

电脑编程技巧 与维护

2004年合订本 (上)

《电脑编程技巧与维护》杂志社 出版

看实例学编程

●●●●● 跟我学编程、编程语言、编程课堂、可视化专栏、数据库

听专家解疑难

●●●●● 专家论谈、编程疑难问题解答、博士信箱

保安全巧维护

●●●●● 计算机安全、计算机维护

请高手传经验

●●●●● 图形图像处理、网络技术

仅售 **58元**
(上下册+光盘)



附录：新增编程高手最新力作 Visual C++ 编程案例
免费赠送源代码光盘

TP314

1

1

电脑编程技巧与维护

Computer Programming Skills & Maintenance

2004 年合订本(上)

《电脑编程技巧与维护》杂志社 编

《电脑编程技巧与维护》杂志社

社 长: 孙茹萍

副 社 长: 毕研元 田 真

总 编: 王路敬

主 编: 张 涛

编 辑: 李相兰 程 芳 管逸群

叶 永

公关部主任: 苏加友

出版发行部: 王增顺 韦玉发

编 辑 出 版: 《电脑编程技巧与维护》杂志社

主 管 部 门: 中华人民共和国信息产业部

主 办 单 位: 中国信息产业商会

社 址: 北京市海淀区长春桥路 5 号

6 号楼 1209 室

投 稿 信 箱: gaojian@comprg. sina. net

gaojian@comprg. com. cn

编辑部信箱: gaojian@comprg. sina. net

发行部信箱: zzsfx@vip. sina. com

网 址: <http://www.comprg.com.cn>

邮 编: 100089

电 话: (010)82561037

传 真: (010)82561614

照 排: 《电脑编程技巧与维护》

杂志社电脑排版部

印 刷 厂: 北京世界知识印刷厂

邮 发 代 号: 82—715

国外发行代号: M6232

刊 号: ISSN 1006-4052
CN11-3411/TP

广告许可证: 京海工商广字 0257

定 价: 58.00 元(上、下册, 含 1CD)

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换
本社购书热线电话 (010)82561614 82561037

优 惠 读 者

直接在杂志社订阅的读者可以享受
以下优惠

★ 订阅 2005 年杂志 定价: 93.60 元

赠送《电脑编程技巧与维护》2004
年增刊——计算机系统安全与维护
编程实例精选集锦(原价: 36 元、全
书 500 页、带源代码光盘)

★ 订阅 2005 年全年杂志

优惠价: 79.40 元

8.5 折

欢迎订阅 2005 年杂志

邮发代号: 82-715

邮 编: 100089

通讯地址: 北京市海淀区长春桥路 5 号
6 号楼 1209 室

收 款 人: 《电脑编程技巧与维护》杂志社发行部

电 话: 010-82561614 010-82561037

传 真: 010-82561614

网 址: www.comprg.com.cn

前言

杂志简介

《电脑编程技巧与维护》杂志自 1994 年创刊以来,始终以“实用第一,智慧密集”为宗旨,坚持“质量第一”的原则,为广大的电脑编程爱好者、软件开发人员和专业计算机系统维护人员提供第一手的技术资料、编程技巧和维护经验,努力为中国的软件业提供切实可行的技术解决方案,在业内获得一致好评,是广大编程和维护人员的首选刊物。在栏目内容上,选题覆盖面广,涉及技术领域宽、信息量大,帮助程序员开阔视野;在技术水平上,始终把握计算机技术发展的大方向,提供先进、详尽、准确的技术指导,并在长期的工作中与国际型大公司建立了良好的合作关系,为读者提供全球最新、最全的实用信息;在实用性上,稿源来自专业开发和维护人员的实践经验,以及普通书籍难以获得的编程心得体会与技巧。

内容提要

《电脑编程技巧与维护》2004 年合订本保留杂志的原有风格,但不是 12 期内容的简单相加,而是按杂志的栏目重新编排。包括:跟我学编程、编程课堂、编程语言、专家论坛、可视化专栏、数据库、网络技术、图形图像处理、计算机维护、计算机安全、编程疑难问题解答、博士信箱等 12 个栏目;另外,以附录的形式给出从未发表过的 VC++ 最新应用技巧和解决方案 11 篇。本套合订本分为上、下两册,上册覆盖前 4 个栏目,共文章 133 篇;下册覆盖后 8 个栏目和附录,共文章 118 篇;并随书附赠一张源代码光盘。

◆跟我学编程

本栏目是为广大初学电脑编程的读者提供一个相互学习、相互交流编程心得体会的园地,主要讲解 VC/C++、VB、Delphi、PB 等编程语言的基本编程知识;帮助读者学习使用这些编程语言,掌握其基本编程思路和方法;通过具体实例提供编程步骤;提供需要掌握的编程中的实战经验和小技巧等。

◆编程课堂

本栏目以专题讲座的形式,针对 Java、C++ Builder、VC++、C# 等内容进行系统全面的讲解,让编程开发人员“学之有道”,使其知识水平上升一个高度。

◆编程语言

本栏目主要介绍当今比较流行的编程语言(如 VC/C++、VB、Delphi、PB、Java、C# 等)的编程技术、技巧以及它们所提供的函数、类等的灵活应用技术,旨在通过实用程序来讲解所应用的方法、技术与技巧。

◆专家论坛

本栏目提供一些在软件开发及应用的某一领域具有权威性的文章,编程人员可以在新的技术开发和应用领域有一定提高,同时也为编程人员相互学习和交流提供了一个论坛空间。

◆ 可视化专栏

本栏目主要介绍比较流行的可视化编程语言的发展动态和编程技巧,如 VB、VC/C++、C# 等应用开发的经验和体会。通过提供一系列用 VB、VC++、C# 等可视化语言开发的实例,介绍其在系统开发、各种控件的制作等方面的应用。读者通过该栏目的学习将真正成为在可视化编程方面的高手。

◆ 数据库

本栏目主要介绍 Visual FoxPro、SQL Server、Oracle、DB2 等数据库的新功能、新用法,以及利用这些数据库进行开发的经验、编程技术、技巧和方法。

◆ 网络技术

本栏目主要介绍网络的应用技术,包括网络操作系统和网络环境下的应用开发技术,Web Services 技术、Web 库的实现及管理、网站的建设和安全管理方面的技巧与方法,以及最新网站开发方面的实用技术。

◆ 图形图像处理

本栏目主要介绍图形图像软件的编制技术、技巧和编程的具体方法,包括软件界面的开发、各种图像格式的介绍及应用,如用 VC++ 等语言实现图像的浏览、位图的显示特技等。

◆ 计算机维护

本栏目涉及软件功能的扩充及软件故障的合理恢复方法,并介绍计算机硬件维护与维修的经验和体会,提供计算机用户修复软、硬件系统故障的实用方法。

◆ 计算机安全

本栏目主要介绍加密技术、反跟踪技术、网络安全及反病毒技术,例如具体病毒的检测和消除方法、如何有效地安装防火墙预防黑客。

◆ 编程疑难问题解答

本栏目主要对编程开发人员的实际技术难题进行答疑解惑,对 VC/C++、VB、Delphi、PB、Java 等编程应用中遇到的实际问题给出了准确、精彩的回答。

◆ 博士信箱

本栏目主要刊登计算机专家解答计算机应用方面的技术问题的文章,如电脑基本工具软件实用技巧与常见问题处理等系列文章。

◆ 附录

在囊括 2004 年杂志内容的基础上,本套合订本增加了一些杂志没有的内容,以附录的形式给出 VC++ 最新应用技巧和解决方案 11 篇。

光盘内容

1. 《〈电脑编程技巧与维护〉》2004 年文章的程序源代码,所有源程序均经调试通过。
2. 附录中,VC++ 最新应用技巧和解决方案的源代码,所有源代码均经调试通过。

读者对象

本书为广大的电脑编程爱好者、软件开发人员和专业计算机系统维护人员提供第一手的技术资料、编程技巧和维护经验,是广大计算机爱好者、编程和维护人员的首选读物和必备参考资料。只要您有志于深入研究计算机的高级技术,相信本书是您正确的选择。

《电脑编程技巧与维护》杂志社

目 录

前 言

1. 跟我学编程

1.1 对常用 Windows 控件的使用	1
1.2 工具栏和菜单之间的交互	5
1.3 程序的安装和部署	7
1.4 VC 编程小技巧四则	13
1.5 利用 VC + + 6.0 对数据库的动态刷新和多表切换	14
1.6 VC 中动态打开/显示数据库的实现	16
1.7 VC 编程小技巧五则	18
1.8 VC 编程小技巧集锦	19
1.9 Delphi 7 开发入门	22
1.10 Delphi 的 Object Pascal 编程实例	25
1.11 Delphi 数据库的 SQL 编程实例	28
1.12 Delphi 面向对象的编程方法	30
1.13 简单数据库应用(上)	33
1.14 简单数据库应用(下)	37

2. 编程课堂

2.1 面向对象编程语言关键特性的剖析——继承、运行时类型识别和内联函数	40
2.2 Java 应用技巧讲座(一)	48
2.3 Java 应用技巧讲座(二)——Java 网络寻址	51
2.4 Java 应用技巧讲座(三)——UDP 数据报在单播模式中的应用	53
2.5 Java 应用技巧讲座(四)——UDP 数据报在组播模式中的应用	56
2.6 在 C + + Builder 工程里使用 Visual C + + DLL——第 1 部分:C 函数	59
2.7 在 C + + Builder 工程里使用 Visual C + + DLL——第 2 部分:C + + 类	63
2.8 Visual C#图形图像编程之一——基础知识及基本绘图方法	67

2.9 Visual C#图形图像编程之二——捕获屏幕	72
2.10 Visual C#图形图像编程之三——处理图片一	78
2.11 Visual C#图形图像编程之四——处理图片二	83

3. 编程语言

3.1 VC 编制串行通信库	87
3.2 VC 中定制 DIB 类	89
3.3 VC 编程实现组合框选项的属性页	92
3.4 在 VC 中实现自定义事件的编程	95
3.5 VC++ 下对匿名管道的编程实现	98
3.6 VC++ 编程实现邮箱数据传输	100
3.7 用 VC++ 实现搜索	103
3.8 用 VC++ 实现通用的报表控件	106
3.9 VC++ 6.0 中实现矢量图的分层绘制、打印	110
3.10 VC++ 6.0 多线程技术在端口扫描程序中的应用	116
3.11 Visual C++ 6.0 中实现三叉切分窗口与多视图	120
3.12 Visual C++ 6.0 中实现运行程序图标在任务栏中的变换及去掉标题图标	123
3.13 用 VC++ 6 实现 FlashGet 的最小化和关闭的效果	124
3.14 Visual C++ 中 MSFlexGrid 控件的功能扩展及其应用	126
3.15 用 VC++ 6.0 编写一个完整的 Windows 屏幕保护程序	130
3.16 Windows XP 中利用 Visual C++ 编程创建快捷方式	137
3.17 Windows XP 滑动面板	140
3.18 Windows Shell 扩展编程基础	142
3.19 对话框应用程序中实现剪贴及打印功能	146
3.20 在 GDI+ 中使用双缓存	148
3.21 在 MFC 应用程序中浏览 PDF、Word 文档文件	150
3.22 Exif 信息查看器	152
3.23 属性页扩展	155
3.24 卡通渲染	157
3.25 编程实现 Wizard 程序	158
3.26 VB 编程实现实现播放多媒体文件	161
3.27 用 VB 实现 GPS 接收机与计算机的串口通信	163
3.28 用 VB 语言设计数字滤波器	165
3.29 在 VB 中调用 API 函数操作指针变量	167
3.30 基于 VB6 的组合查询系统的开发	170
3.31 基于 COM 组件的 VB 与 MATLAB 接口编程	172
3.32 浅谈 VB 在编程中如何实现添加动画小精灵角色	175
3.33 通过 VB 向 Excel 传输数据的方法	180
3.34 具有滚动功能的 ActiveX 按钮控件	183



3.35	编程改变分割窗体尺寸	186
3.36	Delphi 数据库多栏动态组合显示的设计与实现	188
3.37	Delphi 实现试题库出卷系统 Word 文档考卷的输出	191
3.38	Delphi 中带有鼠标轮事件的 DBGrid 控件的设计与应用	195
3.39	Delphi 中利用 Tbitmap 实现窗体图形动态打印	198
3.40	Delphi 中的控件组生成及事件响应	200
3.41	用 Delphi 建立多线程 COM 服务器	202
3.42	用 Delphi 和 VB 实现浏览器中超长文件的上传	207
3.43	用 Delphi 制作“资源管理器”结构的数据查询窗口,并将当前窗口中的数据转出到 Excel	209
3.44	在 Delphi 中编程控制系统 IP 配置	212
3.45	利用 Delphi 编写远程控制程序	215
3.46	利用 Delphi 的流对象对多媒体文件进行操作	222
3.47	利用 WebBrowser Control 实现程序界面与实现的分离	226
3.48	利用“拦截者”技术实现 MIDAS 数据安全	229
3.49	向数据库中导入数据的几种方法	233
3.50	基于 Java 语言的多线程同步机制	240
3.51	用 Java 实现 P2P 网络通信	244
3.52	基于 Java 语言提取网站内部 URL 的算法	247
3.53	用 Java 语言实现 RGB 与 CMYK 色彩空间的转换	249
3.54	在 JBoss 中配置 JAAS 实现登录安全控制	252
3.55	C++ Builder 中有关 TDBGrid 的几个技巧	255
3.56	C++ Builder、VBA 与 Excel 有机组合——生成报表自动化系统	258
3.57	C++ 中用指针调用类成员函数的问题及解决的方法	265
3.58	通过 FTP 实现数据转储	268
3.59	在 VFP 中用两种不同的形式实现动态表的浏览	271
3.60	VFP 数据表文件的程序自动修复	274
3.61	Visual FoxPro 动态报表设计与实现	276
3.62	对数据窗口的 dwItemStatus 研究及应用	279
3.63	PowerBuilder TreeView 控件实现装置名称的动态树型管理	282
3.64	C#中消息的处理	285
3.65	通过 C#实现集合类纵览 .NET Collections 及相关技术	287
3.66	ASP.NET 中用 OWC10 及 XML 技术实现各种动态统计图的方法研究	293
3.67	设计 C#定制控件实现贝塞尔曲线	296
3.68	基于 C#的 GML 文档解析	300
3.69	Linux 下文件系统的编程	306
3.70	RedHat Linux 汉化	310
3.71	脚本组件的设计与使用	312
3.72	内容管理系统中基于 XML 页面发布的实现与应用	315
3.73	Asp 上传与下载文件的技巧	320

3.74 用 XML + XSL 实现网页下拉菜单	324
3.75 一款在线 Word 编辑器及其痕迹跟踪的实现	327
3.76 关于 Web 应用程序的范围与 session 变量丢失问题的深入研究	334

4 可视化专栏

4.1 VC 实现窗口的分割	336
4.2 VC 编程实现显示和隐藏工具条	341
4.3 VC 中实现对多画面窗口的控制	345
4.4 ActiveMovie 控件在 VC + + 多媒体程序开发中的应用	347
4.5 VC + + 环境下多虚拟桌面程序的实现	351
4.6 VC + + 播放 CD 中第一首曲子的实现过程	355
4.7 VC + + 制作个性对话框	358
4.8 VC + + 实现动态创建对话框	363
4.9 VC + + 与 Matlab 混合编程实现卫星遥感影像的三维显示	365
4.10 VC + + 环境下控制线程的运行技术	368
4.11 VC + + 6.0 下利用互斥量同步线程来实现文件读取进度条	371
4.12 VC + + 6.0 下的图像处理及在多文档和无模态对话框中的实时显示	375
4.13 在 VC + + 6.0 下利用共享内存、消息实现内部进程通讯	377
4.14 在 VC 中利用子类化技术扩展通用控件的功能	381
4.15 用 VC 实现 USB 接口智能卡读写器的编程	383
4.16 用 VC + + 5.0/6.0 实现“更改图标”对话框	385
4.17 用 VC + + 实现全屏幕显示	389
4.18 用 VC + + 处理灰度位图	391
4.19 用 VC + + 实现二维等值线的 COM 组件	395
4.20 基于 VC + + 和 L 系统的自然景物模拟	398
4.21 Visual C + + 中双缓存滚动视图类开发及使用	401
4.22 Visual C + + 与 MATLAB 的数字基带传输系统仿真的实现	404
4.23 利用控件聚合技术在 VC 中实现 MSFlexGrid 的编辑功能	407
4.24 基于 IE 接口的通用进度对话框设计	410
4.25 在 VB 中使用 DirectX 技术实现 MIDI 的播放	415
4.26 Visual C# 文件编程之分割合并文件	419
4.27 Visual C# 套接字编程	427
4.28 Visual C# 创建 Tracert 命令	434
4.29 C# 开发 Asp.net 验证控件	442
4.30 三维地形实时动态显示的核心技术研究	448
4.31 动态链接库输出函数的动态加载	451
4.32 编程实现从应用程序中直接关闭计算机的方法	454



1. 跟我学编程

1.1 对常用 Windows 控件的使用

郎 锐

一、按钮类控件

按钮类控件是窗口类名被系统预定义为 `BUTTON` 的一类控件, 该类控件具有十余种不同的窗口风格, 包含了普通的下压式按钮、单选按钮、复选按钮和分组框等多种常用的按钮形式。具体情况列表如下:

按钮风格	说 明
<code>BS_AUTOCHECKBOX</code>	同复选按钮类似, 点击一下选中, 再次点击取消。
<code>BS_AUTORADIOBUTTON</code>	同单选按钮类似, 点击后选中标志将从同组的其他单选按钮处移到当前选项。
<code>BS_AUTO3STATE</code>	同三态复选按钮类似, 只是在用户点击后改变状态。
<code>BS_CHECKBOX</code>	复选按钮。
<code>BS_DEFPUSHBUTTON</code>	下压按钮, 具有较黑的边框。
<code>BS_GROUPBOX</code>	分组框。
<code>BS_LEFTTEXT</code>	同单选按钮或复选按钮配合使用, 标题将显示在左侧。
<code>BS_OWNERDRAW</code>	可创建一个拥有者自绘按钮。
<code>BS_PUSHBUTTON</code>	普通下压按钮。
<code>BS_RADIOBUTTON</code>	单选按钮。
<code>BS_3STATE</code>	三态复选按钮。

以上按钮风格通常需要同窗口风格共同使用, 由 `CreateWindow()` 函数或 MFC 的 `CButton` 类成员函数 `Create()` 完成对按钮的创建:

```
// 按钮类窗口风格
```

```
DWORD STYLE[9] = {BS_AUTOCHECKBOX,
                  BS_AUTORADIOBUTTON,
                  BS_AUTO3STATE,
                  BS_CHECKBOX,
                  BS_DEFPUSHBUTTON,
                  BS_GROUPBOX,
                  BS_PUSHBUTTON,
```

```
BS_RADIOBUTTON,
BS_3STATE};
```

```
// 动态创建按钮
```

```
for (int i = 0; i < 9; i++)
```

```
{
```

```
m_ctrButton[i].Create( " BUTTON", WS_VISIBLE |
WS_CHILD | WS_BORDER | STYLE[i], CRect(10 + 110 *
i, 10, 100 + 110 * i, 40), this, IDC_BUTTON1 + i);
```

```
}
```

如果需要处理按钮发送给它父窗口的消息, 可以在进行消息响应的类实现中添加一个如下形式的消息映射入口和相应的消息处理函数:

```
ON_Notification(id, memberFxn)
```

其中, `id` 为发送通知消息的按钮 ID 号, `memberFxn` 为消息处理函数。如果按钮是以 `new` 操作符的方式在堆 (heap) 内创建一个 `CButton` 对象, 那么就必须确保在关闭窗口前能调用 `delete` 销毁该对象。如果 `CButton` 对象是在栈上创建的不必显式销毁对象了, 应用程序在退出时会自动销毁。

二、编辑类控件

以窗口类名 “`EDIT`” 创建的编辑类控件是一个可以用来接受用户键盘字符输入的矩形区域, 可以在其内进行编辑操作。该控件是程序接受用户字符输入的一种主要手段, 输入的内容存放在其父窗口容量有限的 (32KB) 局部堆中。

MFC 的 `CEdit` 类提供了有关编辑类控件的功能函数。编辑控件既可以在对话框模板上创建, 也可以通过代码来直接创建, 这两种方式均要通过 `CEdit` 的构造函数来构造一个 `CEdit` 对象。`CEdit` 类从 `CWnd` 继承了一些重要的函数, 比如可以通过使用 `CWnd` 类成员函数 `SetWindowText()` 和 `GetWindowText()` 来设定和获取一个编辑控件中的文本。同按钮类控件类似, 如果要处理由编辑控件发送给它父窗口的通知消息, 需要在父窗口类中为每一个待处理消息增添消息映射入口和消息响应函数。

在调用 `Create()` 函数创建编辑控件时, Windows 系统将发出 `WM_NCCREATE`、`WM_NCCALCSIZE`、`WM_CREATE` 和 `WM_GET -`

MINMAXINFO 等消息给编辑控件。这些消息缺省地分别由 OnNcCreate()、OnNcCalcSize()、OnCreate() 和 OnGetMinMaxInfo() 等 CWnd 类成员函数进行处理, 可以根据实际需要对其进行重载使用。作为一种特殊的窗口, 编辑控件除了需要指定普通窗口风格外, 还可以通过选用不同的控件相应的功能效果, 可选用的编辑风格列表如下:

编辑风格	说 明
ES_AUTOHSCROLL	当在行尾添加一个字符后自动向右滚动 10 个字符。
ES_AUTOVSCROLL	当输入回车后自动上滚一行。
ES_CENTER	字符居中显示。
ES_LEFT	字符左对齐。
ES_LOWERCASE	统一转化为小写字母。
ES_MULTILINE	允许多行显示。
ES_NOHIDESEL	当编辑失去焦点时隐藏对字符的选定, 重新获得焦点后以反色显示选中内容。
ES_OEMCONVERT	将 ANSI 字符转化为 OEM 字符。
ES_PASSWORD	以星号显示字符, 多用于回显密码。
ES_RIGHT	字符右对齐。
ES_UPPERCASE	统一转化为大写字母。
ES_READONLY	设置字符为只读。
ES_WANTRETURN	接受回车键输入。

下面给出创建用编辑控件的示例代码, 该示例首先通过 Create() 函数创建一个控件对象, 并通过 SetWindowText() 函数为其设置字符。通过添加对控件通知消息的响应代码可以检测文本是否发生了改变。程序主要实现清单如下:

```
// 控件的创建部分:
// 创建编辑控件
m_ctrEdit.Create(WS_VISIBLE | WS_CHILD |
WS_BORDER | ES_WANTRETURN | ES_MULTILINE | ES_
AUTOHSCROLL | ES_AUTOVSCROLL, CRect(10, 50, 250,
150), this, ID_EDIT1);
// 为控件设置字符
m_ctrEdit.SetWindowText("Hello World!");
.....
// 添件对控件通知消息的响应
//{{AFX_MSG(CSample02View)
afx_msg void OnEnChange();
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
.....
BEGIN_MESSAGE_MAP(CSample02View, CView)
//{{AFX_MSG_MAP(CSample02View)
ON_EN_CHANGE(ID_EDIT1, OnEnChange)
//}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

```
.....
void CSample02View::OnEnChange()
{
    // 通知消息响应代码
    AfxMessageBox("内容已改变!");
}
```

三、列表框与组合框

列表框是窗口类名为“ListBox”的预定义子窗口, 在窗口矩形内包含有一些可以滚动显示的栏状字符串列表。标准的列表框只能允许选中一个条目, 选中的条目将以系统颜色 COLOR_HIGHLIGHT 高亮显示。Windows 还提供了多种不同风格的标准列表框, 其中包括多选列表框、多列显示的列表框和可以显示图像的拥有者画(Owner-draw)列表框等。另一种与列表框相关的控件是组合框, 该控件预定义类名为“COMBOBOX”, 实际是一个编辑控件同一个彼此相关的列表框控件的组合。用户既可以在组合框的编辑栏上直接输入、编辑文字, 也可以从下拉列表中显示的可能选择中进行选取。

列表框风格	说 明
LBS_STANDARD	创建一个具有边界和垂直滚动条、当选择发生变化或条目被双击时能够通知父窗口的标准列表框。所有条目按字母排序。
LBS_SORT	按字母排序。
LBS_NOSEL	条目可视但不可选。
LBS_NOTIFY	当用户选择或双击一个串时, 发出消息通知父窗口。
LBS_DISABLENOSCROLL	在条目不多时依然显示并不起作用的滚动条。
LBS_MULTIPLESEL	允许条目多选。
LBS_EXTENDEDSEL	可用 SHIFT 和鼠标或指定键组合来选择多个条目。
LBS_MULTICOLUMN	允许多列显示。
LBS_OWNERDRAWVARIABLE	创建一个拥有者画列表框, 条目高度可以不同。
LBS_OWNERDRAWFIXED	创建一个具有相同条目高度的拥有者画列表框。
LBS_USETABSTOPS	允许使用 TAB 制表符。
LBS_NOREDRAW	当条目被增删后不自动更新列表显示。
LBS_HASSTRINGS	记忆了添加到列表中的字符串。
LBS_WANTKEYBOARDINPUT	当有键按下时向父窗口发送 WM_VKEYTOITEM 或 WM_CHARTOITEM 消息。
LBS_NOINTEGRALHEIGHT	按程序设定尺寸创建列表框。

MFC 的 `CListBox` 类封装了列表框控件, 由成员函数 `Create()` 完成对列表框的创建, 在创建的同时指定了控件的窗口风格。当列表框中的条目被选中或被鼠标双击后将向父窗口发送 `WM_COMMAND` 消息。如果列表框采用了 `LBS_NOTIFY` 风格, 父窗口就可以通过 `ON_LBN` 消息映射宏直接将通知消息同消息处理函数建立起映射关系。列表框提供了十余种窗口风格, 现将列表框风格及说明如上页表:

缺省的列表框在每添加或删除一个条目后都会自动重绘, 如果在列表框中已经有了几百条甚至上千条条目, 将会因为重绘而引起比较严重的闪烁。可以通过使用 `LBS_NOREDRA` 风格来禁止列表框的自动重绘。在需要更新显示时强制重绘列表框窗口即可。如果在创建时未使用 `LBS_NOREDRA` 风格, 可以在增删条目前向列表框发送 `WM_SETREDRAW` 消息, 指定其不重绘, 增添完毕后再次发送 `WM_SETREDRAW` 消息重新启用自动重绘风格。示例过程如下:

```
CListBox m_ctrlListBox;
// 禁止自动重绘
m_ctrlListBox.SendMessage(WM_SETREDRAW, FALSE, 0);
// 进行条目增删操作
.....
// 允许自动重绘
m_ctrlListBox.SendMessage(WM_SETREDRAW, TRUE, 0);
```

列表框创建之初不含任何条目, 通过 `CListBox` 成员函数 `AddString()` 和 `InsertString()` 向列表框增添或插入条目。如果列表框具有 `LBS_SORT` 风格, 那么新添加字符串的位置是不固定的, 要根据字符串的字母进行排序; 如果不具有该风格, 新字符串将添加到列表框的末尾。

如果有必要, 可以使用 `SetItemDataPtr()` 或 `SetItemData()` 将一个 32 位的指针 (或一个 `DWORD` 的值) 同列表框中的一个条目联系起来, 并且在设置后可以通过调用 `GetItemDataPtr()` 或 `GetItemData()` 而获取。这样做的目的是可以将列表框中的条目同外部数据建立联系。例如: 可以用这种方式非常方便地将一个包含有地址、电话号码和 E-mail 地址等信息的数据结构同列举在列表框中的持有人建立起关联。当从列表框中选中某个人时, 可以同时得到有关该人的通讯信息。

当操作列表框时, 将会通过 `WM_COMMAND` 消息发送通知给父窗口, 消息参数 `lParam` 的高字节包含了通知码标识符。在 MFC 应用程序中, 列表框的通知消息通过 `ON_LBN` 消息映射宏而映射到类成员函数。下表给出了列表框的几个通知消息以及相应的 `ON_LBN` 宏。其中, `LBN_DBLCLK`, `LBN_SELCHANGE` 和 `LBN_SELCANCEL` 通知消息只有在列表

框使用了 `LBS_NOTIFY` 或 `LBS_STANDARD` 风格时才会被发出, 其他通知消息则无此限制。

其中, 最经常使用的两个通知消息是 `LBN_DBLCLK` 和 `LBN_SELCHANGE`。对于不可复选的列表框可以通过 `GetCurSel()` 来获取当前双击的是列表框条目的索引值; 对于允许多选的列表框则需要用 `GetCaretIndex()` 来代替 `GetCurSel()`。下面通过一段示例代码对列表控件的使用做一个直观的演示:

通知码标识符	ON_LBN 宏	值	含 义
<code>LBN_SETFOCUS</code>	<code>ON_LBN_SETFOCUS</code>	4	列表框接收到输入焦点
<code>LBN_KILLFOCUS</code>	<code>ON_LBN_KILLFOCUS</code>	5	列表框接失去输入焦点
<code>LBN_ERRSPACE</code>	<code>ON_LBN_ERRSPACE</code>	-2	列表框存储溢出
<code>LBN_DBLCLK</code>	<code>ON_LBN_DBLCLK</code>	2	双击条目
<code>LBN_SELCHANGE</code>	<code>ON_LBN_SELCHANGE</code>	1	改变选择
<code>LBN_SELCANCEL</code>	<code>ON_LBN_SELCANCEL</code>	3	取消选择

```
// 创建并初始化列表框
// 创建列表框
m_ctrlListBox.Create(WS_VISIBLE |
WS_CHILD | WS_BORDER | LBS_STANDARD, CRect(270,
50, 370, 150), this, IDC_LIST1);
// 添加条目
CString Item[9] = {"Item1", "Item2", "Item3", "
Item4", "Item5", "Item6", "Item7", "Item8", "Item9"};
for (int i = 0; i < 9; i++)
    m_ctrlListBox.AddString(Item[i]);
// 选中第 4 个条目
m_ctrlListBox.SetCurSel(3);
.....
// 父窗口对通知消息的处理函数的声明 (在头文件中)
//{{AFX_MSG(CSample02View)
afx_msg void OnLbnDbClk();
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
.....
// 父窗口对通知消息的映射入口 (在实现文件中)
BEGIN_MESSAGE_MAP(CSample02View, CView)
//{{AFX_MSG_MAP(CSample02View)
ON_LBN_DBLCLK(IDC_LIST1, OnLbnDbClk)
//}}AFX_MSG_MAP
// Standard printing commands
END_MESSAGE_MAP()
.....
// 父窗口对 LBN_DBLCLK 通知消息的处理
void CSample02View::OnLbnDbClk()
{
    // 得到当前选中条目的索引
    int Index = m_ctrlListBox.GetCurSel();
    // 得到此条目的内容
    char Text[20];
```

```

m_ctrListBox.GetText(Index, Text);
// 以信息框报告得到的内容
AfxMessageBox(CString(Text));
}

```

虽然组合框实际是列表框和编辑框的组合,但在使用的表现使得组合框同其他控件一样当作一个独立的控件去使用。MFC 的 CComboBox 类提供了对组合框的功能支持。在使用 Create() 函数创建组合框时可以同时指定组合框的风格(参见下表)。

窗口风格	说 明
CBS_AUTOHSCROLL	当在行尾输入字符时自动将编辑框中的文字向右滚动。
CBS_DROPDOWN	同 CBS_SIMPLE 风格类似,只是只有在用户点击下拉图标时才会显示出下拉列表。
CBS_DROPDOWNLIST	同 CBS_DROPDOWN 类似,只是显示当前选项的编辑框为一静态框所代替。
CBS_HASSTRINGS	创建一个包含了由字符串组成的项目的拥有者画组合框。
CBS_OEMCONVERT	将组合框中的 ANSI 字符串转化为 OEM 字符。
CBS_OWNERDRAWFIXED	由下拉列表框的拥有者负责对内容的绘制;列表框中各项目高度相同。
CBS_OWNERDRAWVARIABLE	由下拉列表框的拥有者负责对内容的绘制;列表框中各项目高度可以不同。
CBS_SIMPLE	下拉列表始终显示。
CBS_SORT	自动对下拉列表中的项目进行排序。
CBS_DISABLENOSCROLL	当下拉列表显示内容过少时显示垂直滚动条。
CBS_NOINTEGRALHEIGHT	在创建控件时以指定的大小来精确设定组合框尺寸。

对组合框进行操作也会向父窗口发送通知消息,处理过程同前面几种控件大同小异,是通过 ON_CBN 消息映射宏完成对通知消息的映射的。右表就给出这些 ON_CBN 宏的详细说明:

最后给出一段有关组合框的示例代码,从代码实现不难看出组合框控件同前面给出的列表控件在编程实现上的相似性。

```

// 组合框的创建
// 创建列表控件
m_ctrComboBox.Create(WS_VISIBLE | WS_CHILD |
WS_BORDER | CBS_DROPDOWN, CRect(400, 50, 470,

```

ON_CBN 宏	对应事件
ON_CBN_CLOSEUP	关闭下拉列表。
ON_CBN_DBLCLK	双击下拉列表中的项目。
ON_CBN_DROPDOWN	下拉显示列表框。
ON_CBN_EDITCHANGE	编辑框中文本内容被改动。
ON_CBN_EDITUPDATE	编辑框内容更新显示。
ON_CBN_ERRSPACE	组合框不能为某个特殊请求分配足够的内存。
ON_CBN_SELENDCANCEL	用户的选择被取消。
ON_CBN_SELENDOK	用户选择了一个项目并且通过回车键或按下鼠标而隐藏组合框的下拉列表。
ON_CBN_KILLFOCUS	组合框失去焦点。
ON_CBN_SELCHANGE	选择发生变化。
ON_CBN_SETFOCUS	组合框获得输入焦点。

```

150), this, IDC_COMBO1);
// 添加条目
CString Item[9] = {"Item1", "Item2", "Item3", "Item4",
"Item5", "Item6", "Item7", "Item8", "Item9"};
for (int i = 0; i < 9; i++)
    m_ctrComboBox.AddString(Item[i]);
// 选中第 4 个条目
m_ctrComboBox.SetCurSel(3);
.....
// 通知消息响应函数的声明(在头文件中)
//{{AFX_MSG(CSample02View)
afx_msg void OnCbnSelChange();
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
.....
// 对通知消息的响应处理(在实现文件中)
BEGIN_MESSAGE_MAP(CSample02View, CView)
//{{AFX_MSG_MAP(CSample02View)
ON_CBN_SELCHANGE(IDC_COMBO1, OnCbnSelChange)
//}}AFX_MSG_MAP
END_MESSAGE_MAP()
.....
void CSample02View::OnCbnSelChange()
{
    // 得到当前选中条目的索引
    int Index = m_ctrComboBox.GetCurSel();
    // 得到此条目的内容
    char Text[20];
    m_ctrComboBox.GetLBText(Index, Text);
    // 以信息框报告得到的内容
    AfxMessageBox(CString(Text));
}

```

1.2 工具栏和菜单之间的交互

丁有和

工具栏上通常有一系列的工具按钮,所有的按钮图像都具有相同的尺寸,一般是 15 像素高,16 像素宽,借助它们可以提高用户的工作效率,并且将常用的菜单命令也放在工具栏上,它们实际是命令不同的用户方式。

1. 添加并设计工具栏

(1) 将解决方案资源管理器窗口切换到“资源视图”,展开后右击 Toolbar,在弹出的快捷菜单中单击“插入 Toolbar”。这样,一个工具栏资源就添加到项目中,默认的标识为 IDR_TOOLBAR1。

(2) 添加并设计两个工具按钮,结果如图 1 所示。

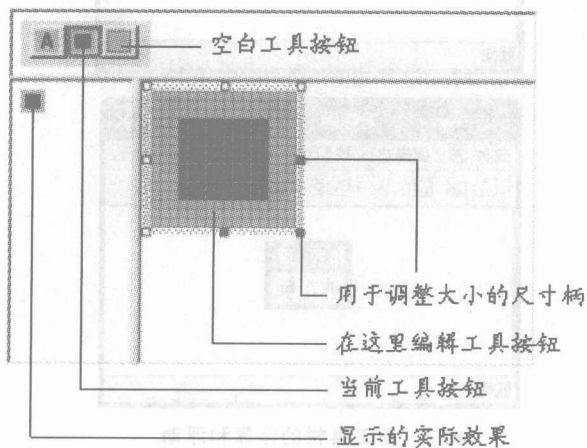


图 1 工具按钮的设计

编辑操作与 Windows 的画图相类似,这里仅列出操作的一些技巧:

① 单击空白按钮后就可以编辑其图像,同时系统在随后的位置自动添加一个空白按钮。

② 用鼠标可以将一个按钮拖放到工具栏上的其他位置上。若拖动时按下 Ctrl 键,则复制一个工具按钮。若将工具按钮拖出工具栏,则该工具按钮被删除。

③ 按 Delete 键可以将当前工具按钮的图像用背景色填充。

④ 在工具按钮之间添加间隔时,可按不同情况来操作。若工具按钮前没有任何间隔,拖动该工具按钮向右直到它覆盖相邻工具按钮的一半以上后,释放鼠标键,则此工具按钮前出现间

隔。若工具按钮前面有间隔而后面没有间隔,拖动该工具按钮向左直到它的左边界接触到它前面的工具按钮为止,释放鼠标键,则此工具按钮后面将出现间隔。

⑤ 若工具按钮前后均有间隔,拖动该工具按钮向右直到它接触相邻工具按钮,则此工具按钮前的间隔保留,工具按钮后的间隔消失。反之,若拖动该工具按钮向左直到它接触相邻的前一个工具按钮,则此工具按钮前面的间隔消失,后面的间隔仍保留。

⑥ 删除工具按钮间隔时,只要将间隔一端的工具按钮拖向间隔另一端的工具按钮,直到与另一个按钮重叠一半以上即可。

(3) 单击第一个工具按钮,在工具按钮的属性窗口中,将其 ID 号选择为 ID_FORMAT_TXTFONT,这是将工具按钮与菜单命令联动的关键。将其 Prompt 属性内容改成“改变显示的字体\n 字体”。Prompt 属性是用来指定工具按钮的提示文本。例如若为“改变显示的字体\n 字体”时,则表示当鼠标移至该工具按钮时,在状态栏中就会显示“改变显示的字体”,稍等片刻后还会弹出一个小的提示窗口,显示出“字体”字样。注意:提示窗口显示的内容是 Prompt 属性字符串中“\n”后的内容。

(4) 将第二个工具按钮的 ID 号选择为 ID_FORMAT_TXTCOLOR, Prompt 设为“改变文本的显示颜色\n 颜色”。

2. 工具栏代码的实现

(1) 在 CMainFrame 类中添加一个成员变量 m_wndFormatBar,变量类型为 CToolBar。CToolBar 类封装了工具栏的操作。

(2) 在 CMainFrame::OnCreate() 函数中添加工具栏的创建代码,如图 2 所示的加框部分。

程序说明:

① 主框架类 CMainFrame 用来负责窗口的菜单栏、工具栏和状态栏的创建和更新工作。因此我们将工具栏的创建代码添加在 CMainFrame 的 OnCreate() 函数中。

② CreateEx() 是 CToolBar 类的成员函数,用来创建一个工具栏对象。

③ if 语句的 LoadToolBar() 函数是用来装载工具栏资源。若 CreateEx() 或 LoadToolBar() 的返回值为 0,即调用不成功,则显

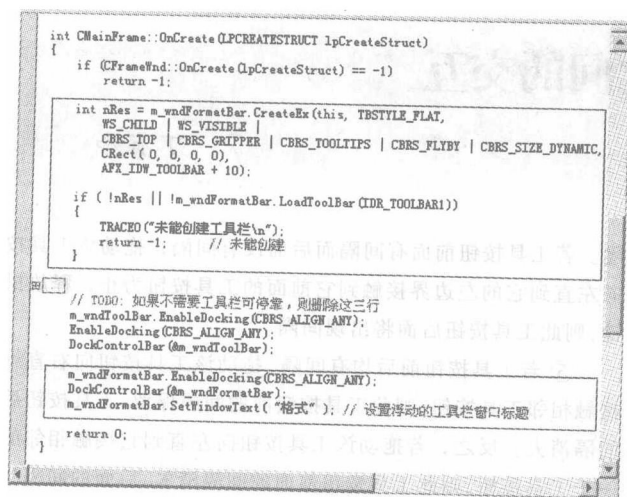


图 2 添加的工具栏的创建代码

示诊断信息“未能创建工具栏”。TRACE0 是一个用于程序调试的跟踪宏。OnCreate() 函数返回 -1 时, 主框架窗口被清除。

④ 应用程序中的工具栏一般具有停靠或浮动特性, `m_wndFormatBar.EnableDocking()` 使得 `m_wndFormatBar` 对象可以停靠, `CBRS_ALIGN_ANY` 表示可以停靠在窗口的任一边。 `EnableDocking(CBRS_ALIGN_ANY)` 是调用的是 `CFrameWnd` 类的成员函数, 用来让工具栏或其他控制条在主框架窗口可以进行停靠操作。 `DockControlBar()` 也是 `CFrameWnd` 类的成员函数, 用来将指定的工具栏或其他控制条进行停靠。

⑤ `AFX_IDW_TOOLBAR` 是系统内部的工具栏子窗口标识, 并将 `AFX_IDW_TOOLBAR + 1` 的值表示默认的状态栏子窗口标识。如果在创建新的工具栏时没有指定相应的子窗口标识, 则会使用默认的 `AFX_IDW_TOOLBAR`。这样, 当打开“视图”菜单并

单击“工具栏”菜单时, 显示或隐藏的工具栏不是原来的工具栏而是新添加的工具栏。因此, 我们需要重新指定工具栏子窗口的标识, 并使其值等于 `AFX_IDW_TOOLBAR + 10`。

(3) 运行程序, 可以看到新添加的工具栏, 如图 3 所示, 左图是工具栏开始的停靠情况, 右图是工具栏浮动的情形。

上述工具按钮与菜单命令联动, 无需进行工具按钮命令的消息映射, 因为该命令已在菜单操作该命令时已映射过。若是单独一个工具按钮, 则需要对该工具按钮进行命令消息的映射, 否则按钮是灰显的。工具按钮的命令消息映射方法与菜单命令相同。

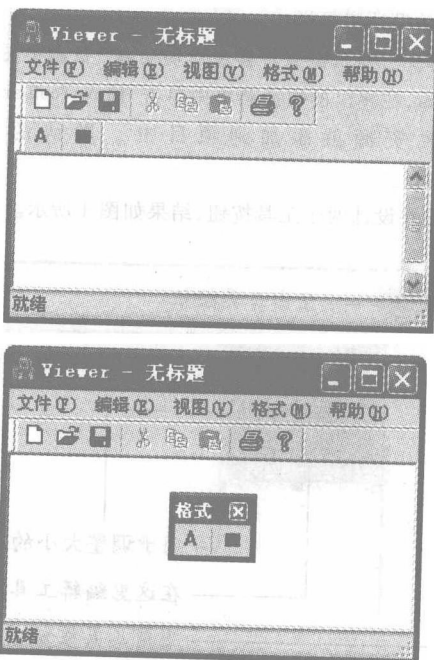


图 3 新工具栏的停靠和浮动

1.3 程序的安装和部署

丁有和

一、安装和部署前的准备

应用程序项目开发之后,就要创建一个安装程序供用户进行安装。现在用 Visual Studio .NET 来创建程序安装和部署项目。

在将程序发布之前,将应用程序项目生成进行重新配置,即将原来默认的 Debug(调试)版本变成了 Release(发布)版本。具体步骤如下:

1. 启动 Visual Studio .NET, 打开单文档应用程序项目 Viewer。
2. 打开“生成”菜单,单击“配置管理器”,弹出对话框。
3. 在“活动的解决方案配置”中选择“Release”,结果如图 1 所示。

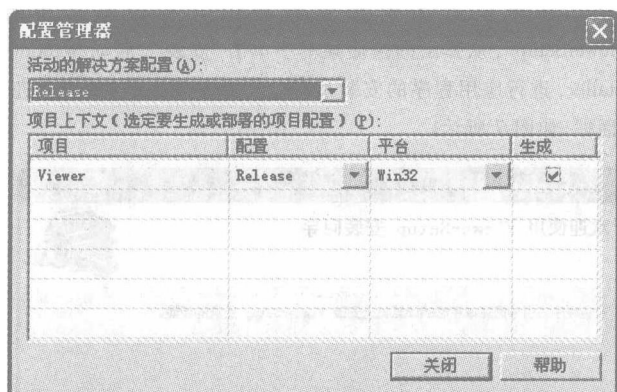


图 1 “配置管理器”对话框

4. 单击“关闭”按钮。打开“生成”菜单,单击“生成解决方案”。生成完后,关闭该解决方案。

需要说明的是,在用 MFC 应用程序向导创建的项目中,可以有 Win32 Debug 和 Win32 Release 两个版本。这两个版本的区别是:Debug 版本的运行程序有相应的调试信息代码,而 Release 版本的运行程序则没有,但 Release 版本的运行程序经过代码的优化,其程序的运行速度被最大加速。因此一般来说,当用户在进行开发的应用程序发布给用户之前,应在项目调试后将其编译成 Win32 Release 版本。编译和生成后的有关文件被放置在项目名文件夹下的 Release 子文件夹中。对于 Debug 版本,其编译和生成后的有关文件被放置在 Debug 子文件夹中。

二、创建一个安装和部署的项目

Visual Studio .NET 不仅提供了创建基于 Windows 或基于 Web 的应用程序和组件的安装程序的工具,也提供了创建用于下载的 .Cab 文件的工具。与 InstallShield 不同的是,Visual Studio .NET 使用 Microsoft Windows Installer 来提供程序安装和配置的功能。

Visual Studio .NET 为用户安装和部署项目提供四种类型的模板:合并模块项目、安装项目、Web 安装项目和 Cab 项目。此外还提供了两个安装程序向导,可帮助我们完成创建部署项目的过程。下面我们使用安装向导来创建一个安装项目。

1. 打开“文件”菜单,选择“新建”→“项目”,弹出“新建项目”对话框。
2. 单击左侧的项目类型“安装和部署项目”,选择右侧的“安装向导”模板,输入项目名称“ViewerSetup”,指定一个项目位置,如图 2 所示。

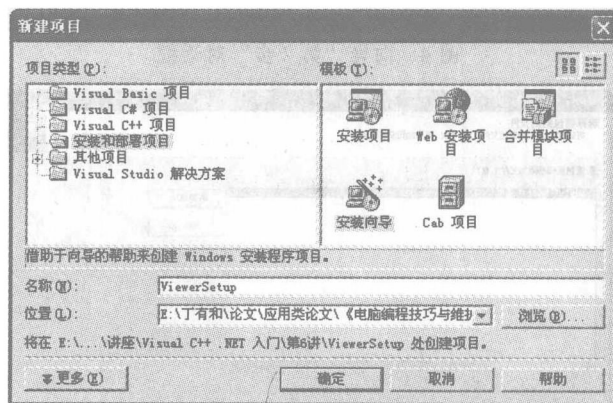


图 2 “新建项目”对话框

3. 单击“确定”按钮,出现如图 3 所示的对话框。
4. 单击“下一步”按钮,出现如图 4 所示的对话框。这里,可以选择安装项目类型以及重新发布的打包类型。
5. 保留默认的选项,单击“下一步”按钮,出现向导“第 3 步”对话框。单击“添加”按钮,在弹出的“添加文件”对话框中,将前面的 Viewer 应用程序可执行文件添加进来,结果如图 5 所示。

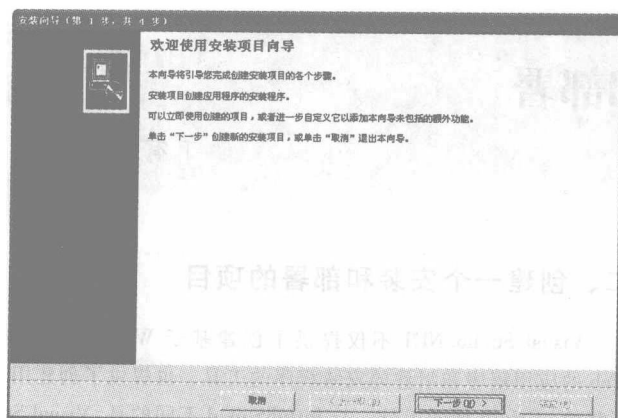


图3 向导“第1步”对话框

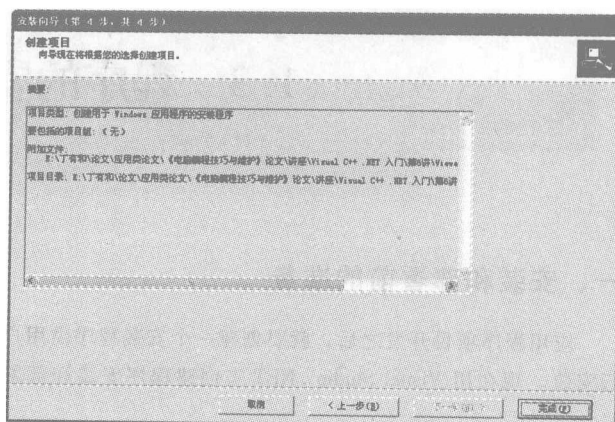


图6 向导“第4步”对话框

应用程序本地化支持的两个模块，一个是 MFC_Loc_FE.msm(东南亚地方)和 VC_MFC.msm(MFC 的支持)，单击“打开”按钮。

三、生成和测试

1. 打开“生成”菜单，单击“生成解决方案”，系统就会在指定项目位置下的“ViewerSetup\Debug”中为该安装项目生成一个 ViewerSetup.msi 安装文件。

2. 在解决方案资源管理器中，右击顶层节点“ViewerSetup”，从弹出的快捷菜单中单击“安装”，启动 Windows Installer，进行应用程序的安装过程。开始出现的是安装“欢迎”对话框，如图 7 所示。

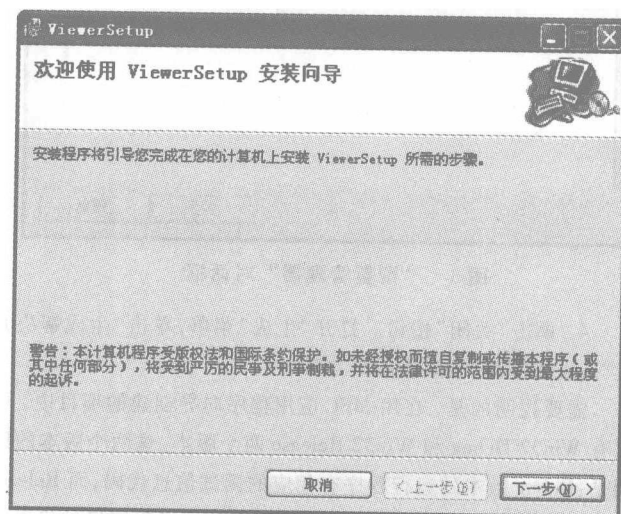


图7 “欢迎”对话框

6. 单击“下一步”按钮，出现如图 6 所示的对话框，显示出我们在前面几个步骤中作出的选择内容。

7. 单击“完成”按钮，系统开始创建。需要说明的是，单击“上一步”按钮可对前面的操作进行修改。

8. 打开“项目”，选择“添加”→“合并模块”，选定用于 MFC

3. 单击“下一步”按钮，进入“选择安装文件夹”对话框，如图 8 所示。

4. 单击“下一步”按钮，进入“确认安装”对话框，如图 9 所示。