

控件的使用

发送消息

`SendMessage` 阻塞且消息不进入消息队列，消息处理完才返回

`PostMessage` 非阻塞且消息进入消息队列，直接返回

控件消息

当控件自身状态改变的时候会向父窗口发送 `WM_COMMAND` 消息(例如按钮的 `BN_XXX`)，通过向控件发送消息(例如按钮的 `BM_XXX`)来控制控件的状态属性

WM_COMMAND消息

消息源	wParam (high word)	wParam (low word)	lParam
菜单	0	菜单ID	0
快捷键	1	快捷键ID	0
控件	控件通知码	控件ID	控件窗口句柄

MFC

微软基础类库（英语：Microsoft Foundation Classes，简称MFC）是微软公司提供的一个类库（class libraries），以C++类的形式封装了Windows API，并且包含一个应用程序框架，以减少应用程序开发人员的工作量。其中包含大量Windows句柄封装类和很多Windows的内建控件和组件的封装类。

入口函数分析

1. `theApp` 全局对象创建，`CMyWinApp::CMyWinApp` 及其父类
2. 进入 `WinMain` 函数，通过 `CWinApp` 对象的指针（指向的是全局对象 `theApp` 的地址）调用 `InitInstance`，`CMyWinApp::InitInstance` 被调用
3. 在 `CMyWinApp::InitInstance` 中创建 `CMyWnd` 对象，触发 `CMyWnd::CMyWnd` 调用，在其中完成 `CreateWindow`
4. `CMyWnd` 创建完后，在 `CMyWinApp::InitInstance` 通过指针调用其 `ShowWindow` 函数，窗口出现

MFC的入口函数为 `_twinMain`

```
extern "C" int WINAPI
_tWinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
_In_ LPTSTR lpCmdLine, int nCmdShow)
#pragma warning(suppress: 4985)
{
    // call shared/exported WinMain
    return AfxWinMain(hInstance, hPrevInstance, lpCmdLine, nCmdShow);
}
```

在 `AfxWinMain` 中创建了应用程序实例

```
int AFXAPI AfxWinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
_In_ LPTSTR lpCmdLine, int nCmdShow)
{
    ASSERT(hPrevInstance == NULL);

    int nReturnCode = -1;
    CWinThread* pThread = AfxGetThread();
    CWinApp* pApp = AfxGetApp();

    // AFX internal initialization
    if (!AfxWinInit(hInstance, hPrevInstance, lpCmdLine, nCmdShow))
        goto InitFailure;

    // App global initializations (rare)
    if (pApp != NULL && !pApp->InitApplication())
        goto InitFailure;

    // Perform specific initializations
    if (!pThread->InitInstance())
    {
        if (pThread->m_pMainWnd != NULL)
        {
            TRACE(traceAppMsg, 0, "Warning: Destroying non-NULL m_pMainWnd\n");
            pThread->m_pMainWnd->DestroyWindow();
        }
        nReturnCode = pThread->ExitInstance();
        goto InitFailure;
    }
    nReturnCode = pThread->Run();
}
```

这里获取到的是 `Ctest2App theApp` 对象

```
// 唯一的 Ctest2App 对象
Ctest2App theApp;
```

`pThread->InitInstance()` 即为 `theApp.InitInstance()`，初始化了公共控件

```
INITCOMMONCONTROLSEX InitCtrls;
InitCtrls.dwSize = sizeof(InitCtrls);
// 将它设置为包括所有要在应用程序中使用的
// 公共控件类。
InitCtrls.dwICC = ICC_WIN95_CLASSES;
InitCommonControlsEx(&InitCtrls);

CWinApp::InitInstance();
```

然后在创建子窗口

名称
test2.exe!Ctest2Doc::Ctest2Doc() 行 31
test2.exe!Ctest2Doc::CreateObject() 行 22
test2.exe!CRuntimeClass::CreateObject() 行 142
test2.exe!CDocTemplate::CreateNewDocument() 行 245
test2.exe!CSingleDocTemplate::OpenDocumentFile(const wchar_t * lpszPath...
test2.exe!CSingleDocTemplate::OpenDocumentFile(const wchar_t * lpszPath...
test2.exe!CDocManager::OnFileNew() 行 912
test2.exe!CWinApp::OnFileNew() 行 21
test2.exe!_AfxDispatchCmdMsg(CCmdTarget * pTarget, unsigned int nID, int ...
test2.exe!CCmdTarget::OnCmdMsg(unsigned int nID, int nCode, void * pExtr...
test2.exe!CWinApp::ProcessShellCommand(CCommandLineInfo & rCmdInf..

然后显示更新窗口

```
// 唯一的一个窗口已初始化，因此显示它并对其进行更新
m_pMainWnd->ShowWindow(SW_SHOW); 已用时间 <= 54ms m_
m_pMainWnd->UpdateWindow();
return TRUE;
```