# 转换到 COFF 期间失败: 文件无效或损坏

方案一：（这个方法比较好，在用qt运行时出现问题也能解决）

复制 C:\Windows\winsxs\x86\_netfx-cvtres\_for\_vc\_and\_vb\_b03f5f7f11d50a3a\_6.1.7601.17514\_none\_ba1c770af0b2031b 目录下的 cvtres.exe 到C:\Program Files\Microsoft Visual Studio 10.0\VC\bin 目录下，直接覆盖替换。

# MFC

## 隐藏对话框,当后台服务

//在初始化对话框中使用

SetTimer(0, 1, NULL);//运行后1ms就隐藏，连任务栏上的 图标都没有了

......

//在定时器函数中写如下代码

ShowWindow(SW\_HIDE);

KillTimer(0);

## 登陆vnc

void login(CString ip,CString pwd)

{

ShellExecute(::GetDesktopWindow(),\_T("open"),

\_T("C:/desktop/vnc-4\_1\_2-x86\_win32\_viewer.exe"),NULL,NULL,

SW\_NORMAL);

HWND h=NULL;

HWND hip=NULL;

while(h==NULL)

h=::FindWindowA(NULL,"VNC Viewer : Connection Details");

while(hip==NULL)

hip=::FindWindowExA(h,NULL,"ComboBox",NULL);

//清?空?

::SendMessage(hip, CB\_RESETCONTENT, NULL, NULL);

//ip

xlib::xSendInputString(hip,ip);

//ok

HWND hok=::FindWindowExA(h,NULL,"Button","OK");

//按ã¡ä钮£¤上¦?按ã¡ä空?格?相¨¤当Ì¡À于®¨²按ã¡ä按ã¡ä钮£¤

xlib::xSendAnjian(hok,VK\_SPACE);

HWND hpwd=NULL;

while(hpwd==NULL)

hpwd=::FindWindowA(NULL,"VNC Viewer : Authentication [No Encryption]");

HWND hedituser=NULL;

while(hedituser==NULL)

hedituser=::FindWindowExA(hpwd,NULL,"Edit",NULL);

//找¨°用®?户¡ì名?的Ì?下?一°?个?就¨ª对?了¢?

HWND heditpwd=::FindWindowExA(hpwd,hedituser,"Edit",NULL);

xlib::xSendInputString(heditpwd,pwd);

//登Ì?陆?

HWND hok2=::FindWindowExA(hpwd,NULL,"Button","OK");

xlib::xSendAnjian(hok2,VK\_SPACE);

}

void CvncDlg::OnBnClickedButton1()

{

login("192.168.43.92:2","sjcsfwq");

login("192.168.43.129:2","sjcsfwq");

}

## 子线程调用控件

### 发送消息

//创建线程的时候，可以这样

AfxBeginThreadProc(MyThreadProc, (LPVOID)GetSafeHwnd());

//在你的线程函数中

UINT MyThreadProc( LPVOID pParam )

{

    HWND hWnd= (HWND)pParam;

    ...

    SendMessage(hWnd, UM\_MSG, ...,...); // 发送自定义的消息到主界面线程，通知它去更新界面控件信息

    return 0;

}

afx\_msg LRESULT OnHandleMsg(WPARAM wParam, LPARAM lParam);// 自定义消息UM\_MSG的消息处理函数

BEGIN\_MESSAGE\_MAP(CXXDlg, CDialog)

...

ON\_MESSAGE(UM\_MSG, &CXXDlg::OnHandleMsg)

END\_MESSAGE\_MAP()

LRESULT CXXDlg::OnHandleMsg(WPARAM wParam, lParam)

{

 ...

 SetDlgItemText(...);

 return 0;

}

## 用户界面线程

通常把用于处理用户输入、响应事件和消息的线程称为用户界面线程，而那些不与用户界面作任何交互活动的线程称为工作者线程。

界面线程包含一个消息循环，并关联一个主窗口。消息循环为主窗口及其子窗口分发消息，进行消息驱动。从而，主窗口可以可以响应用户的交互操作以及其他的系统事件。

要创建一个用户界面线程，必须要：  
·重载CWinThread类，建立自己的线程类。  
·建立自己的窗口类，以作为界面线程的主窗口。  
·重载线程类的InitInstance函数，并在其中注册自己的窗口类，创建线程主窗口。  
·在InitInstance函数的最后 return TRUE; 以进入消息循环。  
启动一个用户界面函数，通常使用AfxBeginThread(RUNTIME\_CLASS(线程类名))，请注意与工作者线程的不同之处。  
结束一个用户界面函数，可以通过向线程主窗口发送WM\_DESTROY消息。  
  
其他的工作，就是在线程主窗口内的事情了。

实例操作：  
1.创建MFC对话框程序，添加一个启动用户界面线程的按钮  
2.向程序中添加一个对话框，并建立相应的对话框类CThreadDlg，该对话框将被作为用户界面线程的主窗口。  
3.向程序中添加一个以CWinThread为基类的新类CUIThread，该类用于启动一个用户界面线程。  
4.在UIThread.cpp中加入#include "ThreadDlg.h"，并在CUIThread::InitInstance()中加入  
　　CThreadDlg dlg;  
　　m\_pMainWnd=&dlg;  
　　dlg.DoModal();  
　　确保InitInstance函数最后return TRUE;  
5.CMyThread::InitInstance()中创建的CThreadDlg将与主窗口在独立的线程中运行，可以在CThreadDlg中加入各种执行耗时任务的代码而不会影响主窗口的运行。  
6.在主窗口的\*\*dlg.h加入#include "UIThread.h"，双击按钮添加CWinThread \*pThread=AfxBeginThread(RUNTIME\_CLASS(CUIThread));开启线程。

注意   
■   工作者线程倾向于琐碎的处理，与它不同的是，用户界面线程具有自己的界面而且实际上类似于运行其他应用程序。创建线程而不是其他应用程序的好处是线程可与应用程序共享程序空间，这样可以简化线程与应用程序共享数据的功能。   
■   典型情况是用户界面线程用于完成查询和替换等功能，或者是其他不希望占用主应用程序大量处理时间但是需要一个界面的功能或服务，或者用户也可完全不考虑界面，将这种类型的线程用于窗口消息服务器作为一种传递其消息的方式，以避免使自己因占用处理时间过多而陷入困境。   
■   在时间要求严格的应用程序(例如实时应用程序)中，不希望因为工作者线程启动而等待，这时可将工作者线程中的控制逻辑内置到用户界面线程中并提前创建线程。当需要处理事务时，向用户界面线程发送消息，此时用户界面线程已经运行并且在等待指令。

## 基础知识

### 整型等类型转字符串

Int i=3232;

CString str;str.Format(“%d”,i);

### 对话框

### 打开对话框等

创建打开文件对话框：

CFileDialog dlg(TRUE,//TRUE是创建打开文件对话框，FALSE则创建的是保存文件对话框

".txt",//默认的打开文件的类型

NULL,//默认打开的文件名

OFN\_HIDEREADONLY | OFN\_OVERWRITEPROMPT,//打开只读文件

"文本文件(\*.txt)|\*.txt|所有文件 (\*.\*)|\*.\*||");//所有可以打开的文件类型

2.点打开文件对话框上面的确定键后

if(dlg.DoModal()==IDOK)

{

CString m\_FilePath = dlg.GetPathName();////////取出文件路径

CString m\_path;

m\_path=m\_FilePath;//将文件的路径放入m\_path

UpdateData(FALSE);

}

3.打开文件：File.Open(m\_path,CFile::modeRead);

4.逐行读取文件：CStdioFile File;///可以逐行读文件的类

CString strLine;

while(File.ReadString(strLine)) //////将每行都放进strLine字符串里

{

AfxMessgeBox(strLine);

}

5判断读出来的字：

strLine=“1|2|3|”;//要判断的字符串

int strIndex1 = strLine.Find('|');//在字符串中寻找“|”

CString a[11];

if(-1 != strIndex1)//只要找到“|”就不会返回-1

{

int i=0;

while( -1 != strIndex1)//

{

strIndex1 = strLine.Find('|');

a[i] = strLine.Left(strIndex1);

strLine = strLine.Right(strLine.GetLength() - strIndex1-1) ;

i++;

if (i > 10)//退出循环

break;

}

}

## Mfc调用C#winform控件

对话框上放一个静态文本

另外一个头文件 manage.h

#ifndef \_\_MANAGED\_CODE\_H\_\_

#define \_\_MANAGED\_CODE\_H\_\_

#include <afxwinforms.h>// MFC Windows Forms support

#using <system.dll>

#using <Microsoft.VisualC.Dll>

#using <System.Drawing.dll>

#using <System.Windows.Forms.dll>

#using <mscorlib.dll>

#using "..\\imgcontroldll\\bin\\Debug\\imgcontroldll.dll"

using namespace System::Windows::Forms;

using namespace System::ComponentModel;

using namespace Microsoft::VisualC::MFC;

using namespace System::Drawing;

using namespace System;

using namespace System::Runtime::InteropServices;

using namespace imgcontroldll;

//using namespace stdcli::language;

#endif //\_\_MANAGED\_CODE\_H\_\_

Dlg.h

CWinFormsControl<imgcontroldll::img> xlfd;

Dlg.cpp

void CmfccallDlg::DoDataExchange(CDataExchange\* pDX)

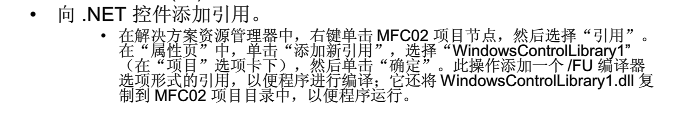
{

CDialogEx::DoDataExchange(pDX);

//DDX\_Control(pDX, IDC\_sss, xlfd);

Microsoft::VisualC::MFC::DDX\_ManagedControl(pDX, IDC\_sss, xlfd);

}



属性，dll，公共语言运行支持clr

## 文件相关

### 浏览新建创建文件夹

CFileDialog dlg(FALSE,//TRUE是创建打开文件对话框，FALSE则创建的是保存文件对话框

"",//默认的保存文件的类型

NULL,//默认打开的文件名

OFN\_HIDEREADONLY | OFN\_OVERWRITEPROMPT,//打开只读文件

"文件夹||所有文件 (\*.\*)|\*.\*||");//所有可以打开的文件类型

if(dlg.DoModal()==IDOK)

{

CString m\_FilePath = dlg.GetPathName();////////取出文件路径

xlib::xCreateDir(m\_FilePath);

}

### 创建目录

方法一、CreateDirectory函数一般用这个就行

查了下MSDN，发现该函数用起来十分方便

BOOL CreateDirectory(

LPCTSTR lpPathName, // pointer to directory path string

LPSECURITY\_ATTRIBUTES lpSecurityAttributes // pointer to security descriptor

);

第一个参数值为文件夹名称，第二个参数值为安全属性，一般设置为NULL即可。如果正确创建，返回值为1，如果没有正常创建文件夹，则返回0。

方法二、mkdir函数

MSDN里面是这样说的，

public boolean mkdir ();

Creates a directory whose path is specified by the ClientStoreFile object.

Return Value:

Returns true if the directory could be created; otherwise, returns false.

试了一下，还管用，要包含头文件"direct.h"，但是返回值好像不太对，我测试的结果表明，“0”表示正确创建了文件夹，而“-1”表示没有正确创建该文件夹。

方法三、用命令行创建

在VC中嵌套dos指令的方法来实现，调用system函数

可以使用system("mkdir \"foldername\"");

或者使用system("md \"foldername\"");

或者干脆写一段命令行的小代码，然后在这里调用，使用此种方法就得注意了，程序在执行时会弹出一个黑白窗口，一闪而过。这种代码实现非常灵活，而且有时候可以解决很多实际问题。

### 打开文件

CFileDialog dlg(TRUE,//TRUE是创建打开文件对话框，FALSE则创建的是保存文件对话框

".txt",//默认的打开文件的类型

NULL,//默认打开的文件名

OFN\_HIDEREADONLY | OFN\_OVERWRITEPROMPT,//打开只读文件

"文本文件(\*.txt)|\*.txt|所有文件 (\*.\*)|\*.\*||");//所有可以打开的

文件类型

if(dlg.DoModal()==IDOK)

{

CString m\_FilePath = dlg.GetPathName();////////取出文件路径

CString m\_path;

m\_path=m\_FilePath;//将文件的路径放入m\_path

UpdateData(FALSE);

CStdioFile File;///可以逐行读文件的类

CString strLine;

File.Open(m\_path,CFile::modeRead);

while(File.ReadString(strLine)) //////将每行都放进strLine

符串里

{

MessageBox(strLine);

}

}

### 选择目录

TCHAR pszPath[MAX\_PATH];

BROWSEINFO bi;

bi.hwndOwner = this->GetSafeHwnd();

bi.pidlRoot = NULL;

bi.pszDisplayName = NULL;

bi.lpszTitle = TEXT("请选择文件夹");

bi.ulFlags = BIF\_RETURNONLYFSDIRS | BIF\_STATUSTEXT;

bi.lpfn = NULL;

bi.lParam = 0;

bi.iImage = 0;

LPITEMIDLIST pidl = SHBrowseForFolder(&bi);

if (pidl == NULL)

{

return;

}

if (SHGetPathFromIDList(pidl, pszPath))

{

AfxMessageBox(pszPath);

}

## 窗口

### 禁止拖动改变窗口大小

ModifyStyle(WS\_THICKFRAME, 0, SWP\_FRAMECHANGED|SWP\_DRAWFRAME);

### 最大化最小化等

在初始化中ShowWindow(SW\_MINIMIZE);

### 设置背景透明

SetWindowLong(this->GetSafeHwnd(),GWL\_EXSTYLE,  
 GetWindowLong(this->GetSafeHwnd(),GWL\_EXSTYLE)^0x80000);  
 HINSTANCE hInst = LoadLibrary("User32.DLL");   
 if(hInst)   
 {   
 typedef BOOL (WINAPI \*MYFUNC)(HWND,COLORREF,BYTE,DWORD);   
 MYFUNC fun = NULL;  
 //取得SetLayeredWindowAttributes函数指针   
 fun=(MYFUNC)GetProcAddress(hInst, "SetLayeredWindowAttributes");  
 if(fun)  
 fun(this->GetSafeHwnd(),0,128,2);   
 FreeLibrary(hInst);   
 }

## 控件

### 使控件变灰

GetDlgItem(IDC\_BUTTON\_START)-> EnableWindow(FALSE);

### 树控件treeControl

#### 创建Tree Control

##### 静态创建

大部分时候我们都是用静态方式创建控件,因为这最简单.直接从Toolbox中拖一个控件放到Dialog中就行了.然后要以在Properties页面设置一些属性.如果要在代码中对控件进行操作,可以这样得到控件指针,GetDlgItem(Ctrl\_ID) .如果想更灵活点可以给控件绑定一个变量.

CTreeCtrl m\_treeShow;

DDX\_Control(pDX, IDC\_TREE1, m\_treeShow);

##### 动态创建

动态创建有两个要注意的地方.

1.怎么指定控件位置和大小. 通过CRect rect(2,2, 4,4,)这样直接的数字指定是很不灵活的,假如页面上控件一多,并且后面又想加入新控件了,会非常混乱不好维护.

比较好的一个方法是在页面上弄一个static text 或picture control,然后把它们的visiable设为false.调整好它们的位置和大小.然后获取它们的CRect作为动态创建控件的CRect.

2.怎么指定控件ID.动态创建时需要显式指定一个ID,我们知道ID只是一个数字.但是关键是确保同一个页面上的ID不能相同.你当然可以简单的定义一个宏.但这样不太好.一个比较简单的方法是在String Table中添加一行,用那个ID.这里不是真的需要string里面的信息,而只是这个ID.当然使用规范还是要遵守,caption里面也随便填点东西别为空.

CTreeCtrl\* m\_pTree;

m\_pTree = new CTreeCtrl;

CRect rect;

GetDlgItem(IDC\_SPACE)->GetWindowRect(&rect); //IDC\_SPCE 是一个static text的ID

ScreenToClient(rect); //屏幕坐标转化成窗口坐标,貌似不能像这样

//GetDlgItem(IDC\_SPACE)->GetClientRect(rect)直接获取窗口坐标的,这样获得到位置是窗口的原点位置.

m\_pTree->Create(WS\_VISIBLE | WS\_TABSTOP | WS\_CHILD | WS\_BORDER

| TVS\_HASBUTTONS | TVS\_LINESATROOT | TVS\_HASLINES

| TVS\_DISABLEDRAGDROP | TVS\_NOTOOLTIPS | TVS\_EDITLABELS

, rect, this, IDS\_TREE); // IDS\_TREE是在string table中随便指定的一ID

HTREEITEM hRoot;

hRoot = m\_pTree->InsertItem(\_T("china"),TVI\_ROOT); //插入一个根节点

#### 插入图片

界面上多整几个图标会比较好看点.那怎么往Tree Control中添加图片呢?

首先要是构造一个类CImageList,往里面添加图片,常用的图片有ico, png, bmp这三种格式.添加它们的方式有点不同

CImageList\* pImgList;

pImgList = new CImageList;

pImgList -> Create(16,16, ILC\_COLOR32 | ILC\_MASK, 0, 1);

//ico图片

pImgList->Add(AfxGetApp()->LoadIconW(IDI\_ICON1));

//bmp图片

CBitmap bmp;

bmp.LoadBitmapW(IDB\_BITMAP1);

pImgList->Add(&bmp, RGB(0,0,0));

//png图片

CPngImage png;

png.Load(IDB\_PNG1);

CBitmap pngBmp;

pngBmp.Attach(png.Detach());

pImgList->Add(&pngBmp, RGB(0,0,0));

//将CImageList与Tree control绑定

m\_pTree-> SetImageList(pImgList, TVSIL\_NORMAL); //这样一绑定,之后插入的节点如果不显式指定图标将默认使用CImageList中第一个图标

HTREEITEM hRoot;

hRoot = m\_pTree->InsertItem( \_T("China"),TVI\_ROOT);

m\_pTree->InsertItem( \_T("China"), 1,1,hRoot, TVI\_LAST) ; //显式指定pImgList中第二图标

#### 插入数据

先所控件绑定到如下变量

CTreeCtrl m\_treeShow;

创建一个以中国为根结点,然后一些省和城市为子结点的树形结构信息.

//设置树形控件样式

m\_treeShow.ModifyStyle(NULL,TVS\_HASBUTTONS | TVS\_LINESATROOT |TVS\_HASLINES);

//HTREEITEM相当于是一个结点的句柄

HTREEITEM hRoot;

HREEITEM hChild;

hRoot = m\_treeShow.InsertItem( "中国", TVI\_ROOT); //插入根结点

hChild = m\_treeShow.InsertItem("湖南省",hRoot);

hChild = m\_treeShow.InsertItem("长沙市",hChild);

hChild = m\_treeShow.InsertItem("岳麓区",hChild);

hChild = m\_treeShow.InsertItem("广东省",hRoot);

#### 删除节点

要删除节点那肯定要想办法怎么找到节点才行.

HTREEITEM hItem, hSelected;

hSelected = m\_treeShow.GetSelectedItem();

m\_treeShow.DeleteItem( hSelected); //删除选择的节点,如果节点下面还有子节点也会被一起删掉.

hItem = m\_treeShow.GetRootItem(); //整棵树的根结点

hItem = m\_treeShow.GetChildItem(hSelected); //获得选择节点的子节点

hItem = m\_treeShow.GetParentItem(hSelected); //获得选择节点的子节点

hItem = m\_treeShow.GetPrevSiblingItem(hSelected); //获得选择节点的上一兄弟结点

hItem = m\_treeShow.GetNextSiblingItem(hSelected); //获得选择节点的下一兄弟结点

m\_treeShow.DeleteAllItems(); //删除所有节点

#### 展开节点

默认情况下节点都是折叠起来的.如果只展开某个结点下面一层的内容.

HTREEITEM hItem, hRoot;

hRoot = m\_treeShow.GetRootItem();

m\_treeShow.Expand ( hRoot, TVE\_EXPAND); //只展开根结点下面一层.

ExpandAllNode( hRoot , m\_treeShow);

那如果要展开所选结点下面所有节点咋整呢

void ExpandAllNode(HTREEITEM hItem, CTreeCtrl& m\_treeShow){

HTREEITEM hChild = m\_treeShow.GetChildItem(hItem); //获得第一个子节点

while(hChild){

m\_treeShow.Expand(hItem,TVE\_EXPAND); //展开当前节点

ExpandTreeNode(hChild, m\_treeShow); //递归展开第一个子节点下所有节点

hChild = m\_treeShow.GetNextSiblingItem(hChild); //获得第二个子节点

}

}

### 计时器Timer使用

我们知道手机里面的闹钟可以设个时间,然后选择重复,这样每天会按时响.在Oracle中设定一个job也可以让它在每天预定的时间执行.

那同样假如你想让一些代码每过多长时间自动执行一次咋整呢?在MFC中可以使用timer,用法如下

#### 示例1,使用消息处理

新建一个MFC project,类型选Dialog,

使用闹钟之前先得设定下时间嘛.这里同时也先设定下多久

//////////////TimerTestDlg.h///////////////////////////

afx\_msg void OnTimer(UINT\_PTR nIDEvent); //回调函数,这里的函数名OnTimer不能换成其他名字.固定的

////////////////////////////////////////////////////////////////

////////////////TimerTestDlg.cpp//////////////////////

#include "afxmt.h"

#define TIMERID 8

**//指定一个timer的ID,你可以同时设定多个timer,就用ID来区分**

BEGIN\_MESSAGE\_MAP(CTimerTestDlg, CDialog)

ON\_WM\_TIMER()

END\_MESSAGE\_MAP()

CCriticalSection g\_criSection;

BOOL CTimerTestDlg::OnInitDialog()

{

  SetTimer(TIMERID,1000,0);  //这里就相当于设定了timer,如果要停掉timer就是KillTimer(TIMERID)

}

void CTimerTestDlg::OnTimer(UINT\_PTR nIDEvent)

{

  g\_criSection.Lock();

  static int nShowNum = 1;

   if( nIDEvent == 8)

   {

       CString strInfo;

       strInfo.Format("I am number %d", nShowNum);

       GetDlgItem(IDC\_STATIC)->SetWindowsText(strInfo);  //假如有一个text control ,用来显示数字

   }

 g\_criSection.Unlock();

}

SetTimer 与 KillTimer两个类的定义

**UINT SetTimer( UINT** nIDEvent**, UINT** nElapse**, void (CALLBACK EXPORT\*** lpfnTimer**)(HWND, UINT, UINT, DWORD) );**

每一个参数表示timer id,用来区分不同的timer,假如同时定义了几个timer,在OnTimer()中做不同处理时,可以判断nIDEvent是哪一个.

nElapse表示多长时间执行一次,单位是毫秒,1000就是一秒.

lpfnTimer是回调函数,上面的示例是写成0,表示不显式指定回调函数,这样就是是调用OnTimer,如果你显式指定了回调函数,则OnTimer()函数就不会被调用了.

**BOOL KillTimer( int** nIDEvent**);**

直接通过timer id就停掉timer就行了.

#### 示例2,直接指定回调函数

void CALLBACK Test(HWND hWnd,UINT para1,UINT\_PTR para2,DWORD para3)

{

}

BOOL CTimerTestDlg::OnInitDialog()

{

SetTimer(TIMERID,1000,Test); //直接调用Test,这样之前写的OnTimer函数就永远不会被调用到了.

}

### 封装dialog控件

新建一个dialog类(带rc的,新建---mfc类,基类选CDialog)

属性中,去掉边框什么的,style设成child

在调用dialog中放一个static控件

然后如此嵌入:

m\_listdialog.Create(IDD\_FORMVIEW\_LIST,GetDlgItem(IDC\_STATIC\_A));

CRect rect;

GetDlgItem(IDC\_STATIC\_A)->GetClientRect(&rect);

m\_listdialog.MoveWindow(&rect);

m\_listdialog.ShowWindow(SW\_SHOW);

### 刷新数据

简单说就是:如果Updatedata(TRUE) == 将控件的值赋值给成员变量;Updatedata(FALSE) == 将成员变量的值赋值给控件

### 随窗口变化大小

总结步骤：

1.创建一个CWnd\*   的数组，在初始化时取的各个控件的句柄，存在这个数组里面     
2.处理WM\_SIZE事件，可以得到新变化后的窗口和最初窗口的变化比例kx,ky     
3.对数组中的每一个控件，计算新位置     
4.MoveWindow移动到新位置

       对话框或视类的大小调后，控件的大小和位置没有变化，界面看起来会很不爽控件是从CWnd派生

的，但不能使用SetWindowPos()或OnSize()或OnSizing()来改变其大小，应在父窗口的WM\_SIZE消息中使

用MoveWindow()来进行调整。

VC++之根据对话框大小调整控件大小

1、在对话框类中加入成员变量CRect m\_rect;用于保存对话框大小变化前的大小；  
2、在对话框的OnInitDialog()函数中获取对话框创建时的大小：GetClientRect(&m\_rect);  
3、在WM\_SIZE的响应函数OnSize()中加入以下代码：  
CWnd \*pWnd;  
pWnd = GetDlgItem(IDC\_LIST);     //获取控件句柄  
if(pWnd)//判断是否为空，因为对话框创建时会调用此函数，而当时控件还未创建  
{  
   CRect rect;   //获取控件变化前大小  
   pWnd->GetWindowRect(&rect);  
   ScreenToClient(&rect);//将控件大小转换为在对话框中的区域坐标  
   //　cx/m\_rect.Width()为对话框在横向的变化比例  
   rect.left=rect.left\*cx/m\_rect.Width();/////调整控件大小  
   rect.right=rect.right\*cx/m\_rect.Width();  
   rect.top=rect.top\*cy/m\_rect.Height();  
   rect.bottom=rect.bottom\*cy/m\_rect.Height();

pWnd->MoveWindow(rect);//设置控件大小  
}  
GetClientRect(&m\_rect);//将变化后的对话框大小设为旧大小

### EditControl CEdit 文本框 文本控件

m\_e1.SetSel(0,-1); //全选  
 m\_e1.Copy();//复制  
 m\_e2.SetSel(0,-1);  
 m\_e2.Paste(); //粘贴

#### 多行

要让EditControl可以多行,要设置属性multiline 和 want return true,回车的时候/r/n切记

#### 追加

CEdit\* pEdit = GetDlgItem(IDC\_EDIT1);

ASSERT(pEdit && pEdit->GetSafeHwnd());

pEdit->SetSel(-1);

pEdit->ReplaceSel(\_T("Hello, World!"));

## 自动操作

### 注意事项:

获取最外层的句柄后,点击的0,0是指的客户区的坐标!

### 获取程序各种鼠标位置

HWND hwndjianxia=::FindWindowA(classname,titlename);

CPoint p;

::GetCursorPos(&p);

//屏幕坐标

CString t;t.Format("%d , %d",p.x,p.y);

GetDlgItem(IDC\_screenpos)->SetWindowText(t);

//相对左上角坐标

CRect lr=NULL;

::GetWindowRect(hwndjianxia,&lr);

t.Format("%d , %d",p.x-lr.left,p.y-lr.top);

GetDlgItem(IDC\_mpos)->SetWindowText(t);

//客户区坐标

::ScreenToClient(hwndjianxia,&p);

t.Format("%d , %d",p.x,p.y);

GetDlgItem(IDC\_mclientpos)->SetWindowText(t);

### 组合键

::keybd\_event(VK\_CONTROL, 0, 0, 0);

xauto::PostAnjian(h,'G');

::keybd\_event(VK\_CONTROL, 0, KEYEVENTF\_KEYUP, 0);

### Vmware

先按ctrl+G进入,完了用全局的mouse\_event

## 多线程

### 第一种方法,获取全局的dlg指针用于在线程函数中调用(可以互相改,推荐使用)

dlg.cpp外部定义:线程处理函数, 全局dlg指针(用于在线程中调用), 全局线程变量(用于暂停等):

//UINT th1(LPVOID lpParam);

CxxxDlg \*dlg;

CWinThread \* pThread;

dlg.cpp初始化时获取dlg指针

dlg = (CxxxDlg \*)AfxGetMainWnd();

按钮事件中开始线程

pThread=AfxBeginThread(th1,NULL);

线程处理函数

UINT th1(LPVOID lpParam)

{

return 0;

}

### 暂停线程

pThread->SuspendThread();

### 恢复线程

pThread->ResumeThread();

### 第二种方法,通过AfxBeginThread中的第二个参数传递结构体(从dlg类向单独线程传,貌似不能互相改)

在dlg.h上面,外面定义线程处理函数和要传递的类

UINT threadfun(LPVOID lpParam);

struct threadInfo//从dlg传到线程

{

CString path;

int num;

};

dlg.h内部定义要传递的类

threadInfo info;

dlg.cpp按钮事件中开启线程,传递参数

info.path=gamepath;

info.num=gamenum;

AfxBeginThread(logfunc,&info);

dlg.cpp外面线程函数中,解析得到传入的参数

UINT logfunc(LPVOID lpParam)

{

threadInfo \*in=(threadInfo\*)lpParam;

msg::loggroup(in->path,in->num);

return 0;

}

## 字符串

### 字符串分割

AfxExtractSubString函数的用法

在我做加密软件时，需要给加密后的文件加个后缀（比如.zq），然后解密时又需要去掉后缀，还原本来的后缀，并在文件名后加“解密”二字，我采取的就是这个比较不常用的MFC分割字符串的函数AfxExtractSubString(CString& rString, LPCTSTR lpszFullString,int iSubString, TCHAR chSep = '\n')，下面大概说一下它的用法：

注：本文系rainy8758原创，转载请注明出处：http://blog.hjenglish.com/rainy8758/articles/1010963.html

参数说明：

rString 得到的字符串；lpszFullString 待分割的字符串；iSubString 要得到第几个字符串；chSep 个子串之间的分隔符

例如，有一个字符串strFullString = "abcd-efg-hijkl-mn";则有：

CString strTmp;

AfxExtractSubString( strTmp, (LPCTSTR)strFullString, 0, '-');//strTmp的内容为abcd

AfxExtractSubString( strTmp, (LPCTSTR)strFullString, 2, '-');//strTmp的内容为hijkl

感觉蛮好用的。

注：本文系rainy8758原创，转载请注明出处：http://blog.hjenglish.com/rainy8758/articles/1010963.htm

AfxExtractSubString应用实例

然后我想要的结果也就可以通过这个函数来实现：比如得到一个字符串m\_savefile = "C:\Downloads\1.txt.zq";

CString m\_cd,m\_sd,m\_savefile;

AfxExtractSubString(m\_cd, (LPCTSTR)m\_savefile, 0, '.');

AfxExtractSubString(m\_sd, (LPCTSTR)m\_savefile, 1, '.');

m\_savefile=m\_cd+"解密."+m\_sd;

这样现在的m\_savefile="C:\Downloads\1解密.txt";

只是这样会有一个问题，就是万一在文件夹中带有"."，比如（C:\Downloads（1.0）\1.txt）这样得出的结果就不是原来想像的那样，而是（C:\Downloads（1解密.0）\1.txt），要是没这样一个路径，解密程序会报错的，所以还得想个改进的方法。这样的话，等忙完这段时间，我再好好想想，找个更好的方法和函数来实现这个功能。

AfxExtractSubString是截取字符串的函数，很好用，不过美中不足的地方在与它只能使用单个字符作为分割符。

但是这种情况在很多时候都行不通，如果分割符需要是两个字符以上呢？

之前因为这个问题试了很久，也在网上搜索过。不过可惜的是，网上的大部分关于VC截取字符串的文章都是那么同样的几篇，都是写的满复杂然后可以实现了AfxExtractSubString功能而已的，也就是只能用单个字符截取，但是标题却写着用字符串截取字符串，好笑！

不找了，自己写吧。CString里面有Find，然后再组成数组。

void Split(CString source, CStringArray& dest, CString division)

{

dest.RemoveAll();

int pos =0;

int pre\_pos =0;

while( -1!= pos ){

pre\_pos = pos;

pos = source.Find(division,(pos+1));

dest.Add(source.Mid(pre\_pos,(pos-pre\_pos)));

}

}

## 钩子

### 钩子过程

新建mfc dll，完了之后改成多节字符集

新建一个类，要按以下格式

xlfd.h

#pragma once

#ifndef XLFD\_EXPORTS

#define XLFD \_\_declspec(dllexport)

#else

#define XLFD \_\_declspec(dllimport)

#endif

class XLFD xlfd

{

public:

xlfd(void);

~xlfd(void);

static void sethook(HWND hwnd);

static void stophook();

private:

static LRESULT CALLBACK keyproc(int code,WPARAM wp,LPARAM lp);

static LRESULT CALLBACK mouseproc(int code,WPARAM wp,LPARAM lp);

};

xlfd.cpp

#include "StdAfx.h"

#include "xlfd.h"

//

//防¤¨¤止1hook对?象¨®自Á?动¡¥释º¨ª放¤?的Ì?

#pragma data\_seg("mydata")

HHOOK hookmouse=NULL;

HHOOK hookkey=NULL;

HWND hz=NULL;

#pragma data\_seg()

xlfd::xlfd(void)

{

}

xlfd::~xlfd(void)

{

}

LRESULT CALLBACK xlfd::keyproc(int code,WPARAM wp,LPARAM lp)

{

AFX\_MANAGE\_STATE(AfxGetStaticModuleState());

if(wp==VK\_SPACE&&((lp&(1<<31))==0))

{

AfxMessageBox("space is pressed");

return true;

}

return CallNextHookEx(0,code,wp,lp);

}

LRESULT CALLBACK xlfd::mouseproc(int code,WPARAM wp,LPARAM lp)

{

AFX\_MANAGE\_STATE(AfxGetStaticModuleState());

if(wp==WM\_LBUTTONDBLCLK)

{

LPMOUSEHOOKSTRUCT pmouse=(MOUSEHOOKSTRUCT FAR\*)lp;

POINT pm=pmouse->pt;

::ScreenToClient(hz,&pm);//hwnd全¨?局?变À?量¢?

CString ss;ss.Format("%d %d",pm.x,pm.y);AfxMessageBox(ss);

}

return CallNextHookEx(0,code,wp,lp);

}

void xlfd::sethook(HWND hwnd)

{

AFX\_MANAGE\_STATE(AfxGetStaticModuleState());

hz=hwnd;

if(!hwnd)

AfxMessageBox("请?先¨¨打ä¨°开a要°ahook的Ì?程¨¬序¨°");

else

{

DWORD dwthreadid=GetWindowThreadProcessId(hwnd,NULL);

HINSTANCE hmod=GetModuleHandle("钩3子Á¨®mfcdll.dll");

hookkey=SetWindowsHookEx(WH\_KEYBOARD,keyproc,hmod,dwthreadid);

hookmouse=SetWindowsHookEx(WH\_MOUSE,mouseproc,hmod,dwthreadid);

}

}

void xlfd::stophook()

{

UnhookWindowsHookEx(hookmouse);

UnhookWindowsHookEx(hookkey);

}

编译，生成dll

新建mfc程序，依然是多节字符集，调用时：

#include "../钩3子Á¨®mfcdll/xlfd.h"

#pragma comment(lib,"../Debug/钩3子Á¨®mfcdll.lib")

### MFC扩展dll(可导出对话框)

#### 注意

* 调用方必须为共享dll的!!!
* 调用方和dll方必须字符集一样!!!

#### Dll对话框:

##### 要在对话框类.h加资源id

#define IDD\_DIALOG(从resource中考过来就行)

##### 要在对话框类名前加 AFX\_EXT\_CLASS

如 class AFX\_EXT\_CLASS xlog:public CDialog

#### 调用方

头文件中包含”dll.h”

载入lib库

#pragma comment(lib,”xdll.lib”)

##### 例子

## Dll

### Mfc调用dll

注意:必须选dll动态,字符集要与dll一样

嵌入STATIC\_TEXT

m\_dlgTree.Create(IDD\_DIALOG\_TREE,GetDlgItem(IDC\_STATIC\_TREE));

CRect rect;

GetDlgItem(IDC\_STATIC\_TREE)->GetClientRect(&rect);

m\_dlgTree.MoveWindow(&rect);

m\_dlgTree.ShowWindow(SW\_SHOW);

### Dll调试

把调用dll的程序放到dll工程debug文件夹中,然后dll项目-属性-调试—选命令参数—浏览---程序即可

## 备忘

### 自动操作,序列号,过期等自己的库

#### xlfd.h

#pragma once

#include"iphlpapi.h"

#pragma comment(lib,"Iphlpapi.lib")//以°?上¦?两¢?行D是º?获?取¨?MAC地Ì?址¡¤所¨´需¨¨

//一°?下?是º?获?取¨?网ª?络?时º¡À间?所¨´需¨¨

#define ERR\_BEGIN (INT)0

#define ERR\_UNSUCCESS (INT)(ERR\_BEGIN-1)

#define ERR\_WSASTARTUP (INT)(ERR\_BEGIN-2)

#define ERR\_CREATESOCKET (INT)(ERR\_BEGIN-3)

#define ERR\_ADDRESS (INT)(ERR\_BEGIN-4)

#define ERR\_SETREUSEADDR (INT)(ERR\_BEGIN-5)

#define ERR\_CREATEEVENT (INT)(ERR\_BEGIN-6)

#define ERR\_SENDTO (INT)(ERR\_BEGIN-7)

#define ERR\_EVENTSELECT (INT)(ERR\_BEGIN-8)

struct NTP\_Packet

{

int Control\_Word;

int root\_delay;

int root\_dispersion;

int reference\_identifier;

\_\_int64 reference\_timestamp;

\_\_int64 originate\_timestamp;

\_\_int64 receive\_timestamp;

int transmit\_timestamp\_seconds;

int transmit\_timestamp\_fractions;

}; //以°?上¦?是º?获?取¨?网ª?络?时º¡À间?所¨´需¨¨

class xlfd

{

public:

xlfd(void);

~xlfd(void);

static void PostInputString(HWND hwnd, CString str);

static void PostRightClick(HWND hwnd, int x, int y);

static void PostLeftClick(HWND hwnd, int x, int y);

static void PostMouseMove(HWND hwnd,int x,int y);

static void delay(int sec);

static void PostAnjian(HWND hwnd, int jianma);

static void SendInputString(HWND hwnd, CString str);

static void SendRightClick(HWND hwnd, int x, int y);

static void SendLeftClick(HWND hwnd, int x, int y);

static void SendAnjian(HWND hwnd, int jianma);

//获?取¨?网ª?络?时º¡À间?

static INT iGetInternetTime(CString strRemoteIP,SYSTEMTIME \*pSysTime,INT iTimeOut);

static CTime GetNetTime(void);

static bool isguoqi(int year,int month,int day,int hour,int minite,int second);

//获?CPUID

static CString GetCPUID();

static CString GetMac(void);

static CString xuliehao();

//剑¡ê侠¨¤世º¨¤界? 沙¦3盘¨¬登Ì?陆?相¨¤关?

static void log(CString gameinfo);//登Ì?陆?一°?个?

static void loggroup(CString path,CString zhanghao,int num);//整?个?，ê?包ã¨¹括¤¡§读¨¢取¨?账?号?密¨¹码?

//读¨¢取¨?文?件t

static void WriteCString(CString str);

static CString ReadCString(void);

private:

};

#### xlfd.cpp

#include "StdAfx.h"

#include "xlfd.h"

xlfd::xlfd(void)

{

}

xlfd::~xlfd(void)

{

}

void xlfd::PostInputString(HWND hwnd, CString str)

{

for(int i=0 ; i<str.GetLength() ; i++)

{

::PostMessage(hwnd,WM\_CHAR,str[i] & 0xFF,0);

}

}

void xlfd::PostRightClick(HWND hwnd, int x, int y)

{

::PostMessage(hwnd, WM\_RBUTTONDOWN, MK\_LBUTTON, MAKELPARAM(x,y));

::PostMessage(hwnd, WM\_RBUTTONUP, 0, MAKELPARAM(x,y));

}

void xlfd::PostLeftClick(HWND hwnd, int x, int y)

{

::PostMessage(hwnd, WM\_LBUTTONDOWN, MK\_LBUTTON, MAKELPARAM(x,y));

::PostMessage(hwnd, WM\_LBUTTONUP, 0, MAKELPARAM(x,y));

}

void xlfd::PostMouseMove(HWND hwnd,int x,int y)

{

::PostMessage(hwnd, WM\_MOUSEMOVE, 0, MAKELPARAM(x,y));

}

void xlfd::delay(int wInterval)

{

LARGE\_INTEGER litmp;

LONGLONG QPart1,QPart2;

double dfMinus, dfFreq, dfTim;

QueryPerformanceFrequency(&litmp);

dfFreq = (double)litmp.QuadPart;// 获?得Ì?计?数ºy器¡Â的Ì?时º¡À钟¨®频¦Ì率¨º

QueryPerformanceCounter(&litmp);

QPart1 = litmp.QuadPart;// 获?得Ì?初?始º?值¦Ì

do

{

QueryPerformanceCounter(&litmp);

QPart2 = litmp.QuadPart;//获?得Ì?中D止1值¦Ì

dfMinus = (double)(QPart2-QPart1);

dfTim = dfMinus / dfFreq;// 获?得Ì?对?应®|的Ì?时º¡À间?值¦Ì，ê?单Ì£¤位?为a秒?

}while(dfTim<0.001\*wInterval);

}

void xlfd::PostAnjian(HWND hwnd, int jianma)

{

::PostMessage(hwnd,WM\_KEYDOWN,jianma,0);

::PostMessage(hwnd,WM\_KEYUP,jianma,0);

}

void xlfd::SendInputString(HWND hwnd, CString str)

{

for(int i=0 ; i<str.GetLength() ; i++)

{

::SendMessage(hwnd,WM\_CHAR,str[i] & 0xFF,0);

}

}

void xlfd::SendRightClick(HWND hwnd, int x, int y)

{

::SendMessage(hwnd, WM\_RBUTTONDOWN, MK\_LBUTTON, MAKELPARAM(x,y));

::SendMessage(hwnd, WM\_RBUTTONUP, 0, MAKELPARAM(x,y));

}

void xlfd::SendLeftClick(HWND hwnd, int x, int y)

{

::SendMessage(hwnd, WM\_LBUTTONDOWN, MK\_LBUTTON, MAKELPARAM(x,y));

::SendMessage(hwnd, WM\_LBUTTONUP, 0, MAKELPARAM(x,y));

}

void xlfd::SendAnjian(HWND hwnd, int jianma)

{//主¡Â要°a看¡ähwnd，ê?别Àe错ä¨ª了¢?

::SendMessage(hwnd,WM\_KEYDOWN,jianma,0);

::SendMessage(hwnd,WM\_KEYUP,jianma,0);

}

INT xlfd::iGetInternetTime(CString strRemoteIP,SYSTEMTIME \*pSysTime,INT iTimeOut)

{

if (pSysTime == NULL)

return -1;

WORD wVersionRequested;

WSADATA wsaData;

SOCKET soc = INVALID\_SOCKET;

INT iErrCode = 1;

struct sockaddr\_in addrSrv;

NTP\_Packet NTP\_Send = {0},NTP\_Recv = {0};

WSAEVENT dwEvent = WSA\_INVALID\_EVENT;

INT intOptVal=1;

int sockaddr\_Size = sizeof(addrSrv);

WSAEVENT \*pEventArray = &dwEvent;

DWORD dwEventIndex = 0;

WSANETWORKEVENTS NetworkEvents;

INT iRevLen = 0;

INT iRevSuccess = ERR\_UNSUCCESS;

DWORD dwStartTick = 0,dwCurrTick;

// 初?始º?化¡¥版ã?本À?

wVersionRequested = MAKEWORD( 2, 2 );

if (0 != WSAStartup(wVersionRequested, &wsaData))

{

iErrCode= ERR\_WSASTARTUP;

goto SocketErrExit;

}

if (LOBYTE(wsaData.wVersion)!=2 || HIBYTE(wsaData.wVersion)!=2)

{

iErrCode= ERR\_WSASTARTUP;

goto SocketErrExit;

}

soc = socket(AF\_INET,SOCK\_DGRAM,IPPROTO\_UDP);

if (INVALID\_SOCKET == soc)

{

iErrCode = ERR\_CREATESOCKET;

goto SocketErrExit;

}

if (setsockopt(soc, SOL\_SOCKET, SO\_REUSEADDR,(char \*)&intOptVal, sizeof(intOptVal)) == SOCKET\_ERROR)

{

iErrCode = ERR\_SETREUSEADDR;

goto SocketErrExit;

}

if ((dwEvent = WSACreateEvent()) == WSA\_INVALID\_EVENT)

{

iErrCode = ERR\_CREATEEVENT;

goto SocketErrExit;

}

if (WSAEventSelect(soc, dwEvent, FD\_READ|FD\_CLOSE) == SOCKET\_ERROR)

{

iErrCode = ERR\_EVENTSELECT;

goto SocketErrExit;

}

addrSrv.sin\_addr.S\_un.S\_addr = inet\_addr(strRemoteIP);

if (INADDR\_NONE == addrSrv.sin\_addr.S\_un.S\_addr)

{

iErrCode = ERR\_ADDRESS;

goto SocketErrExit;

}

addrSrv.sin\_family = AF\_INET;

addrSrv.sin\_port = htons(123);

NTP\_Send.Control\_Word = htonl(0x0B000000);

NTP\_Send.root\_delay = 0;

NTP\_Send.root\_dispersion = 0;

NTP\_Send.reference\_identifier = 0;

NTP\_Send.reference\_timestamp = 0;

NTP\_Send.originate\_timestamp = 0;

NTP\_Send.receive\_timestamp = 0;

NTP\_Send.transmit\_timestamp\_seconds = 0;

NTP\_Send.transmit\_timestamp\_fractions = 0;

if(SOCKET\_ERROR == sendto(soc,(const char\*)&NTP\_Send,sizeof(NTP\_Send),

0,(struct sockaddr\*)&addrSrv,sizeof(addrSrv)))

{

iErrCode = ERR\_SENDTO;

goto SocketErrExit;

}

dwStartTick = GetTickCount();

while(1)

{

dwCurrTick = GetTickCount();

if ((dwCurrTick<dwStartTick && dwCurrTick>(DWORD)iTimeOut)

|| (dwCurrTick-dwStartTick) > (DWORD)iTimeOut )

{

break;

}

if ((dwEventIndex = WSAWaitForMultipleEvents(1, pEventArray, FALSE,50/\*WSA\_INFINITE\*/, FALSE)) == WSA\_WAIT\_FAILED

|| dwEventIndex == WSA\_WAIT\_TIMEOUT)

{

continue;

}

if (WSAEnumNetworkEvents(soc,dwEvent,&NetworkEvents) == SOCKET\_ERROR)

{

continue;

}

WSAResetEvent(dwEvent);

if (NetworkEvents.lNetworkEvents & FD\_READ)

{

if (NetworkEvents.iErrorCode[FD\_READ\_BIT] != 0)

{

continue;

}

iRevLen = recvfrom(soc,(char\*)&NTP\_Recv,sizeof(NTP\_Recv),0,(struct sockaddr\*)&addrSrv,&sockaddr\_Size);

if (iRevLen > 0)

{

float Splitseconds;

struct tm \*lpLocalTime;

time\_t ntp\_time;

// 获?取¨?时º¡À间?服¤t务?器¡Â的Ì?时º¡À间?

ntp\_time = ntohl(NTP\_Recv.transmit\_timestamp\_seconds)-2208988800;

lpLocalTime = localtime(&ntp\_time);

if(lpLocalTime == NULL)

{

break;

}

// 获?取¨?新?的Ì?时º¡À间?

pSysTime->wYear =lpLocalTime->tm\_year+1900;

pSysTime->wMonth =lpLocalTime->tm\_mon+1;

pSysTime->wDayOfWeek =lpLocalTime->tm\_wday;

pSysTime->wDay =lpLocalTime->tm\_mday;

pSysTime->wHour =lpLocalTime->tm\_hour;

pSysTime->wMinute =lpLocalTime->tm\_min;

pSysTime->wSecond =lpLocalTime->tm\_sec;

Splitseconds=(float)ntohl(NTP\_Recv.transmit\_timestamp\_fractions);

Splitseconds=(float)0.000000000200 \* Splitseconds;

Splitseconds=(float)1000.0 \* Splitseconds;

pSysTime->wMilliseconds = (unsigned short)Splitseconds;

iRevSuccess = 1;

break;

}

else if (iRevLen == SOCKET\_ERROR)

{

if (WSAGetLastError() == WSAEWOULDBLOCK)

continue;

}

else

{

break;

}

}

}

iErrCode = iRevSuccess;

SocketErrExit:

if (dwEvent != WSA\_INVALID\_EVENT)

{

WSACloseEvent(dwEvent);

dwEvent = WSA\_INVALID\_EVENT;

}

if (soc != INVALID\_SOCKET)

{

closesocket(soc);

soc = INVALID\_SOCKET;

}

WSACleanup();

return iErrCode;

}

CTime xlfd::GetNetTime(void)

{

SYSTEMTIME newtime;

CString strContent;

if (iGetInternetTime("204.152.184.72",&newtime,4000) > 0)

{

strContent.Format("%d-%d-%d %d:%d:%d",newtime.wYear,newtime.wMonth,newtime.wDay,

newtime.wHour,newtime.wMinute,newtime.wSecond);

CTime temp(newtime.wYear,newtime.wMonth,newtime.wDay,

newtime.wHour,newtime.wMinute,newtime.wSecond);

//MessageBox(strContent);

return temp;

}

else

{

strContent.Format("GetInternetTime error");

//MessageBox(strContent);

CTime temp(2100,11,30,23,59,59);

return temp;

}

}

bool xlfd::isguoqi(int year,int month,int day,int hour,int minite,int second)

{

CTime nowtime=GetNetTime();

CString temp;temp.Format("%d:%d:%d %d:%d:%d",nowtime.GetYear(),nowtime.GetMonth(),nowtime.GetDay(),nowtime.GetHour(),nowtime.GetMinute(),nowtime.GetSecond());

CTime settime=CTime(year,month,day,hour,minite,second);

if(nowtime>settime)

return true;

else

return false;

}

CString xlfd::GetCPUID()

{

CString CPUID;

unsigned long s1,s2;

unsigned char vendor\_id[]="------------";

char sel;

sel='1';

CString VernderID;

CString MyCpuID,CPUID1,CPUID2;

switch(sel)

{

case '1':

\_\_asm{

xor eax,eax //eax=0:取¨?Vendor信?息¡é

cpuid //取¨?cpu id指?令¢?，ê?可¨¦在¨²Ring3级?使º1用®?

mov dword ptr vendor\_id,ebx

mov dword ptr vendor\_id[+4],edx

mov dword ptr vendor\_id[+8],ecx

}

VernderID.Format("%s-",vendor\_id);

\_\_asm{

mov eax,01h //eax=1:取¨?CPU序¨°列¢D号?

xor edx,edx

cpuid

mov s1,edx

mov s2,eax

}

CPUID1.Format("%08X%08X",s1,s2);

\_\_asm{

mov eax,03h

xor ecx,ecx

xor edx,edx

cpuid

mov s1,edx

mov s2,ecx

}

CPUID2.Format("%08X%08X",s1,s2);

break;

case '2':

{

\_\_asm{

mov ecx,119h

rdmsr

or eax,00200000h

wrmsr

}

}

AfxMessageBox("CPU id is disabled.");

break;

}

MyCpuID = CPUID1+CPUID2;

CPUID = MyCpuID;

return CPUID;

}

CString xlfd::GetMac(void)

{

int i=0;CString str;

PIP\_ADAPTER\_INFO pAdapterInfo;ULONG ulOutbufLen = sizeof(IP\_ADAPTER\_INFO);//注Á¡é意°a是º?ULONG类¤¨¤型¨ª,而?非¤?PULONG类¤¨¤型¨ª

pAdapterInfo = ( IP\_ADAPTER\_INFO \*)malloc( sizeof(IP\_ADAPTER\_INFO) );//为a其?动¡¥态¬?分¤?配?内¨²存ä?,若¨?大ä¨®小?不?够?,后¨®面?会¨¢有®D调Ì¡Â整?

if ( ERROR\_BUFFER\_OVERFLOW == GetAdaptersInfo( pAdapterInfo, &ulOutbufLen) )

{//MSDN上¦?介¨¦绍¦¨¹,因°¨°为a获?取¨?的Ì?消?息¡é,可¨¦能¨¹超?过y预¡è先¨¨设¦¨¨置?的Ì?缓o冲?区?长¡è度¨¨,造¨¬成¨¦返¤¦Ì回?,而?一°?旦Ì?返¤¦Ì回?获?取¨?的Ì?资Á¨º料¢?将?为a空?,此ä?处ä|为a防¤¨¤止1这a种?情¨¦况?发¤¡é生¦¨²,

free(pAdapterInfo);//可¨¦以°?测a试º?一°?下?结¨¢果?,去¨£¤掉Ì?这a部?分¤?代ä¨²码?后¨®,返¤¦Ì回?值¦Ì为a空?.此ä?处ä|重?构1,直¡À到Ì?成¨¦功|

pAdapterInfo = (IP\_ADAPTER\_INFO \*) malloc (ulOutbufLen); //第Ì¨²1次ä?大ä¨®小?不?够?,第Ì¨²2次ä?返¤¦Ì回?所¨´需¨¨长¡è度¨¨

}

if ( ERROR\_SUCCESS == GetAdaptersInfo( pAdapterInfo, &ulOutbufLen ))//MSDN上¦?是º?那?样¨´写¡ä,也°2可¨¦写¡ä成¨¦NO\_ERROR,都?正y确¨¡¤

str.Format("%02x-%02x-%02x-%02x-%02x-%02x\n",

pAdapterInfo->Address[0], //注Á¡é意°a此ä?处ä|是º?数ºy组Á¨¦格?式º?,网ª?上¦?有®D的Ì?没?有®D[0][1]等Ì¨¨,str.Format格?式º?就¨ª是º?sprintf的Ì?MFC版ã?

pAdapterInfo->Address[1],

pAdapterInfo->Address[2],

pAdapterInfo->Address[3],

pAdapterInfo->Address[4],

pAdapterInfo->Address[5]);//,pAdapterInfo->Address[6],如¨?果?你?想?知a道Ì¨¤第Ì¨²7个?是º?个?什º2么¡ä值¦Ì?可¨¦以°?一°?试º?,但Ì?是º?值¦Ì为a00(即¡ä空?值¦Ì)

return str;

}

void xlfd::log(CString gameinfo)

{

//gameinfo="460089744xlfd xlfdlengling 天¬¨¬地Ì?无T极?";

delay(12000);//等Ì¨¨待äy进?入¨?账?号?窗ä¡ã口¨²

int i=0;char \*p;char\*s1,\*s2,\*s3;CString str1,str2,str3;

for(p=strtok(gameinfo.GetBuffer(gameinfo.GetLength())," ");p!=NULL;)

{

i++;

if(i==1)

s1=p;

if(i==2)

s2=p;

if(i==3)

s3=p;

p=strtok(NULL," ");

}//分¤?割?

str1.Format("%s",s1);str2.Format("%s",s2);str3.Format("%s",s3 );

HWND hwnd=::FindWindow(NULL,"[#] 剑¡ê侠¨¤世º¨¤界? [#]");//

PostInputString(hwnd,str1);//账?号?输º?入¨?

PostAnjian(hwnd,VK\_TAB);//tab

PostInputString(hwnd,str2);//密¨¹码?

PostLeftClick(hwnd,750,60);//选?服¤t务?器¡Â

delay(2000);

PostLeftClick(hwnd,300,540);//点Ì?输º?入¨?服¤t务?器¡Â

delay(1000);

PostInputString(hwnd,str3);//输º?入¨?服¤t务?器¡Â名?字Á?

delay(1000);

//msg::PostLeftClick(hwnd,300,500);//点Ì?搜?索¡Â到Ì?的Ì?（ê¡§上¦?面?）ê?

//msg::delay(1000);

PostLeftClick(hwnd,600,540);//确¨¡¤定¡§选?择?该?服¤t务?器¡Â

delay(1000);

PostLeftClick(hwnd,400,255);//点Ì?击¡Â账?号?输º?入¨?框¨°

delay(500);

PostLeftClick(hwnd,400,380);//确¨¡¤定¡§

delay(8000);//等Ì¨¨待äy出?现?角?色¦?选?择?

PostLeftClick(hwnd,400,550);//确¨¡¤定¡§

::SendMessage(hwnd,WM\_SYSCOMMAND,SC\_MINIMIZE,0);

//MessageBox(str1);MessageBox(str2);MessageBox(str3);

}

void xlfd::loggroup(CString path, CString zhanghao,int num)

{

HWND hwnd=::FindWindow("SandboxieControlWndClass","Sandboxie Control");

HWND h2=::FindWindowEx(hwnd,NULL,"SysListView32",NULL);

SendAnjian(h2,VK\_DOWN);//先¨¨选?中D第Ì¨²一°?个?沙¦3盘¨¬

CString StrLine;////////读¨¢取¨?一°?行D就¨ª放¤?到Ì?里¤?面?去¨£¤

CStdioFile ReadText( zhanghao,CFile::modeRead|CFile::typeText);//

int hangshu=0;

while(ReadText.ReadString(StrLine))

{

hangshu++;

} //以°?上¦?是º?求¨®行D数ºy

ReadText.Close();

//CString hs;hs.Format("%d",hangshu);AfxMessageBox(hs);

CString \*info=new CString[hangshu];

int count=0;

CStdioFile ReadText2(zhanghao,CFile::modeRead|CFile::typeText);

while(ReadText2.ReadString(StrLine))

{

info[count]=StrLine;

count++;

}

//开a始º?登Ì?陆?

for(int i=0;i<num;i++)

{

PostRightClick(hwnd,0,0);//40开a始º?32一°?个?

PostAnjian(hwnd,VK\_DOWN);//按ã¡ä 下? ，ê?使º1可¨¦选?

PostAnjian(hwnd,VK\_RETURN);//按ã¡äenter，ê?使º1弹Ì¡¥出?右®¨°边À?运?行D

delay(1000);

PostAnjian(hwnd,VK\_DOWN);//两¢?个?下?，ê?选?运?行D程¨¬序¨°

PostAnjian(hwnd,VK\_DOWN);

PostAnjian(hwnd,VK\_RETURN);//打ä¨°开a运?行D程¨¬序¨°对?话¡ã框¨°

delay(1000);

HWND hwndtanchu=::GetForegroundWindow();//当Ì¡À前¡ã最Á?前¡ã端?的Ì?句?柄À¨²

HWND t2=::FindWindowEx(hwndtanchu,NULL,"ComboBoxEx32",NULL);

HWND t3=::FindWindowEx(t2,NULL,"ComboBox",NULL);

HWND t4=::FindWindowEx(t3,NULL,"Edit",NULL);//找¨°到Ì?edit句?柄À¨²

SendInputString(t4,path);//填¬?游®?戏¡¤路¡¤径?

PostAnjian(t4,VK\_RETURN);

delay(1000);

log(info[i]);

if(i%2==1)//往ª¨´下?走Á?一°?个?沙¦3盘¨¬

SendAnjian(h2,VK\_DOWN);

}

}

CString xlfd::xuliehao()

{

//time\_t t;

//time(&t);

//srand(t);//使º1用®?rand每?次ä?不?一°?样¨´的Ì?条¬?件t，ê?种?子Á¨®

CString xlh;

CString cpuid=GetCPUID();

CString mac=GetMac();

xlh=cpuid+mac+"xlfd1988";

for(int i=0;i<xlh.GetLength();i++)

{

int s1=rand()%xlh.GetLength();

int s2=rand()%xlh.GetLength();

char th;

th=xlh[s1];

xlh.SetAt(s1,xlh[s2]);

xlh.SetAt(s2,th);

}

xlh.Replace("\n","");//去¨£¤掉Ì?回?车¦Ì

return xlh;

}

void xlfd::WriteCString(CString str)

{

CStdioFile file;file.Open("D:\\ts.txt",CFile::modeCreate|CFile::modeWrite);

//file.Seek(0,CFile::end);

file.WriteString( str );

file.Close();

}

CString xlfd::ReadCString(void)

{

CString strText = NULL;

CString szLine = NULL;

CStdioFile file;

file.Open("D:\\ts.txt",CFile::modeRead);

while( file.ReadString( szLine ) )

{

strText += szLine;

}

//MessageBox(strText);

file.Close();

return strText;

}