

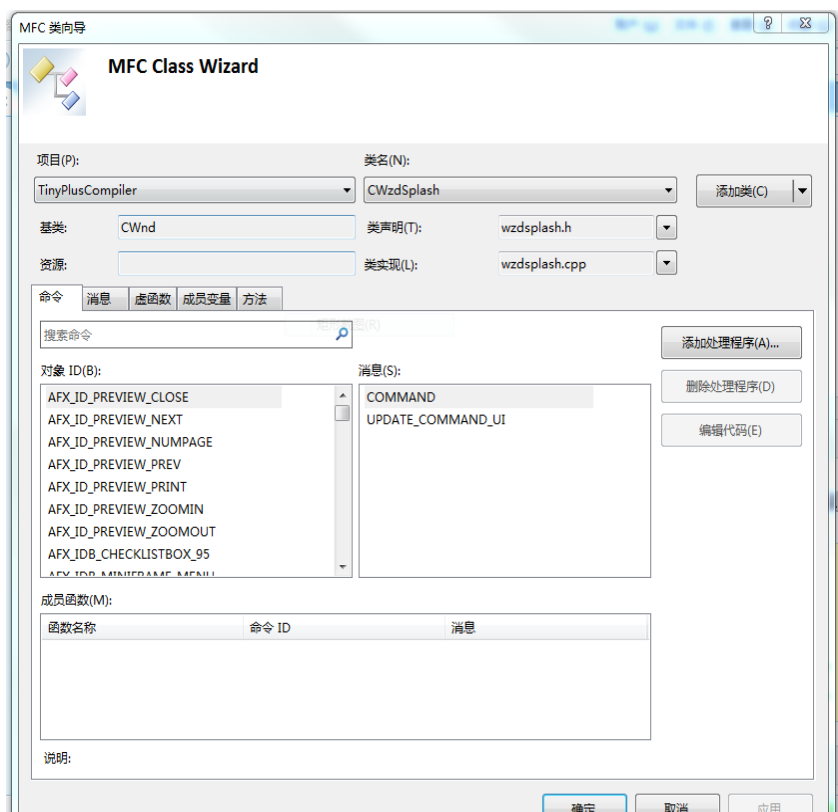
## 转 给 VS2008 / VS2010 MFC 项目添加启动画面

2013年10月15日 19:15:12 阅读数：5345

在VC6.0时代，我们通过在MFC工程中插入Splash组件来简单方便的实现这个功能，但在VS2008以后就没有了这个功能，所以如果想给自己的程序添加一个显示Logo的启动画面，就需要自己添加代码来实现，参考了下VC6.0中这个Splash组件添加后的代码，就很容易在自己的项目里实现这个功能。如下图。



首先，在菜单的“项目”中选择“类向导”，生成一个启动画面的Splash类。



然后，实现代码主要是在CWzdSplash类的头文件和cpp文件中实现。在CWzdSplash类的头文件中我们添加下面的代码：

```
[cpp]
1. #pragma once
2. // CWzdSplash
3. class CWzdSplash : public CWnd
4. {
5.     DECLARE_DYNAMIC(CWzdSplash)
6. public:
7.     CWzdSplash();
8.     virtual ~CWzdSplash();
9. protected:
10.     DECLARE_MESSAGE_MAP()
11. public:
12.     CBitmap m_bitmap;
13.     // void Create(void);
14.     void Create(UINT nBitmapID);
15.     afx_msg void OnPaint();
16.     afx_msg void OnTimer(UINT_PTR nIDEvent);
17. };
```

在CwzdSplash类的cpp文件中我们添加下面的代码：

```
[cpp]
1. // WzdSplash.cpp : 实现文件
2. //
3. #include "stdafx.h"
4. #include "TinyPlusCompiler.h"
5. #include "WzdSplash.h"
6. // CWzdSplash
7. IMPLEMENT_DYNAMIC(CWzdSplash, CWnd)
8. CWzdSplash::CWzdSplash()
9. {
10. }
11. CWzdSplash::~CWzdSplash()
12. {
13. }
14. BEGIN_MESSAGE_MAP(CWzdSplash, CWnd)
15.     ON_WM_PAINT()
16.     ON_WM_TIMER()
17. END_MESSAGE_MAP()
18. // CWzdSplash 消息处理程序
19. void CWzdSplash::Create(UINT nBitmapID)
20. {
21.     m_bitmap.LoadBitmap(nBitmapID);
22.     BITMAP bitmap;
23.     m_bitmap.GetBitmap(&bitmap);
24.     //CreateEx(0,AfxRegisterWndClass(0),L"WS_POPUP|WS_VISIBLE|WS_BORDER,0,0,bitmap.bmWidth,bitmap.bmHeight,NULL,0);
25.     CreateEx(0,
26.         AfxRegisterWndClass(0, AfxGetApp()->LoadStandardCursor(IDC_ARROW)),
27.         NULL, WS_POPUP | WS_VISIBLE, 0, 0, bitmap.bmWidth, bitmap.bmHeight, NULL, NULL);
28. }
29. void CWzdSplash::OnPaint()
30. {
31.     // TODO: 在此处添加消息处理程序代码
32.     // 不为绘图消息调用 CWnd::OnPaint()
33.     CPaintDC dc(this); // device context for painting
34.     BITMAP bitmap;
35.     m_bitmap.GetBitmap(&bitmap);
36.     CDC dcComp;
37.     dcComp.CreateCompatibleDC(&dc);
38.     dcComp.SelectObject(&m_bitmap);
39.     // draw bitmap
40.     dc.BitBlt(0,0,bitmap.bmWidth,bitmap.bmHeight,&dcComp,0,0,SRCCOPY);
41. }
42. void CWzdSplash::OnTimer(UINT_PTR nIDEvent)
43. {
44.     // TODO: 在此添加消息处理程序代码和/或调用默认值
45.     //CWnd::OnTimer(nIDEvent);
46.     DestroyWindow(); //销毁初始画面窗口
47. }
```

最后在项目的主窗口视图的Create函数中添加下面的实例化过程就可以显示出我们自定义的Logo启动画面了：

```
[cpp]
1. //////////////显示Splash, 2010-10-15//////////////////////////
2. CWzdSplash wndSplash; //创建启动窗口类的实例
3. wndSplash.Create(IDB_SPLASH);
4. wndSplash.CenterWindow();
5. wndSplash.UpdateWindow(); //send WM_PAINT
6. Sleep(1500);
7. wndSplash.DestroyWindow(); //销毁初始画面窗口
```

注明一下要注意的地方：wndSplash.Create(IDB\_SPLASH);中的IDB\_SPLASH是在添加位图资源时添加进来的资源ID,自己可以定义自己所喜欢的启动图画。

如果不使用Sleep()函数让线程休眠几秒，我们很难看到这个启动画面，但这样存在一个问题，让线程休眠不是一个好的办法，感觉应该用定时器去实现这功能的，但不是很熟悉定时器的使用，研究一下再把它改过来。

原文地址：<http://blog.csdn.net/libenqing/article/details/5961991> #

文章标签：[vs2008](#) [动画](#) [mfc](#) [启动](#)

个人分类：[纯编程](#)