

Win32Project3.cpp File Reference

```
#include "stdafx.h"
#include "Win32Project3.h"
#include < cwchar>
#include <gdiplus.h>
```

Macros

```
#define MAX_LOADSTRING 100
#define IDC_GDI_CAPTURINGANIMAGE 103
```

Functions

ATOM	MyRegisterClass (HINSTANCE hInstance) Forward declarations of functions included in this code module: More...
BOOL	InitInstance (HINSTANCE, int)
LRESULT CALLBACK	WndProc (HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam) FUNCTION: WndProc(HWND, UINT, WPARAM, LPARAM) More...
INT_PTR CALLBACK	About (HWND, UINT, WPARAM, LPARAM)
BOOL CALLBACK	EnumWindowsProc (HWND hWnd, long lParam) FUNCTION: EnumWindowsProc(HWND hWnd, long lParam) More...
INT	GetEncoderClsid (const WCHAR *format, CLSID *pClsid)
int APIENTRY	_tWinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance, LPTSTR lpCmdLine, int nCmdShow) Main function. More...
int	CaptureAnImage (HWND active)

Variables

HINSTANCE	hInst
TCHAR	szTitle [MAX_LOADSTRING]
TCHAR	szWindowClass [MAX_LOADSTRING]
HWND	cliwin
IStorage *	youStorage = NULL

Macro Definition Documentation

<pre>#define IDC_GDI_CAPTURINGANIMAGE 103</pre>
<pre>#define MAX_LOADSTRING 100</pre>

Function Documentation

<pre>int APIENTRY _tWinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance, LPTSTR lpCmdLine, int nCmdShow)</pre> <p>Main function.</p>
<pre>INT_PTR CALLBACK About (HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)</pre>

```
int CaptureAnImage ( HWND active )
```

FUNCTION: [CaptureAnImage\(HWND active\)](#)

PURPOSE: Captures a screenshot of each window and saves them in .jpg files. Creates iStreams for each window and stores in a compound file.

COMMENTS:

Note: This sample will attempt to create a file with the same title as the window, and a max limit of approximately 50 chars

```
BOOL CALLBACK EnumWindowsProc ( HWND hWnd,
                                long lParam
                              )
```

FUNCTION: [EnumWindowsProc\(HWND hWnd, long lParam\)](#)

PURPOSE: Callback function to enumerate through windows.

COMMENTS:

Set up a timer to enumerate through the windows every millisecond, and set a new timer everytime this function is called

```
int GetEncoderClsid ( const WCHAR * format,
                      CLSID * pClsid
                    )
```

FUNCTION: [GetEncoderClsid\(\)](#)

PURPOSE: Finds the correct encoder given a text string format.

COMMENTS:

Will only be used for BMP to JPG conversion.

```
BOOL InitInstance ( HINSTANCE hInstance,
                   int nCmdShow
                 )
```

FUNCTION: [InitInstance\(HINSTANCE, int\)](#)

PURPOSE: Saves instance handle and creates main window

COMMENTS:

In this function, we save the instance handle in a global variable and create and display the main program window.

```
ATOM MyRegisterClass ( HINSTANCE hInstance )
```

Forward declarations of functions included in this code module:

FUNCTION: [MyRegisterClass\(\)](#)

PURPOSE: Registers the window class.

COMMENTS:

This function and its usage are only necessary if you want this code to be compatible with Win32 systems prior to the 'RegisterClassEx' function that was added to Windows 95. It is important to call this function so that the application will get 'well formed' small icons associated with it.

```
LRESULT CALLBACK WndProc ( HWND    hWnd,  
                           UINT     message,  
                           WPARAM  wParam,  
                           LPARAM  lParam  
                           )
```

FUNCTION: `WndProc(HWND, UINT, WPARAM, LPARAM)`

PURPOSE: Processes messages for the main window.

WM_COMMAND - process the application menu WM_PAINT - Paint the main window WM_DESTROY - post a quit message and return

Variable Documentation

HWND cliwin

HINSTANCE hInst

TCHAR szTitle[MAX_LOADSTRING]

TCHAR szWindowClass[MAX_LOADSTRING]

IStorage* youStorage = NULL