前台线程与后台线程,AfxGetApp->GetMainWnd()与AfxGetMainWnd的不同

　　顾名思义，前台指看得见，既然看得见自然要有窗口，而后台是看不见，看不见可以理解为没有窗口，既然没有窗口，那么用AfxGetMainWnd()取得其窗口句柄，未免有些搞笑，为什么呢？聪明的你想必早已明白，如果你曾经跟踪过AfxGetMainWnd()的调用，会发现它取得的是AFX\_MODULE\_THREAD\_STATE线程模块中保存的活动线程的窗口句柄，而后台线程既然没有窗口，那你又从何而言取得窗口句柄呢，(也许有人对后台的理解是不显示窗口，就算有窗口，只要不显示就是后台线程,严格的说不是这样，窗口主要用来与用户交互，有窗口就难免阻塞，而后台线程往往用来进行一些后面的运算或处理，是通过前台传递的信息或数据进行相应运做的,如果窗口隐藏谈何信息传递呢？况又不能像其它线程那样进行后台运做．不过你如此理解也无防 ，关键是明白有什么区别，及怎么用．）  
AfxGetApp()->GetMainWnd()取得的是主窗口句柄,无论在那个线程里调用都是没有问题的,因为它先取得主线程句柄,再取得主线程的活动窗口(如视图切换可能导致的替代，这种情况我也不大清楚），如果没有活动窗口则取主窗口,任何程序总要有一个主窗口,所以它的调用不会出现问题,如果想取得程序的主窗口建议用AfxGetApp()->GetMainWnd().

　　要注意的是控制台程序没有窗口，其窗口句柄始终为0．其次后台线程其实就是一个控制台式的程序，只不过不是主线程罢了．还有就是用API Createthread之类的函数创建的线程无法生成CWinThread对象．所以如果想使用CWinThread对象里的函数，以及一些全局函数进行操作，如上面的AfxGetMainWnd()，就必须用CWinThread对象的CreateThread函数，或用AfxBeginThread()函数创建线程，否则很可能出现错误，因为MFC对线程的管理是通过CWinThread对象来完成的，通过下面的跟踪代码你不难看出． 　记得当初有一位前辈曾提到过这个问题，只是不知它是否理解这其中的原因

以下是跟踪代码:

\_AFXWIN\_INLINE CWnd\* AFXAPI AfxGetMainWnd()  
{ CWinThread\* pThread = AfxGetThread();  
        return pThread != NULL ? pThread->GetMainWnd() : NULL; }

CWinThread\* AFXAPI AfxGetThread()  
{  
// check for current thread in module thread state  
AFX\_MODULE\_THREAD\_STATE\* pState = AfxGetModuleThreadState();  
CWinThread\* pThread = pState->m\_pCurrentWinThread;

// if no CWinThread for the module, then use the global app  
if (pThread == NULL)  
        pThread = AfxGetApp();

return pThread;  
}

可以看出当取得的线程句柄为空时,则返回主线程句柄，所以在主线程里调用AfxGetMainWnd()是一定没问题的，（控制台程序除外）

\_AFXWIN\_INLINE CWinApp\* AFXAPI AfxGetApp()  
{ return afxCurrentWinApp; }

CWnd\* CWinThread::GetMainWnd()  
{  
if (m\_pActiveWnd != NULL)  
        return m\_pActiveWnd;          // probably in-place active

// when not inplace active, just return main window  
if (m\_pMainWnd != NULL)  
        return m\_pMainWnd;

return CWnd::GetActiveWindow();  
}

那个afxCurrentWinApp就不用说了吧．

嘿嘿,一切完毕,如果你的程序存在类似上面的问题可一定要改回来噢.