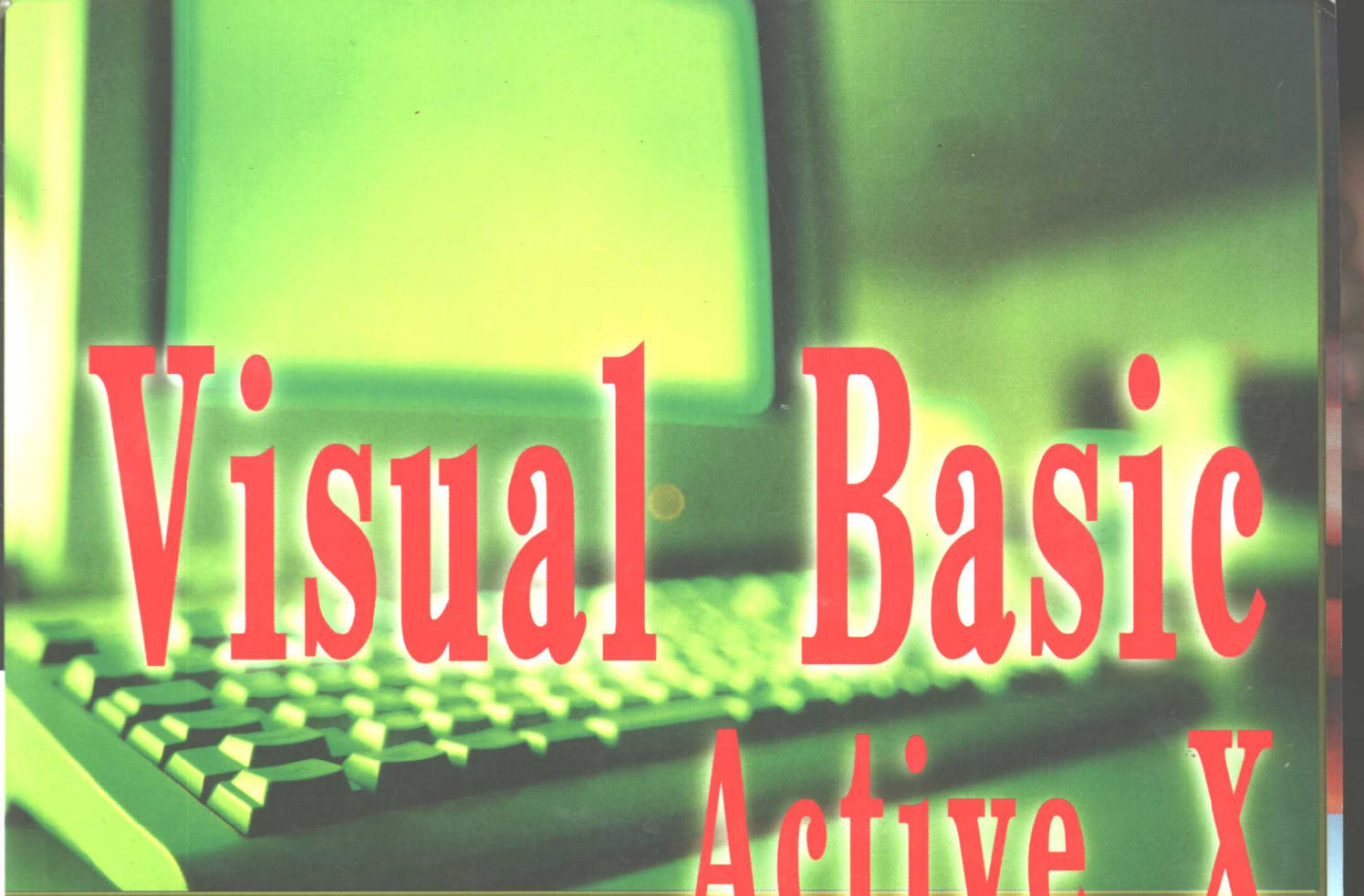




Visual Basic Active X

编程实例 详解

人民邮电出版社
www.pptph.com.cn

王清贤 王红霞 王磊 编著

Visual Basic Active X 编程实例详解

王清贤 王红霞 王磊 编著

人民邮电出版社

内 容 提 要

本书通过大量的精选实例,全面、系统地介绍了使用 Visual Basic 进行 ActiveX 编程的全过程和技巧,内容覆盖了 ActiveX 编程的各个方面:ActiveX 控件、ActiveX DLL、ActiveX EXE、ActiveX 文档。

全书共分为两大部分:ActiveX 控件部分和构件服务器部分。其中,ActiveX 控件部分是本书的重点,其中讲述的控件实例全部都经过精挑细选,不仅演示了 VB ActiveX 控件的编程机制和技巧,而且具有很高的实用价值。关于 ActiveX DLL、ActiveX EXE 和 ActiveX 文档的编程则在第二部分中讲述。

本书的内容翔实,结构合理,实例丰富,是学习 VB ActiveX 编程的难得好书,适合于利用 Visual Basic 编写 ActiveX 程序的软件开发人员阅读参考。

Visual Basic Active X 编程实例详解

◆ 编 著 王清贤 王红霞 王 磊

责任编辑 潘春燕

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@pptph.com.cn

网址 <http://www.pptph.com.cn>

北京汉魂图文设计有限公司制作

北京鸿佳印刷厂印刷

新华书店总店北京发行所经销

◆ 开本:787×1092 1/16

印张:30.75

字数:763 千字

2000 年 9 月第 1 版

印数:1-5 000 册

2000 年 9 月北京第 1 次印刷

ISBN 7-115-08752-0/TP·1800

定价:45.00 元

编 者 的 话

你知道 Visual Basic 应用程序可以具有同 Microsoft Excel 一样的分析与计算功能，同 Microsoft Word 一样的格式化文档功能，同 Microsoft Jet 数据库引擎一样的存储管理数据功能吗？

所有这些功能，或者比这更强的功能，通过使用 Visual Basic 创建的 ActiveX 部件都可以实现。ActiveX 部件是一段可重复使用的程序代码和数据，由用 ActiveX 技术创建的一个或多个对象组成。应用程序可以使用现有的部件，比如包含在 Microsoft Office 应用程序中的部件、各种各样制造厂商所提供的代码部件、ActiveX 文档或 ActiveX 控件（通常称为 OLE 控件）中含有的部件。如果你有 Visual Basic 专业版或企业版软件，你也可以开发自己的 ActiveX 控件。

使用 Visual Basic 可以创建包括从代码库到可以自动执行的应用程序的所有类型的 ActiveX 部件。具体地说，可以包括下列格式的 ActiveX 部件：ActiveX 控件、ActiveX DLL、ActiveX EXE 和 ActiveX 文档。所有这些格式的 ActiveX 部件的相关编程内容，本书都涉及到了。

本书分为两大部分：ActiveX 控件部分和构件服务器部分。其中，ActiveX 控件部分是本书的重点。本部分通过大量精选的控件实例，系统、全面地介绍了使用 Visual Basic 进行 ActiveX 控件编程的全过程以及其中所用到的各种技巧。而构件服务器部分则详细讲述了如何使用 Visual Basic 编写 ActiveX DLL、ActiveX EXE 和 ActiveX 文档。全书内容翔实、实例丰富，不仅涉及了 ActiveX 编程的各个方面，而且具有很高的实用价值。书中所有实例都已调试通过，可以直接应用到实际的程序开发中。

本书由王清贤、王红霞、王磊主编，参加本书编写、录排和制作的人员有孙丽君、张代平、孟昭景、于霞、李强、曾刚、于峰等。由于水平有限，加之时间紧迫，本书难免会有疏漏和遗憾之处，诚望广大读者多提宝贵意见。

编者
2000 年 7 月

目 录

第一部分 ActiveX 控件编程	1
第 1 章 VB 编程入门	3
1.1 Visual Basic 简介	3
1.2 Visual Basic 编程基本概念	3
1.2.1 Windows 工作机制	4
1.2.2 面向事件的编程模式	4
1.2.3 交互式开发环境	4
1.3 集成开发环境	5
1.3.1 菜单条	5
1.3.2 上下文菜单	5
1.3.3 工具栏	6
1.3.4 工具箱	6
1.3.5 工程管理器窗口	6
1.3.6 属性窗口	7
1.3.7 对象浏览器	7
1.3.8 窗体设计器	8
1.3.9 代码编辑器	8
1.3.10 窗体布局窗口	8
1.3.11 立即、本地和监视窗口	8
1.4 面向对象编程	9
1.4.1 理解对象和类的关系	9
1.4.2 对象的属性、方法和事件	9
1.4.3 使用其它应用程序的对象	10
1.5 使用控件编程	11
1.5.1 控件属性	12
1.5.2 控件方法	12
1.5.3 在工程中添加控件	13
1.5.4 从工程中删除控件	14
1.6 ActiveX 部件简介	14
1.6.1 ActiveX 部件的特性	15
1.6.2 选择 ActiveX 部件	16
第 2 章 ActiveX 控件入门	17
2.1 ActiveX 控件简介	17

2.2 基础概念	17
2.2.1 控件类与控件实例	18
2.2.2 控件与控件部件	18
2.2.3 容器与定位	18
2.2.4 接口与外观	18
2.2.5 设计时实例与运行时实例	18
2.3 快速入门	19
2.3.1 准备工作	19
2.3.2 创建 ControlDemo 工程	20
2.3.3 添加测试工程	21
2.3.4 设计时运行 Gradient 控件	22
2.3.5 UserControl 对象的存活期	25
2.3.6 绘制 Gradient 控件	28
2.3.7 添加控件属性	29
2.3.8 添加属性页	33
2.3.9 为 Gradient 控件添加事件	39
2.3.10 添加其它实现代码	40
2.3.11 编译 ControlDemo 部件	44
2.4 控件存活期和关键事件	46
2.4.1 控件存活期	46
2.4.2 UserControl 的关键事件	47
2.4.3 控件实例的实体化和再实体化	48
2.5 小结	50
第 3 章 进度条控件实例	52
3.1 定义控件功能	52
3.2 创建控件工程	53
3.3 添加测试工程	54
3.4 绘制控件	57
3.5 声明变量	58
3.6 添加属性	58
3.7 添加 UserControl_Resize 事件	61
3.8 添加属性页	62
3.9 添加控件实现代码	64
3.10 添加测试代码	66
3.11 编译控件	67
3.12 控件改进	68
3.12.1 功能改进	68
3.12.2 绘制控件	68

3.12.3 事件代码	69
3.12.4 测试控件	70
3.12.5 编译控件	71
3.13 小结	72
第 4 章 三维效果显示控件	73
4.1 Control 3D 控件功能描述	73
4.2 创建 Control 3D 控件工程	74
4.3 添加测试工程	75
4.4 绘制 Control 3D 控件	77
4.5 声明变量	78
4.6 添加控件属性	79
4.7 添加 Initialize 和 Resize 事件代码	83
4.8 添加属性页	84
4.8.1 添加属性页	85
4.8.2 设计控件通用属性页	85
4.8.3 连接属性页	87
4.9 添加控件实现代码	89
4.9.1 剔除命令码数字串中的无效数字	89
4.9.2 添加 PaintTargetControls 方法	91
4.9.3 实现 FindTargetControls 函数	92
4.9.4 实现 CheckForTag 函数	94
4.9.5 实现 DrawLine 过程代码	94
4.10 添加测试代码	98
4.11 编译 Control 3D 控件	100
4.12 小结	100
第 5 章 倒记时报警时钟控件	101
5.1 CntDwnClock 控件功能描述	101
5.2 创建 CountDownClock 控件工程	101
5.3 添加测试工程	103
5.4 绘制 CntDwnClock 控件	105
5.5 声明变量	106
5.6 添加控件属性	108
5.7 添加 Resize 事件	120
5.8 添加属性页	121
5.8.1 添加属性页	121
5.8.2 设计控件 time 属性页	121
5.8.3 设计控件“杂项”属性页	126

5.8.4 连接属性页	128
5.9 添加控件实现代码	130
5.9.1 添加内部时钟控件 tmrTime 的 Timer 事件代码	130
5.9.2 实现 SetTime 过程	131
5.9.3 实现 CheckTime 过程	132
5.9.4 实现设置时间的方法	134
5.9.5 实现时钟倒记时开始方法	135
5.9.6 实现时钟倒记时停止方法	135
5.9.6 实现修改当前时间的方法	137
5.10 添加测试代码	138
5.11 编译 CntDwnClock 控件	142
第 6 章 滑动条控件实例	143
6.1 控件功能描述	143
6.2 创建控件工程	143
6.3 添加测试工程	145
6.4 绘制 HCSSlider 控件	148
6.5 声明变量	151
6.6 添加属性	151
6.7 添加 UserControl_Resize 事件	154
6.8 添加控件实现代码	155
6.9 添加测试代码	157
6.10 编译控件	157
第 7 章 点击计数控件	159
7.1 HitCounter 控件功能描述	159
7.2 创建 HitCount_Project 控件工程	159
7.3 添加测试工程	161
7.4 绘制 HitCounter 控件	166
7.5 声明变量	167
7.6 添加控件属性	168
7.7 添加 Resize 事件	173
7.8 添加控件实现代码	174
7.8.1 加载图像数据的实现代码 LoadNumerls	174
7.8.2 添加控件显示的代码	176
7.8.3 添加控件公共方法	179
7.9 添加测试代码	180
7.10 编译 HitCounter 控件	185
7.11 小结	186

第 8 章 调整控件大小的 Resizer 控件	187
8.1 控件功能描述	188
8.2 创建 Resizer 控件工程	188
8.3 添加测试工程	189
8.4 绘制 Resizer 控件	193
8.5 声明变量	193
8.6 添加控件属性	195
8.7 添加 Resize 事件	197
8.8 添加控件实现代码	197
8.8.1 响应父窗口的 Load 事件	197
8.8.2 响应父窗口的 Resize 事件	198
8.8.3 实现 Rebuild 过程	199
8.8.4 实现 Refresh 过程	201
8.9 测试控件	206
8.10 编译 Resizer 控件	208
第 9 章 文本滚动控件	210
9.1 TextScroller 控件的功能描述	211
9.2 创建 TextScroller 控件工程	211
9.3 添加测试工程	213
9.4 绘制 TextScroller 控件	216
9.5 声明变量	217
9.6 添加控件属性	218
9.7 添加属性页	227
9.7.1 添加属性页	227
9.7.2 设计控件通用属性页	228
9.7.3 连接属性页	232
9.8 添加控件实现代码	235
9.8.1 控件的清除代码	235
9.8.2 添加时钟子控件的 Timer 事件代码	236
9.9 测试控件	239
9.10 编译 TextScroller 控件	243
第 10 章 任务栏图标控件	245
10.1 SystemTray 控件功能描述	245
10.2 创建 SystemTray_Project 控件工程	246
10.3 添加测试工程	247
10.4 绘制 SystemTray 控件	249
10.5 声明变量	250

10.5.1 添加函数 Shell_NotifyIcon 声明	250
10.5.2 函数 Shell_NotifyIcon 的功能	252
10.5.3 “NOTIFYICONDATA”数据类型定义	253
10.5.4 添加常量定义	254
10.5.5 添加其它变量和事件声明	256
10.6 添加控件属性	258
10.7 添加 Resize 事件	261
10.8 添加属性页	262
10.8.1 添加通用属性页	262
10.8.2 设计控件通用属性页	263
10.8.3 连接通用属性页	267
10.9 添加控件实现代码	268
10.9.1 实现增加任务栏图标方法	269
10.9.2 实现修改任务栏图标属性的方法	270
10.9.3 实现删除任务栏图标的方法	271
10.9.4 实现更新任务栏图标和提示文本的方法	271
10.9.5 实现接收回调消息的代码	272
10.9.6 实现控件的终止事件代码	274
10.10 添加测试代码	274
10.11 编译 SystemTray 控件	278
10.12 小结	278
第 11 章 日志文件控件	279
11.1 LogControl 控件功能描述	279
11.2 创建 LogControl_Project 控件工程	280
11.3 添加测试工程	281
11.4 绘制 LogControl 控件	283
11.5 声明变量	285
11.6 添加控件属性	286
11.7 添加属性页	290
11.7.1 添加通用属性页	290
11.7.2 设计控件通用属性页	290
11.7.3 连接通用属性页	294
11.8 添加控件实现代码	295
11.8.1 实现 LogData 方法	295
11.8.2 实现 LimitFiles 过程	296
11.8.3 实现 GetLowestFileNum 函数	297
11.8.4 实现 SizeIsOkay 函数	300
11.8.5 实现 GetHighestFileNum 函数	301

11.8.6 实现 GetNextFileNum 函数	302
11.8.7 实现控件的 ViewLogFiles 方法	302
11.8.8 窗体 frmMain 的实现代码	303
11.9 添加测试代码	304
11.10 编译 LogControl 控件	307
11.11 小结	307
第 12 章 高级按钮控件	308
12.1 CoolCmdCtrl 控件的功能描述	308
12.2 创建 CommandCtrl_Project 控件工程	311
12.3 添加测试工程	312
12.4 绘制 CoolCmdCtrl 控件	313
12.5 变量和函数声明	315
12.6 添加控件属性	318
12.7 添加 Resize 事件代码	327
12.8 添加 AmbientChanged 事件代码	329
12.9 添加属性页	329
12.9.1 添加通用属性页	330
12.9.2 设计控件通用属性页	330
12.9.3 连接通用属性页	333
12.10 添加控件实现代码	335
12.10.1 按钮实现思路	335
12.10.2 实现控件的 DrawButton 过程	336
12.10.3 添加获得焦点和失去焦点事件代码	343
12.10.4 添加鼠标按下事件代码	345
12.10.5 添加鼠标移动事件代码	346
12.10.6 添加鼠标弹起事件代码	347
12.10.7 添加其它事件代码	347
12.11 测试按钮控件	348
12.12 编译 CoolCmdCtrl 控件	351
12.13 小结	352
第二部分 创建构件服务器	353
第 13 章 构件服务器概述	355
13.1 ActiveX 代码部件	355
13.2 构件服务器	355
13.2.1 ActiveX EXE 构件服务器	356
13.2.2 ActiveX DLL 构件服务器	356
13.2.3 ActiveX 文档服务器	356

13.3 在 Visual Basic 开发环境里开发 ActiveX 部件	357
13.3.1 创建 Visual Basic 的 ActiveX 部件工程	357
13.3.2 设置 Visual Basic 的 ActiveX 部件工程的属性	357
13.3.3 编写 Visual Basic 工程代码	360
13.3.4 运行 Visual Basic 应用程序	360
13.3.5 测试和调试 Visual Basic 应用程序	362
第 14 章 ActiveX 的进程外构件服务器	366
14.1 ActiveX 的进程外构件服务器	366
14.1.1 服务的异步通知	366
14.1.2 服务的多线程与并发	368
14.1.3 构件的类模块	370
14.2 建立进程外构件服务器	371
14.2.1 准备数据库	371
14.2.2 创建 ActiveX EXE 工程	378
14.2.3 在工程里添加数据访问对象 (DAO) 的引用	380
14.2.4 prjDBServer 工程的类模块	382
14.2.5 ActiveX 与自动化	383
14.2.6 编写工程代码	383
14.2.7 编写异步通知代码	386
14.2.8 编写多线程代码	388
14.2.9 设置工程属性	388
14.3 建立客户端应用程序	389
14.3.1 创建 prjDBClient 工程	389
14.3.2 添加 clsDBServer 引用	390
14.3.3 为工程增加一个 DAO 对象的引用	391
14.3.4 为工程增加一个模块	392
14.3.5 设计应用程序窗体	393
14.3.6 编写工程代码	397
14.4 编译并运行客户端应用程序	401
14.5 进程外构件服务器小结	402
第 15 章 ActiveX 的进程内构件服务器	403
15.1 ActiveX 的进程内构件服务器	403
15.2 建立进程内构件服务器	403
15.2.1 创建 prjSocketsDLL 工程	403
15.2.2 设计 frmWinsockFrm 窗体	406
15.2.3 为 prjSocketsDLL 编写代码	409
15.2.4 编译 prjSocketsDLL 工程到 DLL	430

15.3 建立测试应用程序	431
15.3.1 创建 prjInternetChat 工程	431
15.3.2 设计 frmInternetChat 窗体	432
15.3.3 为 frmInternetChat 窗体编写代码	442
15.4 编译并运行客户端应用程序	445
15.5 进程内构件服务器小结	447
第 16 章 ActiveX 文档	448
16.1 ActiveX 文档	448
16.1.1 创建和部署 ActiveX 文档的优点	448
16.1.2 创建 ActiveX 文档的基础	448
16.1.3 ActiveX 文档的组成部分	449
16.1.4 ActiveX 文档生命期内的关键事件	450
16.1.5 ActiveX 文档的设计考虑	451
16.2 使用 ActiveX 建立 ActiveX 文档	452
16.2.1 创建 ActiveX 文档工程	452
16.2.2 设计用户文档	453
16.2.3 添加第二个用户文档	455
16.2.4 设计第二个用户文档	456
16.2.5 添加“关于”窗体	457
16.2.6 设计用户文档的菜单	458
16.2.7 添加弹出对话框窗体	459
16.2.8 编写工程代码	461
16.3 测试和调试 ActiveX 文档	466
16.4 编译和发布 ActiveX 文档	468
16.4.1 设置 ActXDoc 工程的版本兼容性	469
16.4.2 编译 ActXDoc 工程	470
16.4.3 为当前工程建立一个 Internet 发布	470
16.5 ActiveX 文档小结	477

第一部分 ActiveX 控件编程

ActiveX 控件编程是本书的主要内容。本部分通过讲解大量的控件实例，不仅使读者可以深入掌握使用 Visual Basic 进行 ActiveX 控件编程的全过程，而且还具有很高的使用价值，读者可以根据实际应用的需要，直接使用本部分中提供的 ActiveX 控件实例。本部分各章内容的布局如下：

- 第 1 章 VB 编程入门。首先简单介绍了 Visual Basic 的一些基本编程概念，如基于 Window 的编程机制和面向对象的编程；然后介绍 VB 集成开发环境中各个因素的功能；最后，简单说明 ActiveX 部件编程的基本概念和分类。
- 第 2 章 ActiveX 控件入门。通过讲解一个简单实例的制作过程，使读者快速了解设计 ActiveX 控件的基本过程。
- 第 3 章到第 12 章，以丰富的实例演示了用 Visual Basic 进行 ActiveX 控件编程的各个方面的技巧。每章都对应一个完整的控件实例。

第 1 章 VB 编程入门

1.1 Visual Basic 简介

Microsoft 公司的 Visual Basic 6.0 给广大编程人员提供了开发 Windows 应用程序最迅速、最简捷的方法。不论是 Windows 应用程序的资深开发人员还是初学编程者, Visual Basic 都为他们提供了整套的开发工具, 以方便各层次的编程人员开发应用程序。

Visual Basic 由早期的 Basic 语言派生而来, 同时又在原有 Basic 语言的基础上进行了扩展。“Visual”指的是一种开发图形用户界面 (GUI) 的方法。它不需编写大量代码去描述界面元素的外观和位置, 而只要把预先建立的对象添加到屏幕上即可。“Basic”指的是 BASIC (Beginners All-Purpose Symbolit Instruction Code) 语言, 它是一种在计算技术发展历史上应用得最为广泛的语言。

Visual Basic 在原有 BASIC 语言的基础上进一步发展, 至今已包含了数百条语句、上百个函数及关键词, 其中很多和 Windows GUI 都有直接关系。专业人员可以用 Visual Basic 实现其它任何 Windows 编程语言能够实现的功能, 而初学者只要掌握几个关键词就可以建立实用的应用程序。从开发个人或小组使用的工具, 到大型企业应用系统及开发 Internet 上的分布式应用程序所使用的工具, 都可在 Visual Basic (以下简称 VB) 提供的工具中各取所需。

VB 所具有的数据访问特性允许其对包括 Microsoft SQL Server 和其它企业数据库在内的大部分数据库格式建立数据库和前端应用程序, 以及可调整的服务器端部件。

有了 ActiveX 技术, VB 就可使用其它应用程序提供的功能, 例如 Microsoft Word 字处理器, Microsoft Excel 电子数据表及其它 Windows 应用程序。它甚至还可直接使用 VBP 或 VBE 创建的应用程序和对象。

1.2 Visual Basic 编程基本概念

为了能掌握使用 VB 进行应用程序开发的全过程, 首先需要理解 Visual Basic 赖以创建的一些关键概念。因为 Visual Basic 是 Windows 开发语言, 所以有必要与 Windows 环境保持一定的相似性。如果不熟悉 Windows 编程, 就需要明白在 Windows 环境下编程和在其它环境下编程的一些根本性的差别。

1.2.1 Windows 工作机制

Windows 的工作机制，简单地说就是三个关键的概念：窗口、事件和消息。

可以将窗口简单地看做是带有边界的矩形区域。也许你已经了解几种不同类型的窗口，如：Windows 95 的“资源管理器”窗口、文字处理程序中的文档窗口或者弹出提示有约会信息的对话框。除了这些最普通的窗口外，实际上还有许多其它类型的窗口。如命令按钮是一个窗口，图标、文本框、选项按钮和菜单条也都是窗口。

Microsoft Windows 操作系统通过给每一个窗口指定一个唯一的标识号 `hWnd`（窗口句柄）来管理所有的窗口。操作系统连续地监视每一个窗口的活动或事件的信号。事件可以通过诸如单击鼠标或按下按键的操作而产生，也可以通过程序的控制而产生，甚至可以由另一个窗口的操作而产生。

每发生一次事件，将引发一条消息发送至操作系统，操作系统处理该消息并传播给其它窗口。然后，每一个窗口才能根据自身处理该条消息的指令而采取适当的操作（例如，当窗口解除了其它窗口的覆盖时，重显自身窗口）。

可以想象，处理各种窗口、事件和消息的所有可能的组合将有惊人的工作量。幸运的是，Visual Basic 使你摆脱了所有的低层消息处理。许多消息由 Visual Basic 自动处理了，其它的作为事件过程由编程者自行处理。这样可以快速创建强大的应用程序而毋需涉及不必要的细节。

1.2.2 面向事件的编程模式

在传统的或“过程化”的应用程序中，应用程序自身控制了执行哪一部分代码和按何种顺序执行代码。从第一行代码开始执行程序，并按应用程序中预定的路径继续往下执行，必要时调用过程。

在事件驱动的应用程序中，代码不是按照预定的路径执行，而是在响应不同的事件时执行不同的代码片段。事件可以由用户操作触发，也可以由来自操作系统或其它应用程序的消息触发，甚至可以由应用程序本身的消息触发。这些事件的顺序决定了代码执行的顺序，因此应用程序每次运行时所经过的代码的路径都是不同的。

在执行代码过程中，代码也可以触发事件。例如，在程序中改变文本框中的文本将引发文本框的 `Change` 事件。如果 `Change` 事件中包含有代码，则将导致该代码的执行。如果原来假设该事件仅能由用户的交互操作所触发，则可能会产生意料之外的结果。正因为这一原因，所以在设计应用程序时理解事件驱动模型并牢记在心是非常重要的。

1.2.3 交互式开发环境

利用传统的语言进行应用程序的开发通常可以分为三个步骤：编码、编译和测试代码，各步骤间具有明显的界限。与传统的语言不同的是使用 Visual Basic 提供的交互式方法开发应用程序，步骤与步骤之间不再有明显的界限。

在大多数编程语言里，如果编写代码时发生了错误，则在开始编译应用程序时该错误才会被编译器捕获。此时必须查找并改正该错误，然后再次进行编译，对每一个发现的错误都要重复这样的过程，Visual Basic 在编程者输入代码时便进行解释，及时捕获并突出显示大多数语法或拼写错误。看起来就像一位专家在监视代码的输入。