

时尚百例丛书

Visual C++ 6.0 MFC

时尚编程百例

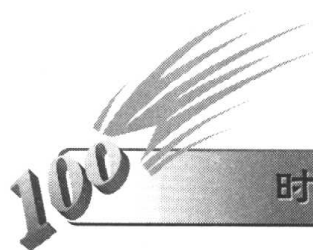
网冠科技 编著

- 流行控件
- 系统和文字处理
- 窗口和对话框
- 图形和图像
- 网络和多媒体
- 拼图游戏



 **机械工业出版社**
CHINA MACHINE PRESS





时尚百例丛书

Visual C++ 6.0 MFC 时尚编程百例

网冠科技 编著

光盘包含书中素材、
效果文件



机械工业出版社

本书讲解如何使用 Visual C++ 6.0 进行软件开发。

书中通过 100 个实例，全面介绍 Visual C++ 的编程方法和目标实现步骤。内容包括：流行控件、系统和文字处理、窗口和对话框、图形和图像、网络和多媒体以及拼图游戏。

本书适合初、中级用户使用。

图书在版编目 (CIP) 数据

Visual C++ 6.0 MFC 时尚编程百例 / 网冠科技编著.

—北京：机械工业出版社，2004.5

（时尚百例丛书）

ISBN 7-111-14316-7

I.V... II.网... III.C 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字 (2004) 第 030065 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

策 划：胡毓坚

责任编辑：时 静

责任印制：施 红

北京忠信诚胶印厂印刷 • 新华书店北京发行所发行

2004 年 5 月第 1 版 • 第 1 次印刷

787mm × 1092mm $\frac{1}{16}$ • 24 印张 • 596 千字

0001-5000 册

定价：40.00 元（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) 68993821、88379646

封面无防伪标均为盗版

时尚百例丛书

追求时尚 追求完美

出版说明

随着计算机迅速应用于人们工作和生活的各个方面，越来越多的职业需要具有计算机应用技能的人才。

如何结合自己的实际工作，选择要学习的软件技术？下面提供了各具体行业的工作人员需学习和掌握的应用软件及相关技术，供读者参考：

- 平面设计及相关行业——Photoshop、CorelDRAW、Illustrator、PageMaker、FreeHand、PhotoImpact 等软件；
- 三维及相关行业——3DS MAX、Maya、SoftImage XSI、Poser、Lightwave 等软件；
- 多媒体设计及相关行业——Authorware、Director、Premiere、Combustion、After Effects、Cult 3D、Web3D 等软件；
- 网络应用及相关行业——Flash、Dreamweaver、Fireworks、FrontPage、ASP、ASP.NET、HTML、PHP、JavaScript、VBScript 等软件及组网建网技术；
- 建筑及装潢设计行业——AutoCAD、3DS MAX、3DS VIZ、Lightscape 等软件；
- 现代工业产品及相关行业——Alias、Pro/E、Solidworks、UG、I-Deas、Rhino、Protel 等软件；
- 软件开发及相关行业——VB.NET、VC.NET、VB、VC、VFP、Delphi、PowerBuilder、C/C++、C++ Builder、JBuilder 等编程软件；
- 办公及应用行业——Windows 9X~2000/XP、Office、WPS Office 等软件及硬件故障排除和网络等技术。

所有与计算机相关的职业，都要求其工作人员有很强的计算机操作技能，熟练地掌握各种相关软件的应用。要做到这一点，必须在掌握软件的基本操作方法的前提下，通过实例演练的方法训练自己，只有通过反复练习，才能做到举一反三，在工作实践中灵活高效地应用。

为了让读者迅速地熟练掌握各种软件的应用方法和技巧，机械工业出版社特别为广大读者推出了这套“时尚百例丛书”，对每一个常用软件都精心制作了 100 个实例，为广大读者提供一条快速掌握计算机应用技能的捷径。

本丛书采用新颖的版式，内容通俗易懂，将软件知识和实例紧密结合。通过对各种实例的详细讲解和操作实践，即使是事先没有学习过这种软件的读者，也能从实例的制作过程中体会到这种软件各项功能的使用方法，并能自己制作出各种实例的效果。这样既节省了读者的大量时间，又能使读者在反复实践的同时，提高学习兴趣，并将学到的知识和技能迅速应用到实际工作中去。

机械工业出版社

前言

随着 21 世纪面向对象程序设计技术的蓬勃发展,在全球软件业掀起了面向对象程序设计的浪潮,在众多的面向对象程序设计语言中,Visual C++ 6.0 将程序设计方法和可视的软件开发环境完美地结合在一起,其崭新的开发数据库和适应 Internet 应用程序的新特点引起了开发人员的广泛关注,尤其受广大软件设计人员的青睐,逐渐成为软件设计、Windows 应用程序开发的首选语言。

Visual C++ 6.0 是汇集了 MS 公司技术精华的主流产品,使用 Visual C++ 6.0 可以开发强大的 32 位应用程序,能为用户提供全方位的服务,具有广阔的商业前景。

一位资深的专家曾说过“真正的程序员用 VC”,现在学习 VC 的读者越来越多。为了适合广大 VC 爱好者的需要,帮助他们迅速掌握和提高自己的软件设计能力,我们精心编写了这本书,希望能够满足广大读者的需要。

本书以实例的形式向读者讲解如何使用 Visual C++ 6.0,我们在以往使用 Visual C++ 6.0 开发应用程序时,最深的感触是关于理论的知识讲解过多,而实例却讲解得非常简略,给人的感觉是虽然理论知识已有基础,但是却不能开发出好的应用程序。本书则以实例为主,在讲解实例的过程中,同时讲解与之相关的知识点,使读者能够通过实例的制作掌握 Visual C++ 6.0 的使用方法。

本书共讲解 Visual C++ 实例 100 个,这些实例深入浅出,可以满足不同层次读者的切实需要。对于初学者,通过本书的学习,可以迅速地了解 Visual C++ 6.0 的开发思想和方法,并可以快速创建自己的应用程序,从而熟练掌握 Visual C++ 6.0 的使用。对于已经有一定基础的中级开发人员,可以通过本书的实例学习到一些具有实际意义的开发方法,进一步提高自己开发应用程序的能力。对于高级开发人员,这 100 个实例讲解了许多处理问题的方法和技巧,非常适合高级人员参考。

在编写本书时,我们注重强调实例的实际意义,每个实例的程序源代码都结合实际开发应用程序过程中的常见问题。在开发应用程序时,读者可以直接使用部分甚至全部代码来完成开发工作。



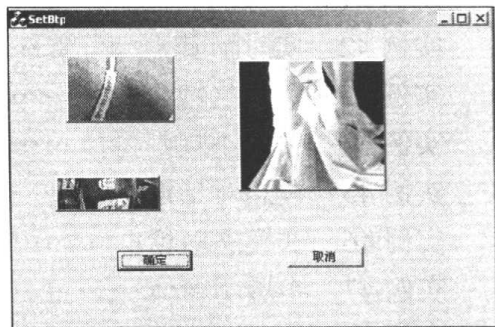
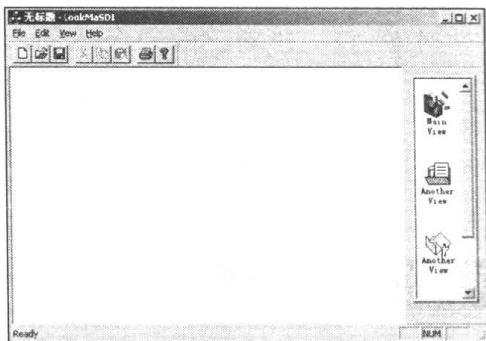
网冠科技

目 录

出版说明 前 言

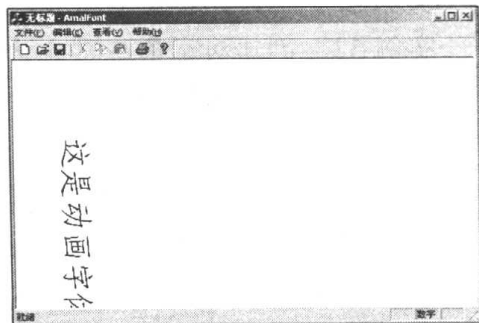
第一篇 流行控件

实例 1	漂亮的图文菜单	2
实例 2	带图标菜单	6
实例 3	彩色按钮	10
实例 4	气球式提示栏	13
实例 5	滑块控件的随时记录	15
实例 6	3D 文本按钮的实现	18
实例 7	动态菜单的实现	22
实例 8	具有不可选选项的组合框	26
实例 9	显示彩色文本的列表框	32
实例 10	颜色渐变的进度条	37
实例 11	树形控件的应用	45
实例 12	动态填充一个组合框	48
实例 13	动画按钮	51
实例 14	在按钮上加载位图	57
实例 15	在状态栏中显示鼠标的位置	60
实例 16	实现快捷菜单	63
实例 17	为菜单添加核对标记	65
实例 18	启用和禁用菜单	69
实例 19	显示当前时间的状态栏	72

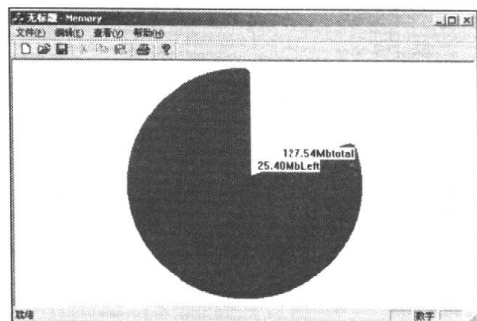
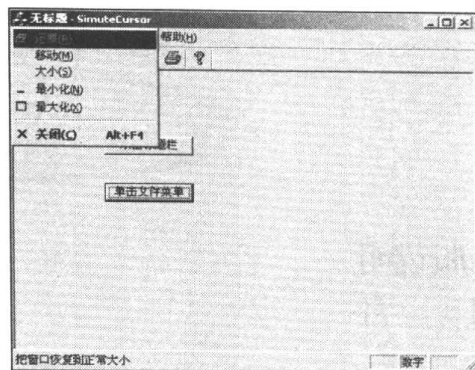


第二篇 系统和文字处理

实例 20	动画字体	76
实例 21	倾斜文字	80
实例 22	以二进制形式查看文件	82
实例 23	创建自己的文本编辑器	87
实例 24	统计中英文字符数	90
实例 25	限定鼠标区域	92

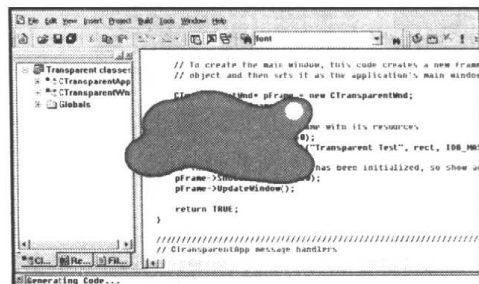
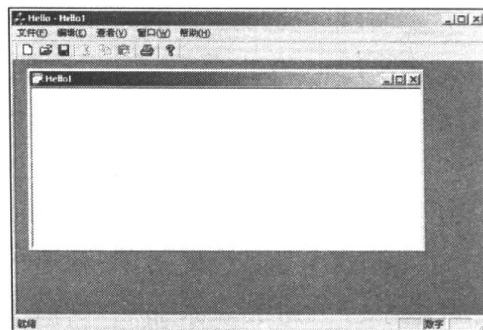


- 实例 26 获取和修改鼠标的双击时间 95
- 实例 27 模拟鼠标的单双击 99
- 实例 28 快速检索指定文件 102
- 实例 29 串行化功能的实现 105
- 实例 30 获取和修改计算机的名称 108
- 实例 31 设置显示器的分辨率 111
- 实例 32 组合框的自动查询技术 114
- 实例 33 定位输出文本 118
- 实例 34 多线程运行程序 123
- 实例 35 线程之间的通信 126
- 实例 36 改变光标形状 130
- 实例 37 独特的复制和粘贴操作 133
- 实例 38 内存监测器 137



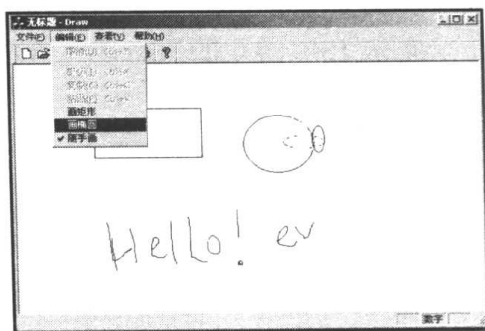
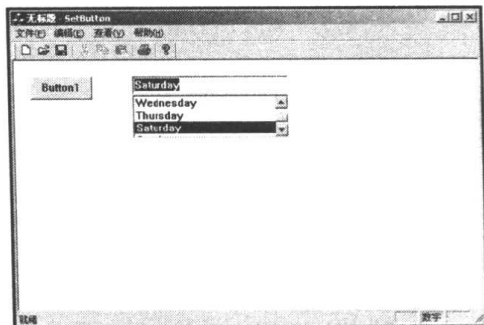
第三篇 窗口和对话框

- 实例 39 AppWizard 编写的应用程序 141
- 实例 40 多色窗口的实现 145
- 实例 41 特殊的光标监视器 148
- 实例 42 实时捕获光标位置 152
- 实例 43 添加气泡帮助 154
- 实例 44 实现不规则窗口 158
- 实例 45 模态对话框的实现 160
- 实例 46 非模态对话框 163
- 实例 47 半透明窗体 166
- 实例 48 在对话框中创建菜单和工具条 168
- 实例 49 修改文件对话框 171
- 实例 50 可扩展对话框 173
- 实例 51 在系统栏中添加动画图标 177
- 实例 52 在对话框中设置属性页 180
- 实例 53 创建对话条 184
- 实例 54 分隔窗体 187
- 实例 55 云形窗口 189
- 实例 56 有渐变背景的对话框 193
- 实例 57 打开消息框 195



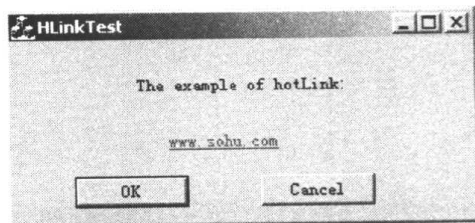
第四篇 图形和图像

- 实例 58 在视图添加控件 198
- 实例 59 半透明三维物体 200
- 实例 60 位图的显示 204
- 实例 61 多功能的画笔 206
- 实例 62 获得指定点的颜色 210
- 实例 63 奇妙的变化线 212
- 实例 64 智能滚动条 215
- 实例 65 实现所见即所得打印 217
- 实例 66 多页打印 222
- 实例 67 制作简单的“日记本” 225
- 实例 68 图像的缩放 229
- 实例 69 获取图像的大小 234
- 实例 70 使窗体保持在最前 238
- 实例 71 显示多姿多彩的动画图标 240
- 实例 72 列表视图的实现 245
- 实例 73 用鼠标画线 249
- 实例 74 画椭圆、矩形和随手画功能 253

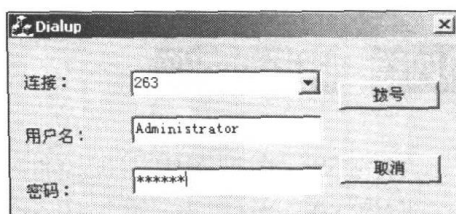


第五篇 网络和多媒体

- 实例 75 制作 MIDI 播放器 259
- 实例 76 检索局域网内的计算机 263
- 实例 77 播放 WAV 文件 267
- 实例 78 Media Player 播放器 272
- 实例 79 Web 浏览器 276
- 实例 80 网络浏览器 279
- 实例 81 判断网址是否有效 283
- 实例 82 实现对 HTTP 的监控 287
- 实例 83 广播信息 290
- 实例 84 用 ActiveMovie 控件制作播放器 293
- 实例 85 用 MSCom 控件进行串口通信 296
- 实例 86 HTTP 查询 300
- 实例 87 利用 FTP 传送文件 303
- 实例 88 监控电子邮件的收发 306
- 实例 89 创建超级连接界面 310



- 实例 90 利用 SMTP 发送 E-mail.....316
- 实例 91 服务器端应用程序.....323
- 实例 92 客户端应用程序.....329
- 实例 93 电话拨号上网.....335



第六篇 拼图游戏

- 实例 94 创建游戏的初始界面.....339
- 实例 95 位图的读入.....342
- 实例 96 用 Static 控件显示位图.....346
- 实例 97 图格的移动.....351
- 实例 98 编写游戏的启动代码.....359
- 实例 99 游戏完成条件的判断.....362
- 实例 100 游戏的进一步完善.....368



第一篇

流行控件

本篇主要讲解在 Visual C++ 6.0 中用于开发程序界面的一些流行控件,控件是一种特殊的窗口,它是实现用户和应用软件之间进行良好交互的桥梁,因此,制作控件是程序开发人员必须掌握的一种技能,本篇针对当前学习和程序开发的需要,主要讲解了界面开发所用的主要控件,包括命令按钮、组合框、进度条及菜单等。

通过对本篇的学习,读者将深入了解如何制作控件,并能够运用控件的有关知识来开发一些应用程序界面,从而为应用 Visual C++ 6.0 进行程序设计打下基础。

实例 1 漂亮的图文菜单

实例说明

本实例实现的是在视图的对话框组件中显示漂亮的图文菜单，其运行效果如图 1-1 所示。

本实例的特点有：显示带文本的 32 像素 × 32 像素的图标；当鼠标移动到图标上方时按钮弹起；可以自动换行的多行显示文本等。

本实例用到的主要知识点有：SetBarStyle() 函数、SetItem() 函数和 OnSelchangeListContent() 函数等。

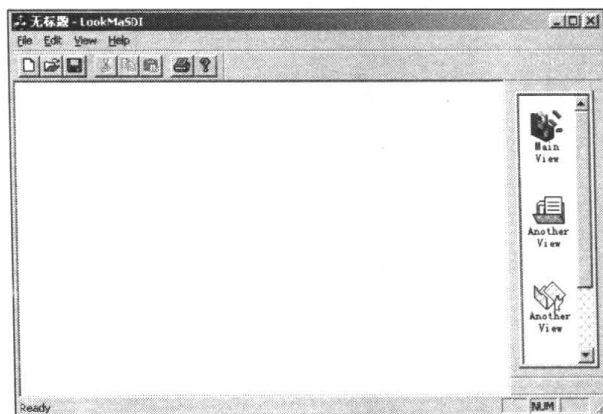


图 1-1

编程思路

首先，由于本实例实现的是漂亮的图文菜单，所以，首先应为应用程序添加图标资源，本实例为应用程序添加 4 个图标资源。

其次，为了实现在菜单上显示图文，应为应用程序创建一个类，使用 Class Wizard 为图文菜单创建一个新类 CContentMenu，其基类为 CListBox。

最后，映射对话框对应的类，执行 View→Class Wizard... 菜单项，弹出 Adding a class 对话框。选择 Selecting an existing class 项，单击【OK】按钮，把对话框和已有的类联系起来，在 Class List 中选择 CMainFrame 类，单击【Select】按钮，把对话框映射给 CMainFrame 类，这样就可以用 CMainFrame 类来管理对话框，可以将对话框放置在主框架中。

具体知识如下：

```
void OnSelchangeListContent() //当选择列表框中的内容时响应函数
void SetItems(
    pCONTENT_ITEMS pItem, //设置的内容
    int nNumItems //设置的序数
)
void SetBarStyle(
    DWORD dwStyle //对话框条的风格
)
```

创作步骤

1. 启动 Visual C++ 6.0，执行 New→Project 新建菜单项，新建一个 MFC AppWizard(Exe) 应用，在左

上方的工程文本中输入“LookMaSDI”，完成后单击【OK】按钮，进入 MFC AppWizard 向导。

2. 在 MFC AppWizard 向导的第一步选择 Single Document，单击【Finish】按钮完成工程的建立。

3. 加入对话框组件。单击 Project→Add to Project→Visual C++ Componets...菜单项，打开 Visual C++ Componets 对话框，选择 Dialog bar 项，如图 1-2 所示，然后单击【Insert】按钮，将弹出一个消息框，询问“是否插入该组件？”单击【确定】按钮。

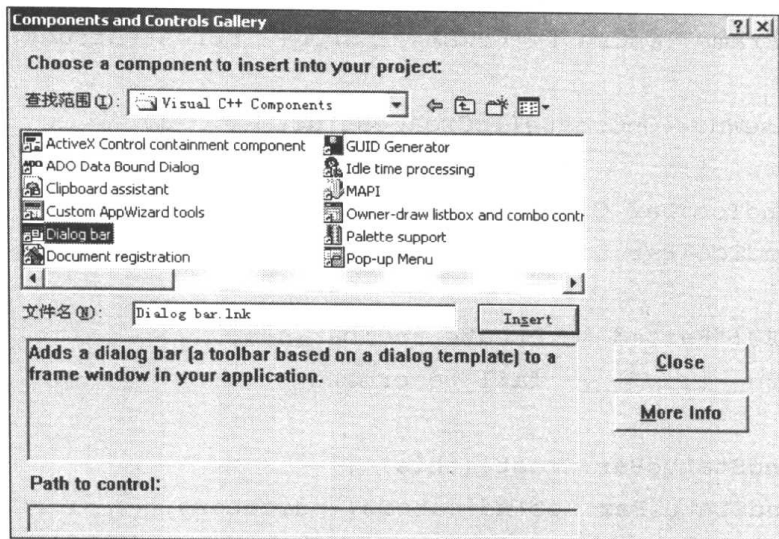


图 1-2

4. 修改组件对话框。进入 Dialog bar 对话框，在 Dialog bar Name 项中，修改其名为 ContentMenu；在 Member variable name 项中，修改其名为 m_wndContentMenu；在 Default docking edge 组件框中，修改默认属性为 Right。修改完对话框的属性后，单击【OK】按钮，插入对话框组件。

5. 添加图标资源。执行 Insert→Resource...菜单项；打开 Insert Resource 对话框；在 Resource Type 列表框中选择 Icon，执行【New】按钮添加位图资源。重复为工程添加 4 个位图资源，其 ID 分别为 IDI_CALENDAR、IDI_COMPOSE、IDI_FILED 和 IDI_FINISHED。

6. 编辑对话框资源，为对话框添加一个 List Box 控件。映射对话框所对应的类，在打开 CG_IDD_CONTENTMENU 对话框资源的情况下，执行 View→Class Wizard...菜单项，弹出 Adding a class 对话框。选择 Select an existing class 项，单击【OK】按钮，把对话框和已有的类联系起来，在 Class List 中选择 CMainFrame 类，单击【Select】按钮，把对话框映射给 CMainFrame 类。

7. 创建图文菜单类。执行 View→Class Wizard，利用 Class Wizard 为图文菜单创建一个新类 CcontentMenu，基类为 CListBox，其实现文件和头文件分别为 ContentMenu.cpp 和 ContentMenu.h。

8. 实现图文菜单。打开 MainFrame.h 文件，为 CmainFrame 类增加一个成员变量和引用头文件，代码如下：

```
// MainFrm.h : interface of the CMainFrame class
protected:
CContentMenu    m_ContentMenu;
#include "Contentmenu.h"
```

在 MainFrame.cpp 文件中，创建一个结构 ContentInfo，用于记录菜单信息，如下：

```
static CONTENT_ITEMS    ContentInfo[] =
{
```

```
{IDI_COMPOSE, "Main View"},
{IDI_FINISHED, "Another View"},
{IDI_FILED, "Another View"},
{IDI_CALENDAR, "Another View"},
};
```

在 CMainFrame 类的消息函数 OnCreate() 中, 修改代码以支持图文菜单。修改后的源代码如下:

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CFrameWnd::OnCreate(lpCreateStruct) == -1)
        return -1;
    if (!m_wndToolBar.Create(this) ||
        !m_wndToolBar.LoadToolBar(IDR_MAINFRAME))
    {
        TRACE0("Failed to create toolbar\n");
        return -1;    // fail to create
    }
    if (!m_wndStatusBar.Create(this) ||
        !m_wndStatusBar.SetIndicators(indicators,
            sizeof(indicators)/sizeof(UINT)))
    {
        TRACE0("Failed to create status bar\n");
        return -1;    // fail to create
    }
    // ODO: Remove this if you don't want tool tips or a resizable toolbar
    m_wndToolBar.SetBarStyle(m_wndToolBar.GetBarStyle() |
        CBRS_TOOLTIPS | CBRS_FLYBY | CBRS_SIZE_DYNAMIC);
    // TODO: Delete these three lines if you don't want the toolbar to
    // be dockable
    m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY);
    EnableDocking(CBRS_ALIGN_ANY);
    DockControlBar(&m_wndToolBar);

    if (!m_wndContentMenu.Create(this, CG_IDD_CONTENTMENU,
        CBRS_RIGHT | CBRS_TOOLTIPS | CBRS_FLYBY | CBRS_HIDE_INPLACE,
        CG_ID_VIEW_CONTENTMENU))
    {
        TRACE0("Failed to create dialog bar m_wndContentMenu\n");
        return -1;    // fail to create
    }
    m_wndContentMenu.EnableDocking(CBRS_ALIGN_LEFT
        CBRS_ALIGN_RIGHT);
    EnableDocking(CBRS_ALIGN_ANY);
    DockControlBar(&m_wndContentMenu);
}
```

```
m_ContentMenu.SubclassDlgItem(IDC_LIST_CONTENTS,
&m_wndContentMenu);
m_ContentMenu.SetItems(ContentInfo, sizeof(ContentInfo) /
sizeof(CONTENT_ITEMS));
return 0;
}
```

9. 响应图文菜单。编辑的代码如下：

```
void CMainFrame::OnSelchangeListContents()
{
    switch (m_ContentMenu.GetCurSel()) //得到当前选择的项
    {
        case 0:
            AfxMessageBox("You have selected First Item"); //弹出消息框
            break;
        case 1:
            AfxMessageBox("You have selected Second Item");
            break;
        case 2:
            AfxMessageBox("You have selected Third Item");
            break;
        case 3:
            AfxMessageBox("You have selected Fourth Item");
            break;
    }
}
```

10. 编译并运行程序，其效果如图 1-1 所示。

实例 2 带图标的菜单

实例说明

本实例实现的是带图标的菜单，其运行效果如图 2-1 所示。

运行本实例后，当打开“文件”菜单后可以看到文件菜单的“新建”、“打开”、“保存”和“打印”等菜单项旁边都有一个意义非常明确的图标。

本实例用到的主要知识点有：LoadMenu()函数和 ModifyODMenu()函数等。

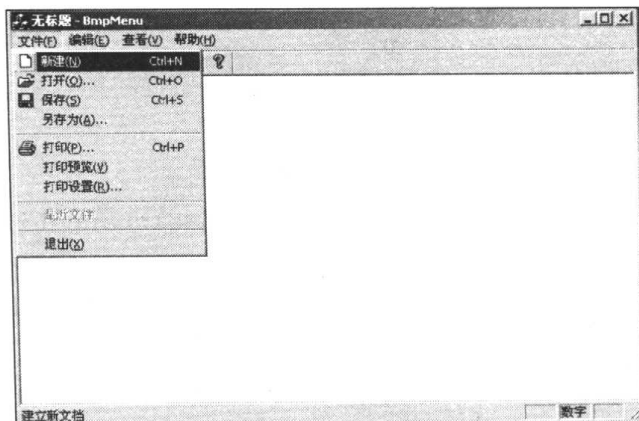


图 2-1

编程思路

首先，由于本实例实现的是带图标的菜单，所以首先为应用程序添加位图资源。

其次，定义一个功能比较完善的菜单类 BCMenu，在该类中实现图标菜单的效果。

最后，先生成一个 BCMenu 类的对象，然后用 LoadMenu 函数将菜单资源与菜单对象联系起来，接着通过 ModifyODMenuA 函数修改某个菜单项的图标，最后将其设置为 CMainFrame 的菜单。

创作步骤

1. 启动 Visual C++ 6.0，执行 New→Project 新建菜单项，新建一个 MFC AppWizard(Exe) 应用，在左上方的工程文本中输入“BmpMenu”，完成后单击【OK】按钮，进入 MFC AppWizard 向导。

2. 在 MFC AppWizard 向导的第一步选择 Single Document，单击【Finish】按钮完成工程的建立。

3. 向工程中添加 BCMenu.cpp 和 BCMenu.h 文件。单击 Project→Add To Project→File...菜单项，打开 Insert Files into Project 对话框，选择 BCMenu.cpp 文件，单击【OK】按钮，如图 2-2 所示，把 BCMenu.cpp 加入到工程中，然后用同样的方法把 BCMenu.h 文件加入到工程中。

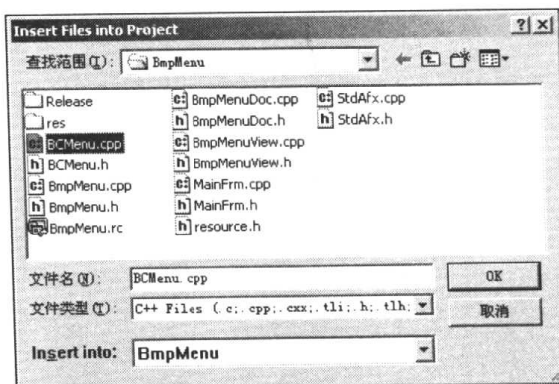


图 2-2

4. 在 CMainFrame 头文件中定义一个 BMenu 的对象，同时包含 BCMenu.h 头文件。如下所示：

```
// MainFrm.h : interface of the CMainFrame class
#include "BCMenu.h"
public:
```

BCMenu m_menu;

5. 在 CMainFrame 中加入一个生成图标菜单的函数 NewMenu, 其定义如下:

```
//生成图标菜单
HMENU CMainFrame::NewMenu()
{
    m_menu.LoadMenu(IDR_MAINFRAME);
    //将图标加到菜单中
    m_menu.ModifyODMenuA(NULL, ID_FILE_NEW, IDB_FILE_NEW);
    m_menu.ModifyODMenuA(NULL, ID_FILE_OPEN, IDB_FILE_OPEN);
    m_menu.ModifyODMenuA(NULL, ID_FILE_SAVE, IDB_FILE_SAVE);
    m_menu.ModifyODMenuA(NULL, ID_FILE_PRINT, IDB_FILE_PRINT);
    //返回图标菜单的句柄
    return m_menu.Detach();
}
```

6. 在 CbmpMenuApp 的 InitInstance 函数中加入如下代码, 设置图标菜单为主窗口的菜单。

```
BOOL CbmpMenuApp::InitInstance()
{
    AfxEnableControlContainer();
#ifdef _AFXDLL
    Enable3dControls();          // Call this when using MFC in a shared DLL
#else
    Enable3dControlsStatic();    // Call this when linking to MFC statically
#endif
    SetRegistryKey(_T("Local AppWizard-Generated Applications"));
    LoadStdProfileSettings();    // Load standard INI file options (including
MRU)

    // Register the application's document templates. Document templates
    // serve as the connection between documents, frame windows and views.
    CSingleDocTemplate* pDocTemplate;
    pDocTemplate = new CSingleDocTemplate(
        IDR_MAINFRAME,
        RUNTIME_CLASS(CbmpMenuDoc),
        RUNTIME_CLASS(CMainFrame),      // main SDI frame window
        RUNTIME_CLASS(CbmpMenuView));
    AddDocTemplate(pDocTemplate);
    // Parse command line for standard shell commands, DDE, file open
    CCommandLineInfo cmdInfo;
    ParseCommandLine(cmdInfo);
    // Dispatch commands specified on the command line
    if (!ProcessShellCommand(cmdInfo))
        return FALSE;
    //将默认的菜单清除
    CMenu* pMenu = m_pMainWnd->GetMenu();
```

```

if (pMenu)pMenu->DestroyMenu();
//设置菜单为新的图标菜单
HMENU hMenu = ((CMainFrame*) m_pMainWnd)->NewMenu();
pMenu = CMenu::FromHandle( hMenu );
m_pMainWnd->SetMenu(pMenu);
((CMainFrame*)m_pMainWnd)->m_hMenuDefault = hMenu;
// The one and only window has been initialized, so show and update it.
m_pMainWnd->ShowWindow(SW_SHOW);
m_pMainWnd->UpdateWindow();
return TRUE;
}

```

7. 添加消息响应函数。执行 View→Class Wizard, 打开 Class Wizard 对话框, 选择 Message Maps 选项卡, 在 Class name 框中选择 CMainFrame 类, 在 CMainFrame 类中加入 WM_MEASUREITEM, WM_MENUCHAR 和 WM_INITMENUPOPUP 三个消息的相应函数, 如图 2-3 所示。

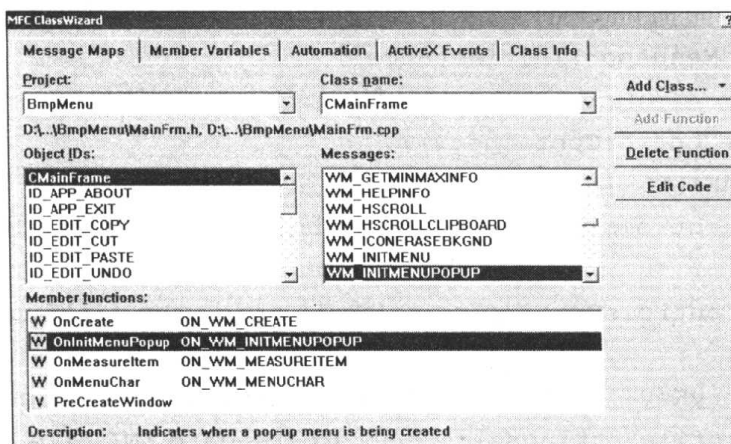


图 2-3

8. 编辑消息响应函数的消息代码。添加的代码如下:

```

void CMainFrame::OnMeasureItem(int nIDCtl, LPMEASUREITEMSTRUCT
lpMeasureItemStruct)
{
// TODO: Add your message handler code here and/or call default
BOOL setflag=FALSE;
if(lpMeasureItemStruct->CtlType==ODT_MENU){
    if(IsMenu((HMENU)lpMeasureItemStruct->itemID)){
        CMenu* cmenu =
CMenu::FromHandle((HMENU)lpMeasureItemStruct->itemID);
        if(m_menu.IsMenu(cmenu)){
            m_menu.MeasureItem(lpMeasureItemStruct);
            setflag=TRUE;
        }
    }
}
}

```