

CButtonST 使用技巧

注：本文根据杜修杏在网络上的文章参考写成，加入了自己试验的一些结果。

(一)

[下载CButtonST示例源代码](#)

[下载CButtonST类的源代码](#)

步骤：

- 1、将文件BCMenu.cpp BCMenu.h BtnST.cpp BtnST.h复制到当前工程下
- 2、在当前工程下按“项目/添加现有项”将上述四个文件添加到当前工程
- 3、添加CButtonST对象，用相应的代码实现不同效果

本篇主要包括了以下的效果：

1. [在按钮上加入Icon，使Icon和文字同时显示](#)
2. [显示平面按钮](#)
3. [使按钮上的图标可变](#)
4. [设置按钮在不同状态下的底色和文字颜色](#)
5. [设置图标和文字的位置](#)
6. [根据Icon的形状设置按钮的形状](#)
7. [添加Tooltips](#)

以下是Basic Features的演示界面：



效果一：在按钮上加入 Icon，使 Icon 和文字同时显示

假设按钮 ID 为 IDC_BUTTON1

1. 添加成员变量

```
CButtonST m_btn;
```

2. 添加 Icon 资源，设其 ID 设为 IDI_ICON1

3. 在 OnInitDialog 函数中初始化按钮

```
m_btn.SubclassDlgItem(IDC_BUTTON1, this); 此行删除  
m_btn.SetIcon(IDI_ICON1);  
m_btn.SetFlat(FALSE);
```

注意:为达到最好效果，请根据 Icon 的大小调整按钮的大小。

实例：演示程序中的 Standar 按钮。



效果二：显示平面按钮

只需要在效果一的基础上把 `m_btn.SetFlat(FALSE);` 语句去掉。



效果三：使按钮上的图标可变

假设按钮 ID 为 IDC_BUTTON1

1. 添加成员变量

`CButtonST m_btn;`

2. 添加两个 Icon 资源，ID 设为 IDI_ICON1 和 IDI_ICON2

IDI_ICON1 是普通状态的图标，IDI_ICON2 是按下时的图标

3. 在 OnInitDialog 函数中初始化按钮

```
m_btn.SubclassDlgItem(IDC_BUTTON1, this); 此行删除
m_btn.SetIcon (IDI_ICON2, IDI_ICON1);
```

实例：演示程序中的 Halloween 按钮。



效果四：设置按钮在不同状态下的底色和文字颜色

假设按钮 ID 为 IDC_BUTTON1

1. 添加成员变量

`CButtonST m_btn;`

2. 添加 Icon 资源，设其 ID 设为 IDI_ICON1

3. 在 OnInitDialog 函数中初始化按钮

```
m_btn.SubclassDlgItem(IDC_BUTTON1, this); 此行删除
m_btn.SetIcon (IDI_ICON1);
m_btn.SetColor (CButtonST::BTNST_COLOR_BK_OUT, RGB (208,208,208));
m_btn.SetColor (CButtonST::BTNST_COLOR_BK_FOCUS, RGB (208,208,208));
m_btn.OffsetColor (CButtonST::BTNST_COLOR_BK_IN, 30);
```

提示：SetColor 函数和 OffsetColor 函数的第一个参数表示按钮的各个状态，前景色就是文字的颜色，它们的取值表示：

`BTNST_COLOR_BK_IN` // 鼠标放在按钮内时的背景色

`BTNST_COLOR_FG_IN` // 鼠标放在按钮内时的前景色

`BTNST_COLOR_BK_OUT` // 鼠标不在按钮内时的背景色

`BTNST_COLOR_FG_OUT` // 鼠标不在按钮内时的前景色

`BTNST_COLOR_BK_FOCUS` // 焦点位于按钮上的背景色

`BTNST_COLOR_FG_FOCUS` // 焦点位于按钮上的前景色

注：在试验中 `BTNST_COLOR_FG_FOCUS` 不起作用。

实例：演示程序中的 48×48 icon 按钮、Zip 按钮等。



效果五：设置图标和文字的位置

假设按钮 ID 为 IDC_BUTTON1

1. 添加成员变量

`CButtonST m_btn;`

2. 添加 Icon 资源，ID 设为 IDI_ICON1

3.在 OnInitDialog 函数中初始化按钮

```
m_btn.SubclassDlgItem(IDC_BUTTON1,this);  
m_btn.SetIcon(IDI_ICON1);  
m_btn.SetAlign(CButtonST::ST_ALIGN_VERT);
```


提示: SetAlign 函数的第一个参数表示位置信息, 缺省情况下, 文字在图标右边

ST_ALIGN_HORIZ // 文字在右

ST_ALIGN_VERT // 文字在下

ST_ALIGN_HORIZ_RIGHT // 文字在左

实例: 演示程序中的 Search 按钮。

 效果六: 根据 Icon 的形状设置按钮的形状

假设按钮 ID 为 IDC_BUTTON1

1.添加成员变量

```
CButtonST m_btn;
```

2.添加 Icon 资源, 设其 ID 设为 IDI_ICON1

3.在 OnInitDialog 函数中初始化按钮

```
m_btn.SubclassDlgItem(IDC_BUTTON1,this);  
m_btn.SetIcon (IDI_ICON1);  
m_btn.DrawBorder (FALSE);  
m_btn.SetColor (CButtonST::BTNST_COLOR_BK_OUT, RGB(208,208,208));  
m_btn.SetColor (CButtonST::BTNST_COLOR_BK_IN, RGB(208,208,208));  
m_btn.SetColor (CButtonST::BTNST_COLOR_BK_FOCUS, RGB(208,208,208));
```

注意: 为达到最好效果, 请根据 Icon 的大小调整按钮的大小。
并根据实际情况设置背景颜色。请事先将按钮的文字去掉。

 效果七: 添加 Tooltips

假设按钮 ID 为 IDC_BUTTON1

1.添加成员变量

```
CButtonST m_btn;
```

2.添加 Icon 资源, 设其 ID 设为 IDI_ICON1

3. 在 OnInitDialog 函数中初始化按钮

```
m_btn.SubclassDlgItem(IDC_BUTTON1,this);  
m_btn.SetIcon (IDI_ICON1);  
m_btn.SetTooltipText(_T("This is a tooltip.\r\nMultiline!"));
```

(二)

本篇主要包括了以下的效果:

出自: <http://www.vckbase.com/document/viewdoc/?id=519>

1. [改变鼠标进入按钮时的形状\(超链接效果\)](#)

2. [菜单按钮](#)

3. [位图按钮](#)
4. [按钮焦点](#)
5. [CheckBox按钮](#)
6. [透明按钮\(按钮背景和窗口背景一样\)](#)
7. [特殊的PictureBox](#)

 效果一：改变鼠标进入按钮时的形状(超链接效果)

假设按钮ID为IDC_BUTTON1


1. 添加成员变量

CButtonST **m_btn;**

2. 添加 Icon 资源，设其 ID 设为 IDI_ICON1

3. 在 OnInitDialog 函数中初始化按钮

```
m_btn.SubclassDlgItem(IDC_BUTTON1,this);  
m_btn.SetIcon (IDI_ICON1);  
m_btn.OffsetColor (CButtonST::BTNST_COLOR_BK_IN, 30);  
m_btn.SetURL ("www.scut.edu.cn");  
m_btn.SetToolTipText ("www.scut.edu.cn");  
m_btn.SetBtnCursor (IDC_CURSOR1);
```

 效果二：菜单按钮

假设按钮 ID 为 IDC_BUTTON1

1. 添加成员变量

CButtonST **m_btn;**

2. 添加 Icon 资源，设其 ID 设为 IDI_ICON1

3. 添加 Menu，ID 设为 IDR_MENU

4. 在 OnInitDialog 函数中初始化按钮

```
m_btn.SubclassDlgItem(IDC_BUTTON1,this);  
m_btn.SetIcon(IDI_ICON1);  
m_btn.OffsetColor(CButtonST::BTNST_COLOR_BK_IN, 30);  
m_btn.SetMenu(IDR_MENU, m_hWnd);
```

注意:菜单的效果类似与右键菜单，所有在设置菜单选项时只设置一列就够了。另外菜单的宽度与菜单的文字长度有关，可以用空格占位的办法以达到最佳效果。

在菜单中引入位图的方法


1. 添加 ToolBar，依次使用 Menu 菜单项 ID 作为 ToolBar 按钮的 ID。

2. ToolBar 上的按钮图标将显示在相应的菜单项中。

将上面代码中的 **m_btn.SetMenu(IDR_MENU, m_hWnd);**

改为 **m_btn.SetMenu(IDR_MENU, m_hWnd, TRUE, IDR_TOOLBAR);**

IDR_TOOLBAR 为相应 ToolBar 的 ID。

 效果三：位图按钮

假设按钮 ID 为 IDC_BUTTON1

1.添加成员变量


CButtonST m_btn;

2.添加位图资源，ID 设为 IDB_BITMAP1

3.在 OnInitDialog 函数中初始化按钮

```
m_btn.SubclassDlgItem(IDC_BUTTON1,this);  
m_btn.SetBitmaps (IDB_BITMAP1,RGB(0,0,0));  
m_btn.OffsetColor (CButtonST::BTNST_COLOR_BK_IN, 30);
```

提示: 上面的 SetBitmaps 函数会将图片中颜色值为 RGB(0,0,0)的点设为透明。

效果四：按钮焦点

假设按钮 ID 为 IDC_BUTTON1

1.添加成员变量

CButtonST m_btn;

2. 添加 Icon 资源，设其 ID 设为 IDI_ICON1

3. 在 OnInitDialog 函数中初始化按钮

```
m_btn.SubclassDlgItem(IDC_BUTTON1,this);  
m_btn.SetIcon (IDI_ICON1);  
m_btn.SetAlign (CButtonST::ST_ALIGN_VERT);  
m_btn.DrawFlatFocus (TRUE);
```

效果五：CheckBox 按钮

假设按钮 ID 为 IDC_BUTTON1

1.添加成员变量

CButtonST m_btn;

2. 添加 Icon 资源，ID 设为 IDI_ICON1 和 IDI_ICON2


3.在 OnInitDialog 函数中初始化按钮

```
m_btn.SubclassDlgItem (IDC_CHECK1,this); 待测试  
m_btn.SetIcon (IDI_ICON1,IDI_ICON2);  
m_btn.OffsetColor (CButtonST::BTNST_COLOR_BK_IN, 30);
```

提示:

IDI_ICON1 为选中时候显示的图标

IDI_ICON2 为为选中时候的图标

效果六：透明按钮

假设按钮 ID 为 IDC_BUTTON1

1.添加成员变量

CButtonST m_btn;

2. 添加 Icon 资源，ID 设为 IDI_ICON1

3.在 OnInitDialog 函数中初始化按钮

```
m_btn.SubclassDlgItem (IDC_BUTTON1,this);  
m_btn.SetIcon (IDI_ICON1);
```

```
m_btn.DrawTransparent(TRUE);
```

效果七：特殊的 PictureBox

利用 CButtonST 的特殊显示风格，完全可以把它当作一个 PictureBox 控件使用。这时一般会把按钮的属性设置为 disable。

实例：演示程序的 About 页中使用了 CButtonST 来代替 PictureBox。

（三）

本篇将向大家介绍两个 CButtonST 的派生类。

一、CWinXPButtonST 类

CWinXPButtonST 类 是一个 CButtonST 的派生类。它的特点是使用了 Windows XP 中的圆角、黑色边框的风格。它包括 WinXPButtonST.h 和 WinXPButtonST.cpp 两个文件，而由于它是从 CButtonST 派生而来的，所以在引用它时,也要一并引入 CButtonST 的文件。

使用：

1. 引入文件，添加文件
2. 导入文件后，在 SdtAfx.h 文件中添加

```
#include "WinXPButtonST.h"
```

3. 添加成员变量

```
CWinXPButtonST m_btn2;
```

4. 假设按钮 ID 为 IDC_BUTTON

添加 Icon 资源，ID 设为 IDI_ICON1

5. 在 OnInitDialog 函数中初始化按钮

```
m_btn2.SubclassDlgItem(IDC_BUTTON,this);  
m_btn2.SetIcon(IDI_ICON1);  
m_btn2.SetRounded(TRUE);
```

注意：但是它存在一个不足的地方，就是按钮的圆角是画出来的，按钮的形状仍然是原来的矩形。所以如果在有背景色的窗口中使用该按钮的话就会显得相当碍眼。

如图所示，在黑色的背景色中显示出难看的尖角：



解决的办法是通过 SetColor 函数使得按钮的底色与窗口的底色一致。

二、CShadeButtonST 类

CShadeButtonST 类 是 CButtonST 的另一个派生类。它的特点是支持更为多样化的背景，通过 SetShade 函数可以为按钮设置 9 种不同的背景效果。它包括四个文件，它们分别是

CeXDib.h、CeXDib.cpp、ShadeButtonST.h 和 ShadeButtonST.cpp。在引用它时,同样要一并引入 CButtonST 的文件。

使用:

1.引入文件, 添加文件

2.导入文件后, 在 SdtAfx.h 文件中添加

```
#include "ShadeButtonST.h"
```

3.添加成员变量

```
CShadeButtonST m_btn3;
```

4.假设按钮 ID 为 IDC_BUTTON3

添加 Icon 资源, ID 设为 IDI_ICON1

5.在 OnInitDialog 函数中初始化按钮

```
m_btn3.SubclassDlgItem(IDC_BUTTON3,this);  
m_btn3.SetIcon(IDI_ICON1);  
m_btn3.SetShade(CShadeButtonST::SHS_METAL);
```

下面给出各种背景效果和对应的参数值。

