

万水创作效果百例丛书

# Visual C++ .NET

## 精彩编程 百例

吕智 马瑞力 等编著



含 1CD



中国水利水电出版社  
www.waterpub.com.cn

万水创作效果百例丛书

# Visual C++.NET 精彩编程百例

吕智 马瑞力 等编著

中国水利水电出版社

## 内 容 提 要

本书通过 100 个精选的例子讲解了利用 Visual C++.NET 进行应用程序开发的各个方面, 涵盖了界面编程、图像和多媒体编程、对话框与控件编程、文件与系统编程、线程操作与 COM 组件编程、网络和数据库编程、OpenGL 编程、OLE 和 GDI+ 编程等方面的内容。选例经典而具有代表性, 分析透彻, 编程逻辑思路简练, 具有很强的实用性、可操作性和参考价值。使用本书的最好方法是通过学习掌握实例中的技术和技巧, 然后使用这些技术去尝试实现更复杂的功能, 应用到更多方面。

本书很适合 Visual C++.NET 的初学者和中级程序员, 也可作为计算机专业师生的自学、教学用书。本书所附光盘为全书所有实例的源代码。

## 图书在版编目 (CIP) 数据

Visual C++.NET 精彩编程百例 / 吕智等编著. —北京: 中国水利水电出版社, 2004

(万水创作效果百例丛书)

ISBN 7-5084-1290-7

I. V… II. 吕… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 087349 号

书 名	Visual C++.NET 精彩编程百例
作 者	吕智 马瑞力 等编著
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: <a href="http://www.waterpub.com.cn">www.waterpub.com.cn</a> E-mail: <a href="mailto:mchannel@263.net">mchannel@263.net</a> (万水) <a href="mailto:sales@waterpub.com.cn">sales@waterpub.com.cn</a>
经 售	电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水) 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787mm×1092mm 16 开本 22.25 印张 498 千字
版 次	2004 年 9 月第 1 版 2004 年 9 月第 1 次印刷
印 数	0001—5000 册
定 价	38.00 元 (含 1CD)

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

# 前 言

Visual C++.NET 是运行在 Windows 平台上的高度综合性的软件开发工具,它集界面设计、代码编辑、编译、连接和调试功能于一体,在高性能的执行效率与底层控制和快速可视化开发方面均表现出色,其应用程序向导为很大一部分类型的程序提供了框架代码,用户不必编写很多烦琐的程序代码。

要想真正掌握 Visual C++.NET 的精髓并且编写出功能强大的应用程序,仅掌握编程语言的语法概念是远远不够的,编程水平的提高需要在实际应用中积累点滴经验。本书的目的是通过 100 个典型实用的应用实例详细讲解 Visual C++.NET 编程所涉及的知识。本书实例按照功能来分类,每个编程实例作为一个相对完整的知识点,由浅入深,从基础编程方法到高级编程技巧,内容丰富,能够满足不同层次读者的各种需求。

本书共分八篇,介绍了以下内容。

第一篇:界面编程。介绍了使用 Visual C++.NET 创建各种界面的方法和技巧。

第二篇:图像和多媒体编程。介绍了使用 Visual C++.NET 实现文本、图形、图像、动画、音频和视频等的方法和技巧。

第三篇:对话框与控件编程。介绍在 Visual C++.NET 编程中各种风格的 Windows 对话框和控件的使用方法和编程技巧。

第四篇:文件与系统编程。介绍使用 Visual C++.NET 实现各种文件与系统编程的方法和技巧。

第五篇:线程操作与 COM 组件编程。介绍在 Visual C++.NET 编程中线程操作的技术和如何使用 COM 技术开发应用程序。

第六篇:网络和数据库编程。介绍使用 Visual C++.NET 创建网络应用程序所用到的各种编程方法和技巧以及 Visual C++.NET 提供的几种数据库访问方式。

第七篇:OpenGL 编程。介绍在 Visual C++.NET 编程中一些常用的 OpenGL 编程方法和技巧。

第八篇:OLE 和 GDI+编程。介绍 OLE 和 GDI+这两项技术在 Visual C++.NET 编程中的常用编程方法和技巧。

本书所附光盘的内容包括了开发实例的所有程序源代码,所有程序均在 Visual C++.NET 中编译通过,运行这些源程序时需要在本机安装 Visual C++.NET 编程环境。本书主要由吕智、马瑞力、王强、胡浩编写,另外参加本书编写的还有胡新诚、刘志国、徐桂园、李锦卫、方多诚、王晓青等。由于时间仓促及编者水平有限,书中疏漏和错误之处在所难免,恳请各位专家和广大读者批评指正。

编 者

2004 年 5 月

# 目 录

前言

## 第一篇 界面编程

实例 1	“欢迎”界面 .....	2
实例 2	动态字幕效果 .....	5
实例 3	透明窗体 .....	9
实例 4	静态切分窗口 .....	12
实例 5	滚动视图 .....	15
实例 6	字形窗口 .....	18
实例 7	多边形窗口 .....	21
实例 8	HTML 浏览器 .....	23
实例 9	窗口的动画效果 .....	27
实例 10	实现动画图标 .....	30
实例 11	动态切分窗口 .....	34
实例 12	橡皮区矩形 .....	38
实例 13	多个定时器 .....	41
实例 14	椭圆窗体 .....	45

## 第二篇 图像和多媒体编程

实例 15	BMP 文件浏览器 .....	51
实例 16	图像的镜像处理 .....	55
实例 17	图像的分割处理 .....	62
实例 18	使用画笔 .....	68
实例 19	使用画刷 .....	71
实例 20	显示空心字 .....	74
实例 21	显示渐变字 .....	77
实例 22	显示艺术字体 .....	80
实例 23	马赛克效果 .....	85
实例 24	MIDI 播放器 .....	89
实例 25	CD 播放器 .....	93
实例 26	纹理场景效果 .....	97
实例 27	Bezier 线框曲面 .....	100
实例 28	输出旋转文本 .....	103

### 第三篇 对话框与控件编程

实例 29	树形控件 .....	107
实例 30	列表控件 .....	112
实例 31	时间控件 .....	116
实例 32	彩色按钮 .....	121
实例 33	组合框控件 .....	124
实例 34	单选按钮和复选框 .....	127
实例 35	工具栏编程 .....	130
实例 36	状态栏编程 .....	133
实例 37	对话框显示图片 .....	136
实例 38	模式对话框 .....	140
实例 39	非模式对话框 .....	143
实例 40	位图按钮 .....	146
实例 41	对话框编程 .....	149

### 第四篇 文件与系统编程

实例 42	鼠标绘图 .....	153
实例 43	键盘输入 .....	157
实例 44	获取文件属性 .....	160
实例 45	访问和修改注册表 .....	163
实例 46	设定计算机名称 .....	167
实例 47	列举驱动器 .....	170
实例 48	关闭操作系统 .....	172
实例 49	查找文件 .....	175
实例 50	获取磁盘空间 .....	178
实例 51	读写文件 .....	181
实例 52	系统进程检测 .....	184
实例 53	获取系统信息 .....	187
实例 54	获取文件夹路径 .....	191
实例 55	设定系统时间 .....	194

### 第五篇 线程操作与 COM 组件编程

实例 56	创建线程 .....	198
实例 57	线程间的通信 .....	201
实例 58	使用自定义消息 .....	205

实例 59	线程间的优先级 .....	208
实例 60	线程间的同步 .....	213
实例 61	重用 C++对象 .....	217
实例 62	创建永久对象 .....	220
实例 63	MFC 建立 COM .....	223
实例 64	测试 COM 组件 .....	225

## 第六篇 网络和数据库编程

实例 65	聊天室服务器 .....	230
实例 66	HTTP 应用程序 .....	234
实例 67	FTP 应用程序 .....	239
实例 68	HTTP 服务器 .....	244
实例 69	发送电子邮件 .....	247
实例 70	接收电子邮件 .....	250
实例 71	ATL Server Web .....	253
实例 72	DAO 获取信息 .....	255
实例 73	DAO 数据库编程 .....	257
实例 74	ADO 访问数据库 .....	261

## 第七篇 OpenGL 编程

实例 75	绘制对比矩形 .....	266
实例 76	绘制直线 .....	268
实例 77	绘制圆和椭圆 .....	270
实例 78	绘制三角形 .....	272
实例 79	缩放变换 .....	274
实例 80	旋转变换 .....	277
实例 81	平移变换 .....	280
实例 82	绘制点 .....	283
实例 83	OpenGL 编程 .....	285
实例 84	实现简单动画 .....	292
实例 85	视口投影变换 .....	298
实例 86	键盘控制图形 .....	301

## 第八篇 OLE 和 GDI+编程

实例 87	OLE 容器 .....	305
实例 88	保存 OLE 信息 .....	307

实例 89	设置跟踪矩形 .....	310
实例 90	显示多个 OLE 项 .....	315
实例 91	ActiveX 服务器 .....	317
实例 92	插值模式 .....	321
实例 93	图像区域平铺 .....	324
实例 94	文字填充 .....	327
实例 95	图形容器 .....	330
实例 96	线帽组合 .....	333
实例 97	半透明线条 .....	336
实例 98	颜色再变换表 .....	339
实例 99	旋转和扭曲图像 .....	342
实例 100	显示垂直文本 .....	345



# 第一篇

## 界面编程

丰富多彩的界面设计是 MFC 的一大特色，利用 Visual C++.NET 可以很容易地制作出各种独具特色的界面。本篇将向读者介绍创建这些界面的方法和技巧。主要内容包括如何创建简单的应用程序、如何实现各种不规则形状的窗体、如何实现各种对话框、如何实现动态与静态切分窗口、如何创建动态效果的自定义界面等。通过这些简单有趣的实例为以后进一步的编程学习打下基础。

*Let's GO!* 

## 实例

## 1

## “欢迎”界面



## 实例说明

在本例中，我们将利用 MFC 应用程序向导创建一个单文档应用程序，在单文档的窗口中，将显示一行字：“你好！欢迎使用 VC++.NET”。本例的重点是学习如何启动 Visual C++.NET 应用程序，以及如何使用 MFC 应用程序向导创建一个基本的应用程序，同时也将学习如何在单文档的视图中显示文字。



## 操作步骤

① 启动 Visual C++.NET，在 Windows 的任务栏中选择【开始】|【程序】| Microsoft Visual Studio.NET| Microsoft Visual Studio.NET，调用 Microsoft Visual Studio.NET 的开发环境。选择 MFC 应用程序模板，项目名称为 Welcome。

② 在 MFC 应用程序向导中的【应用程序类型】选项中，应用程序类型选择【单文档】，其余选项采用默认值，如图 1-1 所示。

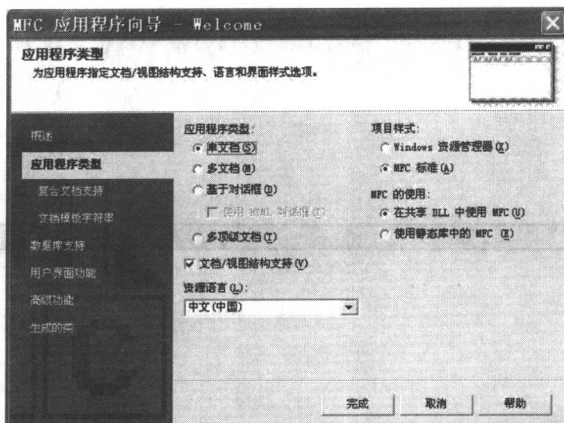
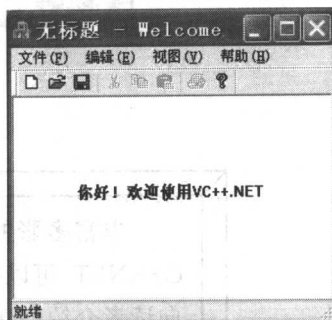


图 1-1 选择应用程序类型

③ 在【用户界面功能】选项中，可以选择应用程序的界面特征，这里采用默认值，如图 1-2 所示。

④ 在【生成的类】选项中可以看到应用程序即将创建的类、头文件和实现文件的名称，如图 1-3 所示。



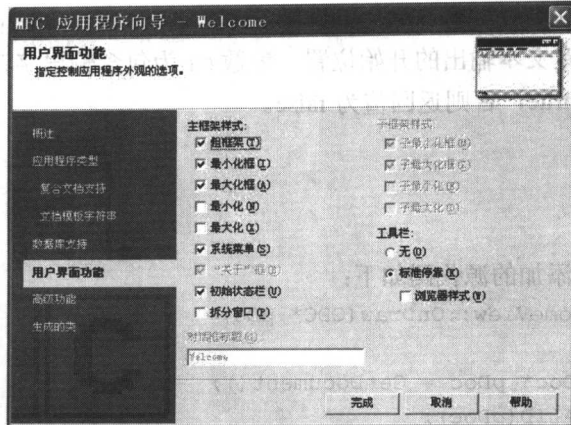


图 1-2 选择用户界面特征

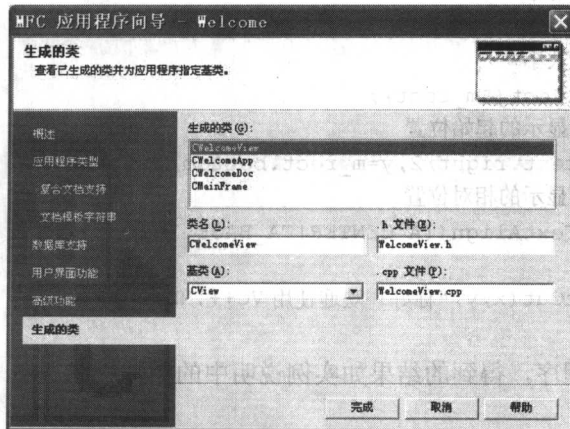


图 1-3 查看生成的类

⑤ 其他选项均采用默认属性，单击【完成】按钮，向导将为应用程序生成必要的框架文件并在工作区窗口中打开生成的项目。

⑥ 为了能够在单文档窗口的客户区中显示文字，需要为 `CWelcomeView` 类的 `OnDraw()` 函数添加代码。

⑦ 至此，程序编写完成，编译并运行它，即可看到“欢迎”界面。

### 技术要点

① 如何获得窗口客户区的大小：`CWnd` 类的成员函数 `GetClientRect()` 用来获得窗口客户区的大小，其原型为：

```
void GetClientRect(LPRECT lpRect) const;
```

其中参数 `lpRect` 为指向结构体 `Rect` 或 `CRect` 对象的指针，该指针存储了客户区的尺寸。

② 如何在客户区中输出文字：`CDC` 类的成员函数 `TextOut()` 用来在客户区中输出文字，该函数的的原型为：

```
BOOL TextOut(int x, int y, const CString& str);
```

其中参数(x,y)指定文本输出的开始位置, 参数 str 为包含输出字符的 CString 对象, 若输出成功, 返回值为 true, 否则返回值为 false。



### 归纳注释

OnDraw()函数中添加的源代码如下:

```
void CWelcomeView::OnDraw(CDC* pDC)
{
    CWelcomeDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    // TODO: 在此处为本机数据添加绘制代码
    //创建矩形对象
    CRect m_rect;
    //获取窗口大小
    GetClientRect(&m_rect);
    //设置文字显示的起始位置
    int x=m_rect.right/2,y=m_rect.bottom/2;
    //设置文字显示的相对位置
    pDC->SetTextAlign(TA_CENTER|TA_BASELINE);
    //显示文字
    pDC->TextOut(x,y,"你好! 欢迎使用 VC++.NET ");
}
```

按 F5 键运行该程序, 得到的结果如实例说明中的图片所示。

## 实例

## 2

## 动态字幕效果



## 实例说明

在本例中，我们利用定时器制作一个可以移动的字幕。当用户在窗口区内右击时，字幕开始移动，单击时，字幕暂停移动。通过本例我们可以学习为应用程序添加消息处理函数、使用定时器以及响应鼠标事件。



## 操作步骤

- ① 启动 Visual C++.NET。在 Windows 的任务栏中选择【开始】|【程序】|Microsoft Visual Studio.NET| Microsoft Visual Studio.NET，调用 Microsoft Visual Studio.NET 的开发环境。选择 MFC 应用程序模板，项目名称为 MoveText。
- ② 在 MFC 应用程序向导中的【应用程序类型】选项中【单文档】，其余选项采用默认值，单击【完成】按钮实现项目的建立。
- ③ 在【类视图】窗口中选择 CMoveTextView 项，然后右击，选择弹出菜单中的【属性】项，调出【属性】窗口。
- ④ 在【属性】窗口的【消息】项下为 CMoveTextView 添加定时器消息 WM\_TIMER 的相应函数 OnTimer()，单击 WM\_LBUTTONDOWN 的处理函数 OnLButtonDown()，右击 WM\_RBUTTONDOWN 的处理函数 OnRButtonDown()，界面分别如图 2-1~图 2-3 所示。

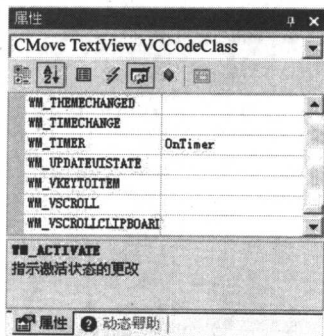
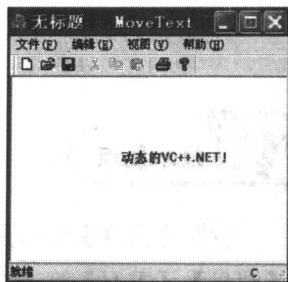


图 2-1 设置响应函数 OnTimer()

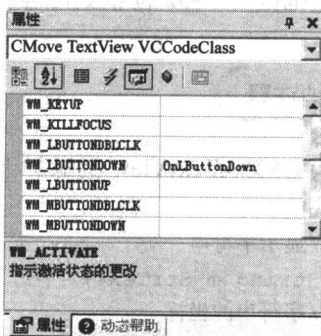


图 2-2 设置处理函数 OnLButtonDown()

- ⑤ 在【属性】窗口的【重写】项下为 CMoveTextView 类重载 OnInitialUpdate()函数。
- ⑥ 相应函数源程序代码的编写。

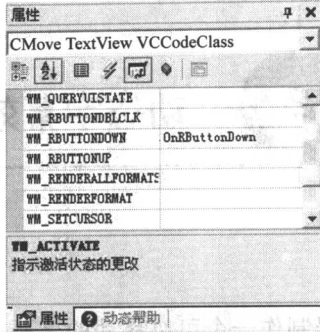


图 2-3 设置处理函数 OnRButtonDown()

**技术要点**

① 【属性】窗口是 Visual C++.NET 的一个新增特性。该窗口包括【属性】、【事件】、【消息】、【重写】等工具栏。这些工具栏中规定了 Windows 程序中各种触发事件、消息和控件的属性。通过该窗口可以为应用程序添加各种功能。

② CDC 类的成员函数 SetTextColor()用于设置字体的颜色，其函数原型如下：

```
Virtual COLORREF SetTextColor(COLORREF crColor);
```

参数 crColor 是用来设置字体颜色的 RGB 值，该函数返回以前字体颜色的 RGB 值。

③ 定时器的使用方法。对于定时器，我们需要了解两个重要的函数 CWnd::SetTimer() 和 CWnd::KillTimer()。CWnd::SetTimer()用来设置一个定时器以指定间隔时间触发，KillTimer()使一个正在运行的定时器停止。定时器以两种方式通知应用程序定时器间隔时间已经到了，发送 WM\_TIMER 消息映射和调用应用程序定义的回调函数。设置定时器最简单的方法是调用 SetTimer()，参数值用一个计时器 ID 和一个超时值，然后把 WM\_TIMER 消息映射到应用程序窗口类的一个 OnTimer()函数中，计时器 ID 是一个标记计时器的非 0 值，传入 SetTimer()的超时值指定了两个 WM\_TIMER 消息之间的间隔，以 1/1000 秒计。该值的取值范围为 1~32 位整数可容纳的最大数字。

**归纳注释**

① 为 CMoveTextView 类添加下述成员变量：

```
Public:
//显示字符串
CString m_StrText;
//字符串的坐标
float m_StartX;
CRect m_Rect;
```

在 CMoveTextView 类的构造函数中初始化显示字符串。

```
CMoveTextView:: CMoveTextView()
```

```

{
    // TODO: 在此处添加构造代码
    m_StrText="动态的 VC++.NET";
}

```

- ② 在 CMoveTextView 类的 OnInitialUpdate() 函数中设置显示字符串的初始坐标。

```

void CMoveTextView::OnInitialUpdate()
{
    CView::OnInitialUpdate();

    // TODO: 在此添加专用代码和/或调用基类
    //获取客户区的大小
    GetClientRect(&m_Rect);
    //设置现实字符串的初始坐标
    m_StartX=m_Rect.left;
}

```

- ③ 编写 CMoveTextView 类的 OnDraw() 函数，用于输出字符串。

```

void CMoveTextView::OnDraw(CDC* pDC)
{
    CtextmoveDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);

    // TODO: 在此处为本机数据添加绘制代码
    //设置字体的颜色
    pDC->SetTextColor( RGB(255,0,0) );
    //在指定位置输出字符串
    pDC->TextOut(m_StartX,80,m_StrText);
}

```

- ④ 编写单击鼠标左键的消息处理函数 OnLButtonDown(), 用于停止定时器。

```

void CMoveTextView::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    //停止计时器
    KillTimer(1);
    CView::OnLButtonDown(nFlags, point);
}

```

- ⑤ 编写单击鼠标右键的消息处理函数 OnRButtonDown(), 用于设置定时器。

```

void CMoveTextView::OnRButtonDown(UINT nFlags, CPoint point)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    //设置定时器的值
    SetTimer(1,500,NULL);
    CView::OnRButtonDown(nFlags, point);
}

```

- ⑥ 编写定时器的消息处理函数 OnTimer(), 用于实现字幕的移动效果。

```

void CMoveTextView::OnTimer(UINT nIDEvent)

```

```
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    //CDC * pDC=GetDC();
    //设置字符串的显示位置
    m_StartX+=60;
    if (m_StartX>=m_Rect.right)
        m_StartX=m_Rect.left;
    //重新绘制视图区
    Invalidate();
    CView::OnTimer(nIDEvent);
}
```

运行该程序，得到的结果如实例说明中的图片所示。



## 实例

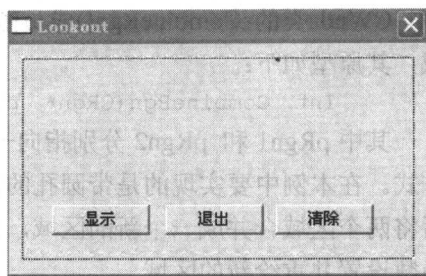
## 3

## 透明窗体



## 实例说明

在本例中，我们创建一个具有透视功能的窗体，在窗体客户区的中心有一圆孔，通过该孔可以看到后面的界面。单击【显示】按钮时，窗体中心出现圆孔，单击【清除】按钮时，窗体恢复正常。本例的重点是学习如何使用 `CombineRgn()` 函数实现带孔的窗体。



## 操作步骤

- ① 启动 Visual C++.NET，选择 MFC 应用程序模板，项目名称为 Lookout。
- ② 在 MFC 应用程序向导的【应用程序类型】选项中选择【基于对话框】，其余选项采用默认值，单击【完成】按钮实现项目的建立。
- ③ 从【资源视图】窗口中调出主对话框资源 `IDD_LOOKOUT_DIALOG`，移去所有控件，并从工具箱中添加三个按钮。
- ④ 右击新添加的按钮，在调出的【属性】窗口中，依次设置各个按钮的 ID，【显示】按钮的 ID 为 `IDC_SHOW`，【清除】按钮的 ID 为 `IDC_CLEAR`，【退出】按钮的 ID 为 `IDC_EXIT`，如图 3-1 所示。

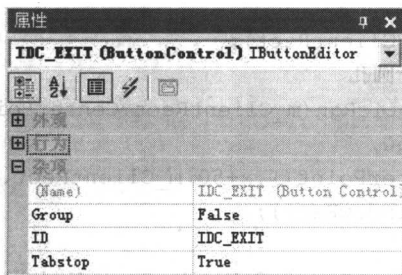


图 3-1 设置按钮的 ID

- ⑤ 在【属性】窗口中，为各按钮添加鼠标单击事件 `BN_CLICKED` 的处理函数，分别为 `OnBnClickedShow()`、`OnBnClickedExit()`、`OnBnClickedClear()`。
- ⑥ 在【显示】按钮的 `OnBnClickedShow()` 函数中添加代码，以实现鼠标单击该按钮时窗口客户区中心出现一圆孔。