

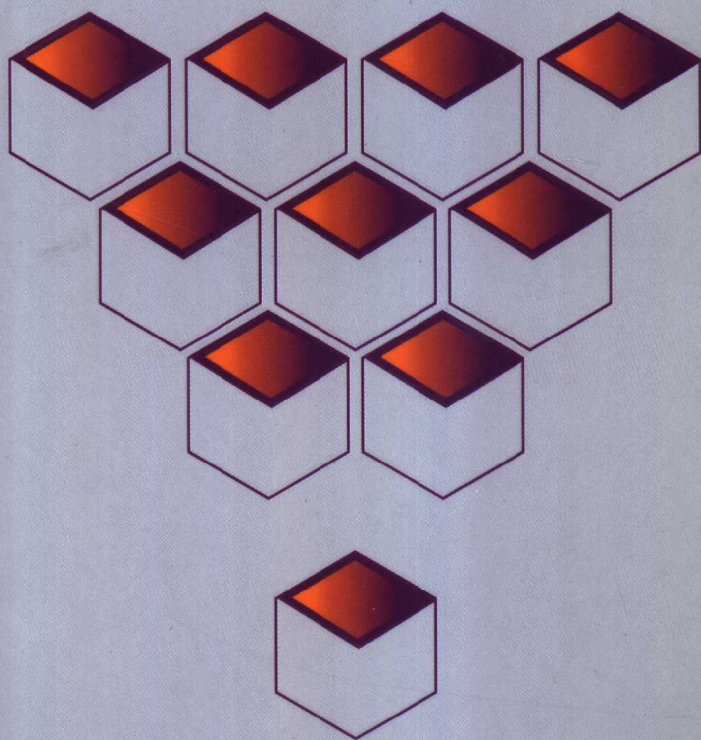
# Visual C#.NET 2003

## 开发与技巧

Visual C#.NET 2003

谢世亮 编著

Visual C#.NET 2003



清华大学出版社

# Visual C# .NET 2003 开发与技巧

谢世亮 编著

清华大学出版社  
北京

## 内 容 提 要

C#是微软公司近年来推出的一种核心编程语言,能快速、方便地开发数据库、Windows 应用程序、Web 应用程序、图形设计、图像处理以及多媒体技术等。

本书以 Visual Studio.NET 为开发平台、以实例为中心,提出软件开发解决方案。其中,重点介绍在使用 Visual C#开发 Windows 应用程序、Web 应用程序、数据库、制作报表、自定义控件和组件,以及通过 Visual C#使用 Windows 的功能、Web 服务过程中的解决方案。

在完成本书的学习和实践后,能够使读者在 Visual C#的应用程序开发技能方面得到较大提高。鉴于本书所提供的解决方案的实用性很强,可作为广大 Visual C#软件工程师的实用参考资料。

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激光防伪标签,无标签者不得销售。

### 图书在版编目(CIP)数据

Visual C# .NET 2003 开发与技巧/谢世亮编著.

—北京:清华大学出版社,2004.9

ISBN 7-302-09775-5

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

中国版本图书馆 CIP 数据核字(2004)第 108629 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:科海

文稿编辑:陈轶

封面设计:林陶

版式设计:科海

印 刷 者:北京科普瑞印刷有限责任公司

发 行 者:新华书店总店北京发行所

开 本: 异 16 印张: 17.75 字数: 432 千字

版 次: 2004 年 11 月第 1 版 2004 年 11 月第 1 次印刷

书 号: ISBN 7-302-09775-5/TP·6749

印 数: 1~4000

定 价: 28.00 元

---

本书如存在文字不清、漏印以及缺页、倒角、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: 82896445

# 前 言

虽然微软所推出的Visual Studio.NET将过去近似独立的几种开发工具的界面统一了起来，但新的开发环境好看不好用，不仅刚入行的新秀难以熟悉有关技术，就是经验充足的软件工程师对其中的很多技术也变得陌生起来，只好重新看书学习。

目前，在图书市场中介绍Visual Studio.NET的图书有很多，但是，我们可以发现在众多的图书中还没有一本关于Visual Studio.NET的技术参考大全——能够从中迅速查找到一些实际具体问题的解决方案。众所周知，几年前的美国图书市场上出现了一本《Visual Basic 5.0 Development's Workshop》，其中的内容并不是介绍基本理论，而是一些实际具体问题的解决方案。这本书让软件工程师们爱不释手，因为书中讲述的解决方案能够很好地为软件工程师解决了实际工作中碰到的棘手问题。当时编者就想寻机出一本与此类似的图书。

当微软推出Visual Studio .NET 时，编者意识到这一机会来了，因此用了近3年的时间来收集和与实践有关Visual Studio .NET软件开发技术中的最新解决方案。

不过，在刚完成本书的手稿时，现在的一部分内容是没有的，也鲜有与内容对应的图形，真有点像只丑小鸭。幸运的是，夏非彼老师和陈轶编辑帮我解决了这些问题，使得它足以成为软件工程师们的良师益友。

本书是《Visual Studio.NET开发与技巧》系列中的一本，我们经常需要翻阅的是前3章。第1章是一些Windows程序设计方面的解决方案，第2章是一些Web程序设计的解决方案，与此相关的数据访问技术则主要放在第3章中。

尽管编者为您提供这些解决方案付出了努力，但其中难免有不当之处，如您发现，请告诉编者，联系方法是：[freee@yspc.com](mailto:freee@yspc.com)。

编者

2004年9月

# 目 录

第1章 用C#开发Windows程序 .....	1
1.1 初识Windows窗体程序设计 .....	1
1.1.1 新建Windows窗体 .....	1
1.1.2 编写应用程序的代码 .....	1
1.1.3 测试应用程序 .....	1
1.2 让多个控件共享单个上下文菜单对象 .....	2
1.2.1 新建应用程序 .....	2
1.2.2 测试应用程序 .....	4
1.3 新建一个从基窗体继承的窗体 .....	5
1.3.1 新建包含基窗体的类库项目 .....	6
1.3.2 添加基窗体的继承者可修改的按钮 .....	6
1.3.3 添加不能由基窗体的继承者修改的按钮 .....	7
1.3.4 新建包含从基窗体继承的窗体的项目 .....	7
1.3.5 添加继承的窗体 .....	8
1.4 调整复杂用户界面的大小 .....	9
1.4.1 新建Outlook样式的用户界面 .....	9
1.4.2 以编程方式新建Outlook样式的用户界面 .....	10
1.5 新建没有用户界面的程序 .....	13
1.5.1 新建并配置服务 .....	13
1.5.2 将自定义事件日志功能添加到服务中 .....	15
1.5.3 定义当服务启动时发生的情况 .....	15
1.5.4 定义当服务停止时发生的情况 .....	16
1.5.5 定义服务的其他操作 .....	16
1.5.6 为服务新建安装程序 .....	16
1.5.7 生成服务项目 .....	17
1.5.8 为服务新建安装项目 .....	17
1.5.9 向安装项目添加MyNewService.exe .....	17
1.5.10 添加自定义操作 .....	18
1.5.11 安装Windows服务 .....	18

---

1.5.12	启动和停止服务.....	18
1.5.13	验证服务的事件日志输出.....	18
1.5.14	卸载服务.....	19
1.6	用对象从对话框中公开一组相关数据.....	19
1.6.1	新建通过对象公开其数据的对话框.....	19
1.6.2	显示新建的对话框并使用对象检索数据.....	21
1.7	以编程方式控制状态栏面板中数据.....	23
1.7.1	加入所需的控件.....	23
1.7.2	通过计时器更新状态栏.....	26
1.7.3	测试应用程序.....	26
1.8	切换菜单结构.....	26
1.8.1	设计一个切换菜单结构的程序.....	27
1.8.2	通过代码来创建菜单.....	30
<b>第2章</b>	<b>用C#开发Web程序.....</b>	<b>33</b>
2.1	创建一个简单的Web页.....	33
2.1.1	创建项目和窗体.....	33
2.1.2	了解Web窗体结构.....	34
2.1.3	添加控件和文本.....	35
2.1.4	创建事件处理程序.....	37
2.1.5	生成并运行Web窗体页.....	38
2.2	如何创建类.....	39
2.2.1	创建Web应用程序项目.....	39
2.2.2	创建数据访问组件.....	41
2.2.3	绑定DataGrid.....	44
2.2.4	测试项目.....	49
2.2.5	向Web窗体添加详细信息面板.....	49
2.2.6	使用Visual C#创建业务对象.....	52
2.2.7	使用业务对象.....	53
2.2.8	部署.....	53
2.3	使用Web窗体中的验证控件.....	54
2.3.1	创建基本的窗体.....	54
2.3.2	添加验证控件.....	55
2.3.3	测试验证控件.....	58
2.4	在Web窗体页中向用户传送数据.....	59

2.4.1	创建项目和窗体.....	59
2.4.2	创建和配置数据集.....	59
2.4.3	添加DataGrid以显示数据.....	60
2.4.4	填充数据集并在DataGrid控件中显示数据.....	60
2.4.5	测试Web窗体页.....	61
2.5	用网格控件读取和编辑数据.....	61
2.5.1	创建项目和窗体.....	62
2.5.2	创建和配置数据集.....	62
2.5.3	添加显示数据的DataGrid控件.....	63
2.5.4	填充数据集并在DataGrid控件中显示数据.....	63
2.5.5	添加编辑数据的功能.....	64
2.5.6	测试现有的功能.....	66
2.5.7	更新数据集和数据库.....	66
2.5.8	对页进行测试.....	69
2.6	Web窗体中用户控件的创建和使用.....	69
2.6.1	创建用户控件.....	70
2.6.2	向Web窗体页公开属性.....	77
2.6.3	初始化控件.....	78
2.6.4	使用用户控件.....	78
2.6.5	测试Web窗体页.....	81
2.7	将Web窗体页转换为可用在其他处的用户控件.....	81
2.7.1	创建Web窗体页.....	82
2.7.2	将页转换为用户控件.....	82
2.7.3	测试用户控件.....	83
2.8	Web自定义控件的创建和使用.....	84
2.8.1	创建项目.....	84
2.8.2	将控件添加到工具箱.....	85
2.8.3	个性化控件.....	85
2.8.4	创建自定义设计器.....	87
2.8.5	测试控件.....	89
<b>第3章</b>	<b>用C#开发数据库程序.....</b>	<b>90</b>
3.1	Windows窗体中的数据访问.....	90
3.1.1	新建项目.....	90
3.1.2	配置数据集.....	90

3.1.3	向窗体加入显示数据的数据网格.....	95
3.1.4	填充数据网格.....	95
3.1.5	用 Update 方法更新数据库.....	96
3.1.6	运行和测试.....	96
3.2	建一个包含多个相关表的数据集.....	97
3.2.1	新建项目并向它加入数据集.....	97
3.2.2	向数据集加入代表表的元素.....	99
3.2.3	定义表的主键.....	101
3.2.4	建立表与表之间的关系.....	101
3.3	返回用户所需的数据.....	104
3.3.1	新建项目.....	105
3.3.2	配置数据集.....	105
3.3.3	加入几个必需的控制.....	107
3.3.4	加入代码以填充数据集.....	107
3.3.5	将文本框绑定到数据集.....	107
3.3.6	加入导航控制.....	108
3.3.7	创建一个显示当前记录位置的函数.....	109
3.3.8	运行和测试.....	109
3.4	浏览相关表中的记录.....	110
3.4.1	新建项目和窗体.....	110
3.4.2	建立数据集.....	111
3.4.3	建立表与表之间的关系.....	112
3.4.4	插入几个显示数据的控制.....	113
3.4.5	填充数据集.....	114
3.4.6	运行和测试.....	114
3.5	使用不同类型的数据源来产生数据集.....	114
3.5.1	新建项目.....	115
3.5.2	新建带有预定义架构的数据集.....	115
3.5.3	为数据访问准备数据适配器和表映射.....	116
3.5.4	用一个DataGrid控件来阐释映射名.....	118
3.5.5	运行和测试.....	118
3.6	如何使Web页中的数据只读.....	118
3.6.1	新建项目.....	119
3.6.2	配置数据连接.....	119
3.6.3	使用一个数据网格来显示authors表中的信息.....	120



---

3.6.4	插入显示数据的代码.....	120
3.6.5	测试Web窗体页的只读数据访问.....	121
<b>第4章</b>	<b>关于XML Web services .....</b>	<b>122</b>
4.1	新建XML Web services.....	122
4.1.1	新建一个项目.....	122
4.1.2	添加XML Web services方法.....	123
4.1.3	XML Web services的运行与调试.....	124
4.1.4	将XML Web services部署到可供客户使用的服务器上.....	125
4.2	访问XML Web services.....	126
4.2.1	新建一个ASP.NET程序.....	126
4.2.2	添加一个Web引用以简化开发.....	127
4.2.3	访问XML Web services.....	128
4.2.4	运行与调试.....	128
4.2.5	将程序部署到Web服务器.....	129
4.3	调用XML Web services.....	130
4.3.1	调用同步Web服务.....	130
4.3.2	异步Web服务.....	132
4.4	创建访问XML Web services的设备应用程序.....	132
4.4.1	创建XML Web services项目.....	133
4.4.2	实现XML Web services.....	133
4.5	在设备项目中访问XML Web services.....	134
4.5.1	在部署时提示选择目标设备.....	134
4.5.2	创建客户端应用程序.....	135
4.5.3	测试客户端应用程序.....	137
<b>第5章</b>	<b>使用Windows的内置功能.....</b>	<b>138</b>
5.1	处理Windows中的事件日志.....	138
5.1.1	创建用户界面和控件.....	138
5.1.2	创建自定义日志和删除自定义日志.....	140
5.1.3	利用EventLog组件实例将项写入日志中.....	140
5.1.4	从自定义日志中清除一项.....	141
5.1.5	测试自定义日志和源字符串是否存在.....	141
5.1.6	删除事件源.....	141
5.1.7	显示日志中的消息.....	142
5.1.8	验证代码的执行情况.....	142

5.2 设置安装组件 .....	144
5.2.1 创建组件实例 .....	144
5.2.2 为EventLog组件添加安装程序 .....	144
5.2.3 生成安装项目 .....	145
5.2.4 将应用程序添加到解决方案 .....	145
5.2.5 添加Windows服务项目输出 .....	145
5.2.6 添加自定义操作以安装MyNewService.exe文件 .....	146
5.3 与Windows消息队列交互 .....	146
5.3.1 创建消息队列 .....	146
5.3.2 将MessageQueue组件添加到消息队列中 .....	147
5.3.3 用户界面的创建 .....	147
5.3.4 编写代码以便向队列发送消息 .....	148
5.3.5 显示队列的内容 .....	149
5.3.6 清除队列的内容 .....	149
5.3.7 测试应用程序 .....	149
5.4 监视服务器上的Windows进程并与之交互 .....	149
5.4.1 新建应用程序 .....	150
5.4.2 进程的启动 .....	150
5.4.3 为“Stop Process”按钮编写代码 .....	150
5.4.4 编写等待进程完成的代码 .....	151
5.4.5 枚举计算机上的进程 .....	152
5.5 列出网络上的资源 .....	152
5.5.1 新建用户界面 .....	152
5.5.2 设置DirectoryEntry组件的属性 .....	153
5.5.3 向TreeView控件添加顶级节点 .....	154
5.5.4 测试程序 .....	155
5.6 监视文件和目录并作出反应 .....	156
5.6.1 创建FileSystemWatcher组件 .....	157
5.6.2 确定FileSystemWatcher监视的内容 .....	157
5.6.3 定义事件的处理程序 .....	157
5.6.4 测试FileSystemWatcher组件 .....	158
5.7 设置来自性能计数器的原始值和计算值 .....	158
5.7.1 新建Windows应用程序 .....	159
5.7.2 配置PerformanceCounter组件 .....	160
5.7.3 检索和显示系统计数器的原始值 .....	160

5.7.4	创建新类别和自定义计数器.....	160
5.7.5	设置自定义计数器的原始值.....	161
5.7.6	为“将自定义计数器的值增加1”按钮编写代码.....	161
5.7.7	为“将自定义计数器的值减少1”按钮编写代码.....	161
5.7.8	为“将自定义计数器增加一个值”按钮编写代码.....	162
5.7.9	测试应用程序.....	162
5.8	用性能计数器来检索类别和计数器.....	162
5.9	新建一个应用程序.....	163
5.9.1	为“获取类别”按钮编写代码.....	163
5.9.2	为“获取计数器”按钮编写代码.....	164
5.9.3	测试程序.....	165
<b>第6章</b>	<b>自定义控件和组件.....</b>	<b>166</b>
6.1	用Visual C#来开发用户控件.....	166
6.1.1	新建一个控件库和控件.....	166
6.1.2	在用户控件中加入标签和计时器.....	167
6.1.3	在控件中添加用户能更改颜色的属性.....	168
6.1.4	测试所生成的控件.....	169
6.1.5	新建一个继承的用户控件.....	172
6.1.6	编写测试程序.....	176
6.2	用Visual C#开发简单的组件.....	177
6.2.1	新建项目.....	177
6.2.2	添加构造函数和析构函数的代码.....	178
6.2.3	在类中加入属性.....	179
6.2.4	组件的测试.....	179
6.2.5	对象生存期及相关问题.....	180
6.3	开发多线程组件.....	181
6.3.1	创建新项目.....	181
6.3.2	接收用户输入.....	185
6.3.3	测试程序.....	187
6.3.4	向组件添加多线程.....	187
6.3.5	加速更新窗体上的显示.....	190
6.3.6	优化线程.....	192
6.4	创建从CollectionBase类继承的集合.....	193
6.4.1	创建类.....	193

---

6.4.2	实现Add和Remove方法.....	195
6.4.3	实现Item属性.....	196
6.4.4	测试.....	197
6.5	通过继承得到控件.....	200
6.5.1	创建控件库和ValueButton控件.....	200
6.5.2	在继承的控件中添加属性.....	200
6.5.3	测试控件.....	201
6.6	在设备项目中创建组件.....	203
6.6.1	选择设备.....	203
6.6.2	创建新项目.....	203
6.6.3	加入构造函数和析构函数.....	204
6.6.4	将属性加到类中.....	205
6.6.5	组件测试.....	205
6.6.6	CDemo 对象的创建和释放.....	206
<b>第7章</b>	<b>制作报表.....</b>	<b>208</b>
7.1	Windows 应用程序中的报表.....	208
7.1.1	创建一个Windows应用程序.....	209
7.1.2	向Windows窗体添加一个Viewer.....	209
7.1.3	向项目添加一个Crystal报表.....	210
7.1.4	将报表对象绑定到Windows窗体Viewer.....	211
7.1.5	向Windows窗体中的源文件添加代码.....	211
7.1.6	运行Windows应用程序.....	212
7.1.7	在Windows窗体中查看Crystal报表.....	212
7.2	Web应用程序中的报表.....	212
7.2.1	创建Web应用程序.....	212
7.2.2	向应用程序添加Viewer.....	212
7.2.3	在项目中添加Crystal报表.....	213
7.2.4	绑定报表.....	213
7.2.5	处理事件.....	214
7.2.6	运行Web应用程序.....	214
7.2.7	在Web窗体中查看报表.....	214
7.2.8	Web窗体中的报表打印.....	215
7.3	将报表作为Web服务来发布.....	215
7.3.1	创建报表Web服务.....	215

7.3.2 向程序添加窗体Viewer .....	216
7.3.3 绑定报表 .....	217
7.3.4 运行应用程序 .....	221
7.4 报表中的数据访问 .....	222
7.4.1 选择报表将要引用的数据源 .....	222
7.4.2 从ADO.NET数据集制作报表 .....	224
<b>第8章 其他 .....</b>	<b>231</b>
8.1 图形技术 .....	231
8.1.1 创建一个渐变的背景 .....	231
8.1.2 GDI+ 及使用 .....	232
8.2 开发分布式应用程序 .....	235
8.2.1 建立中间层 .....	236
8.2.2 建立用户界面和运行程序 .....	240
8.3 使用动态属性来降低维护成本 .....	247
8.3.1 创建一个项目 .....	247
8.3.2 添加与Log属性值有关的代码 .....	248
8.3.3 组件的Log属性 .....	248
8.3.4 修改项目.config文件的存储值 .....	249
8.3.5 修改部署的.config文件的存储值 .....	249
8.4 Windows程序区域化 .....	249
8.4.1 用.NET来生成资源文件 .....	250
8.4.2 为查看某一种语言所写的资源而设置UI区域性 .....	251
8.4.3 向项目中添加资源文件并编辑这些文件 .....	252
8.4.4 访问8.4.3节中添加的资源 .....	252
8.5 Web程序区域化 .....	253
8.5.1 创建和编辑XML资源文件 .....	253
8.5.2 访问有关的资源 .....	255
8.5.3 通过HTML文本来显示静态资源字符串 .....	256
8.5.4 将资源字符串分配到控件的属性 .....	256
8.6 使应用程序更具亲和力 .....	257
8.6.1 计划要增加的辅助功能 .....	257
8.6.2 创建项目 .....	258
8.6.3 使窗体中的控件具有辅助功能 .....	259
8.6.4 提高可读性 .....	262

---

8.6.5	可调整菜单栏、标题栏、边框和状态栏的大小.....	264
8.6.6	在程序中使用声音.....	265
8.6.7	测试应用程序.....	265
8.7	合并模块.....	266
8.7.1	新建组件.....	266
8.7.2	新建合并模块项目.....	266
8.7.3	在“解决方案资源管理器”中添加使用合并模块的安装项目.....	267
8.7.4	安装合并模块.....	267
8.7.5	部署到网上的另一台计算机上.....	267
8.7.6	卸载合并模块.....	268

# 第 1 章 用 C#开发 Windows 程序

开发Windows应用程序和Web应用程序是程序员经常要做的工作，因此我们将在第1章和第2章中讨论一些在这两方面出现频率很高的问题，而且为了能够使读者活学活用，每个问题都将通过清晰而且非常实用的步骤来讲解。

## 1.1 初识Windows窗体程序设计

让我们从零开始，首先生成并运行一个简单的Windows窗体。

### 1.1.1 新建Windows窗体

1. 启动Visual Studio。
2. 新建名为GoodSun的Windows应用程序。
3. 将一个Button控件从“工具箱”中拖动到窗体上。
4. 单击此Button控件将其选定。在“属性”窗口中，将Text属性设置为“太阳真好”。

### 1.1.2 编写应用程序的代码

1. 双击该Button控件，为Button1的Click事件添加事件处理程序。此时将打开代码编辑器，代码插入点已位于事件处理程序中。
2. 插入下列代码：

```
// C#  
MessageBox.Show("太阳照在祖国的大地上!");
```

### 1.1.3 测试应用程序

1. 按F5键运行该应用程序。
2. 当运行应用程序时，单击“太阳真好”按钮将出现图1.1所示对话框。

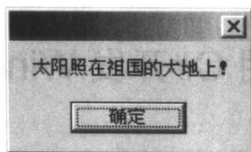


图 1.1 仅有一句代码的程序运行结果

3. 关闭此Windows窗体并返回Visual Studio。

## 1.2 让多个控件共享单个上下文菜单对象

为了在开发时减少工作量，可以让多个控件共享单个上下文菜单对象，这样，就可以减少应用程序中控件所需的上下文菜单总数。此时，我们所要做的是利用一个只包含该控件必需菜单项的“动态”上下文菜单（或快捷菜单）。

### 1.2.1 新建应用程序

现在首先新建一个Windows应用程序，其中有两个控件：**RadioButton**、**CheckBox**。在运行时，如果右击每个控件（只要它具有焦点，即被选定），将显示相应的上下文菜单。**RadioButton**控件的上下文菜单将包含两个菜单项；**CheckBox**控件的上下文菜单将包含三个菜单项。具体步骤如下：

1. 新建一个Windows应用程序。
2. 将**RadioButton**控件及**CheckBox**控件添加到窗体上。
3. 双击“工具箱”中的“**ContextMenu**”组件，将其添加到窗体中，我们将使它成为共享的上下文菜单。

Windows窗体**ContextMenu**组件用于为用户提供一个易于访问的菜单，该菜单包含与选定对象关联的常用命令。上下文菜单项常常是在应用程序其他位置出现的主菜单项的子集，一般可通过右击鼠标获得。在Windows窗体中，这些菜单与控件相关联。方法是控件的**ContextMenu**属性设置为**ContextMenu**组件。单个上下文菜单可以与多个控件关联，但每个控件只能有一个上下文菜单。

**ContextMenu**组件的主要属性是**MenuItems**属性。添加菜单项的方法有：在设计时使用菜单编辑器，或以编程方式创建**MenuItem**对象，然后将它们添加到上下文菜单的**MenuItems**集合中。鉴于上下文菜单中的菜单项通常取自其他菜单，因此向上下文菜单中添加项的最常用方法是复制这些项。

4. 在“属性”窗口中，将**CheckBox**控件和**RadioButton**控件的**ContextMenu**属性均设置



为contextMenu1。

5. 在“属性”窗口中，将CheckBox控件的ThreeState属性设置为true，该属性用于控制用户是否可以选复选框。

6. 从设计器中双击ContextMenu组件，为该组件的Popup事件新建默认的处理程序。

7. 在事件处理程序中插入执行以下任务的代码：

- 添加两个菜单项，一个表示控件的Checked状态，另一个表示Unchecked状态。
- 用If语句检验CheckBox控件是否为窗体上的SourceControl。根据检验结果，动态地添加第三个菜单项，该菜单项表示控件的Indeterminate状态。

下面的代码将显示如何使用Add方法来设置菜单项的Text属性，以及如何定义与该菜单项相关的事件处理程序。

```
// C#
protected void contextMenu1_Popup(System.Object sender, System.EventArgs e)
{
    // Clear the contents of the context menu.
    contextMenu1.MenuItems.Clear();

    // Add a menu item for the Checked state.
    contextMenu1.MenuItems.Add("Checked", new
        System.EventHandler(this.Checked_OnClick));
    // Add a menu item for the Unchecked state.
    contextMenu1.MenuItems.Add("Unchecked", new
        System.EventHandler(this.Unchecked_OnClick));

    // Test which control it is.
    // If it is the CheckBox, add a menu item for the Indeterminate state.
    if (contextMenu1.SourceControl == checkBox1)
    {
        this.contextMenu1.MenuItems.Add("Indeterminate", new
            System.EventHandler(this.Indeterminate_OnClick));
    }
}
```

8. 为MenuItem1新建一个事件处理程序。其中将检验窗体的SourceControl属性，然后根据检验结果设置RadioButton或CheckBox控件的Checked属性。添加代码如下：

```
// C#
protected void Checked_OnClick(System.Object sender, System.EventArgs e)
{
    if (contextMenu1.SourceControl == radioButton1)
        radioButton1.Checked = true;
    else if (contextMenu1.SourceControl == checkBox1)
```