

Using ID TECH Universal SDK Library Files in a C++ Project

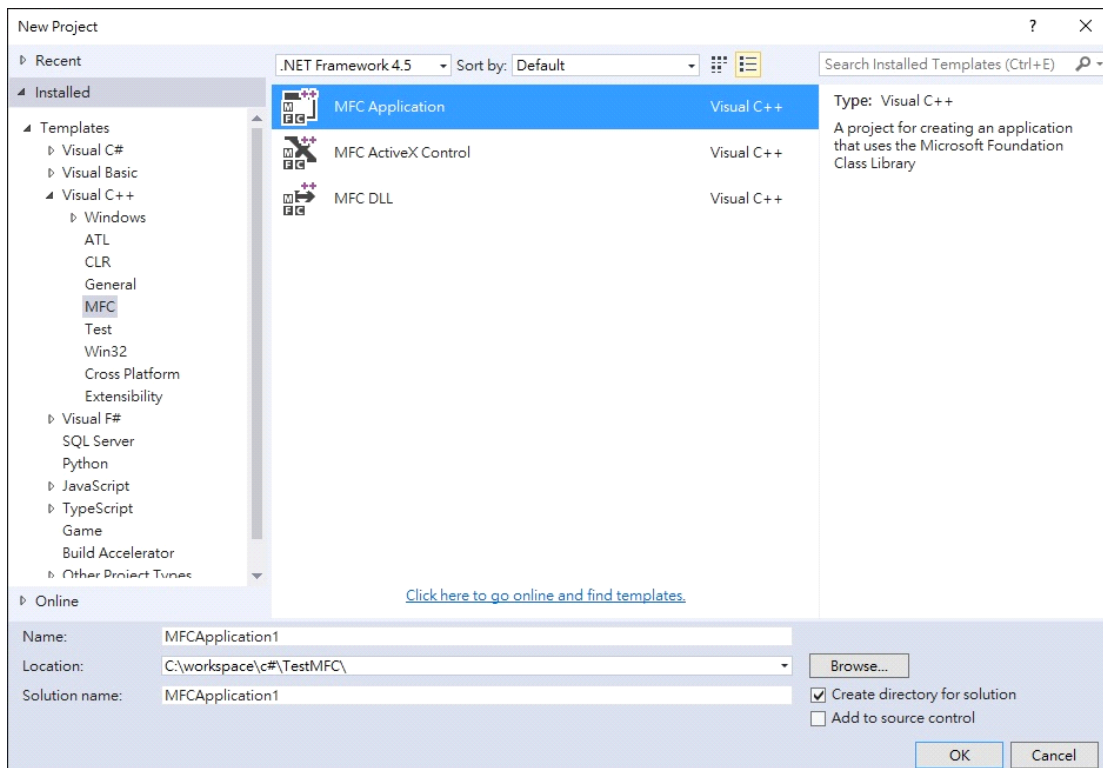
Introduction

From time to time, customers who wish to use ID TECH's Universal SDK for Windows (which is .NET-based and comes with C# code examples) ask if it is possible to do development against the SDK solely in C++ (on Windows). The answer is yes. Universal SDK library files (DLLs) are COM-visible and ready to be accessed from C++ code. (SDK runtimes require the .NET Common Language Runtime, but your C++ binaries can still use the SDK.)

Note that while the example shown in this document involves Microsoft's Visual Studio, it is also possible to use SDK libraries in C++ projects created in Eclipse or other IDEs.

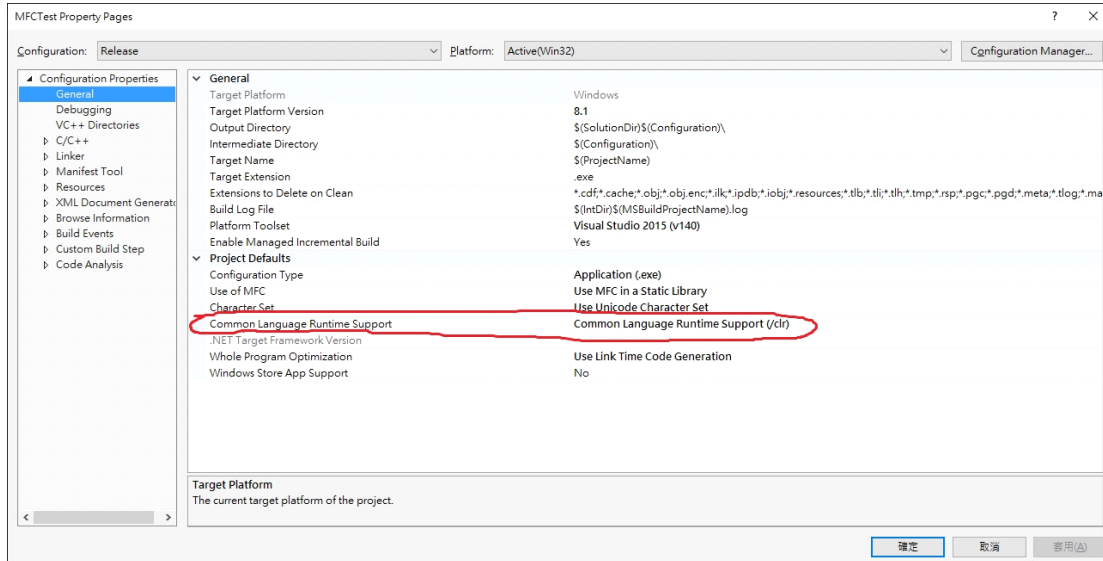
How to Use the IDTechSDK.dll File in a C++ Project:

1. Create a Visual C++ project in Visual Studio 2015 (shown below, an MFC Application as an example).

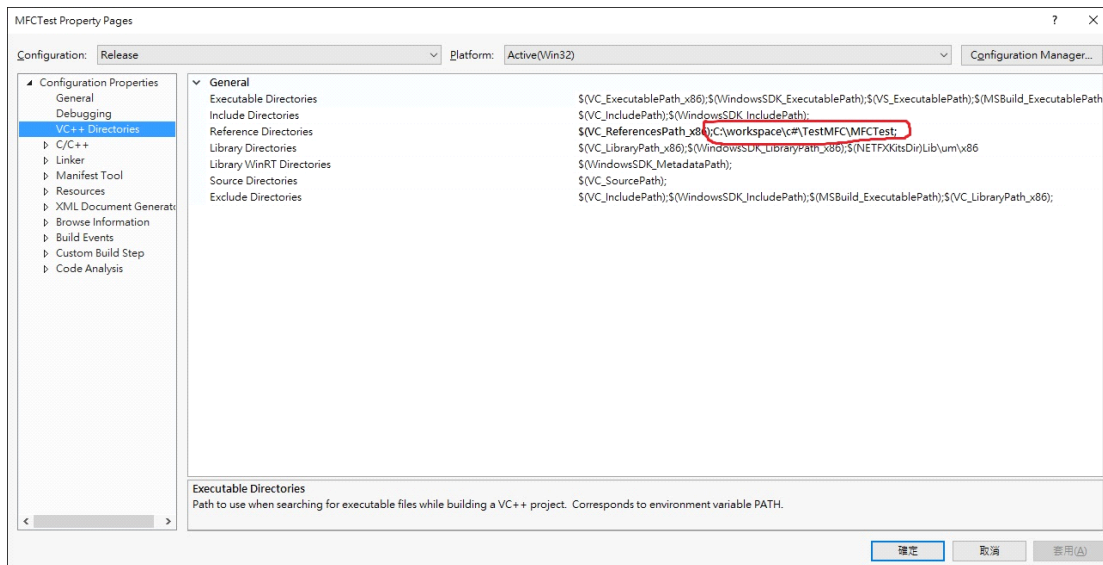


2. Change the properties of the Visual C++ project.

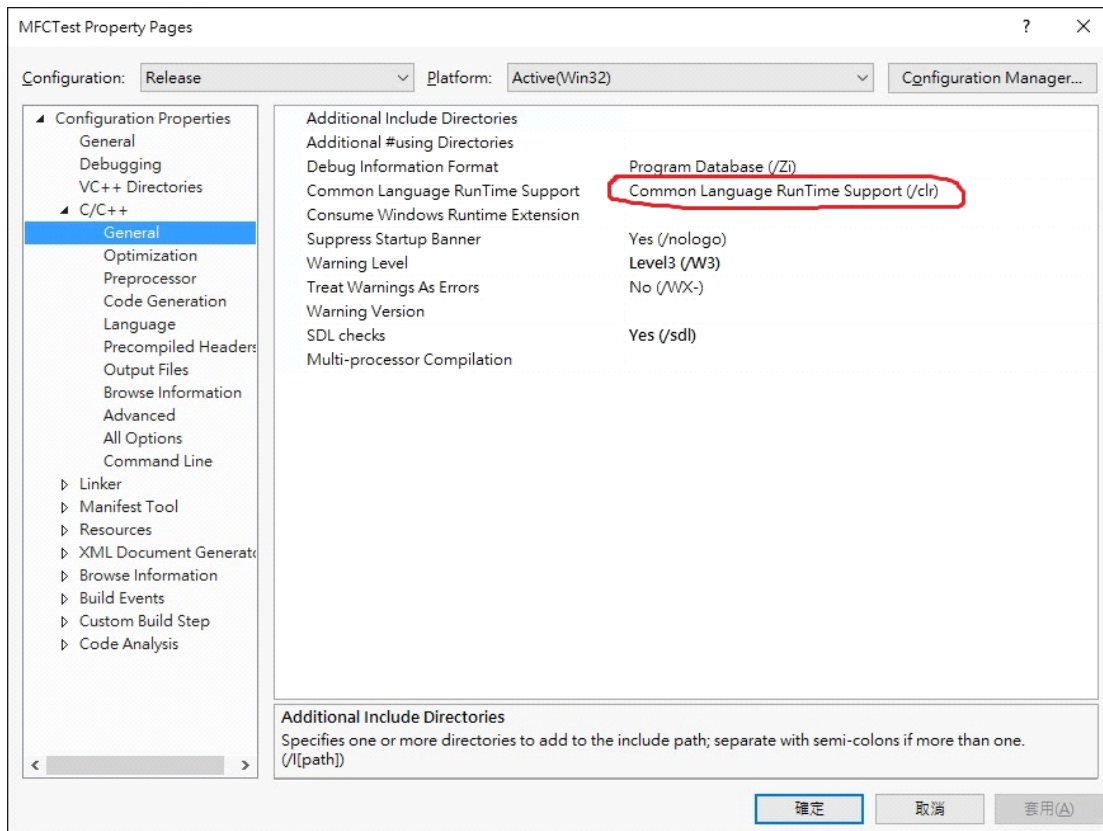
Under the **General** tag, set Common Language Runtime Support under Target Platform to "Common Language Runtime Support (/clr)" under Windows.



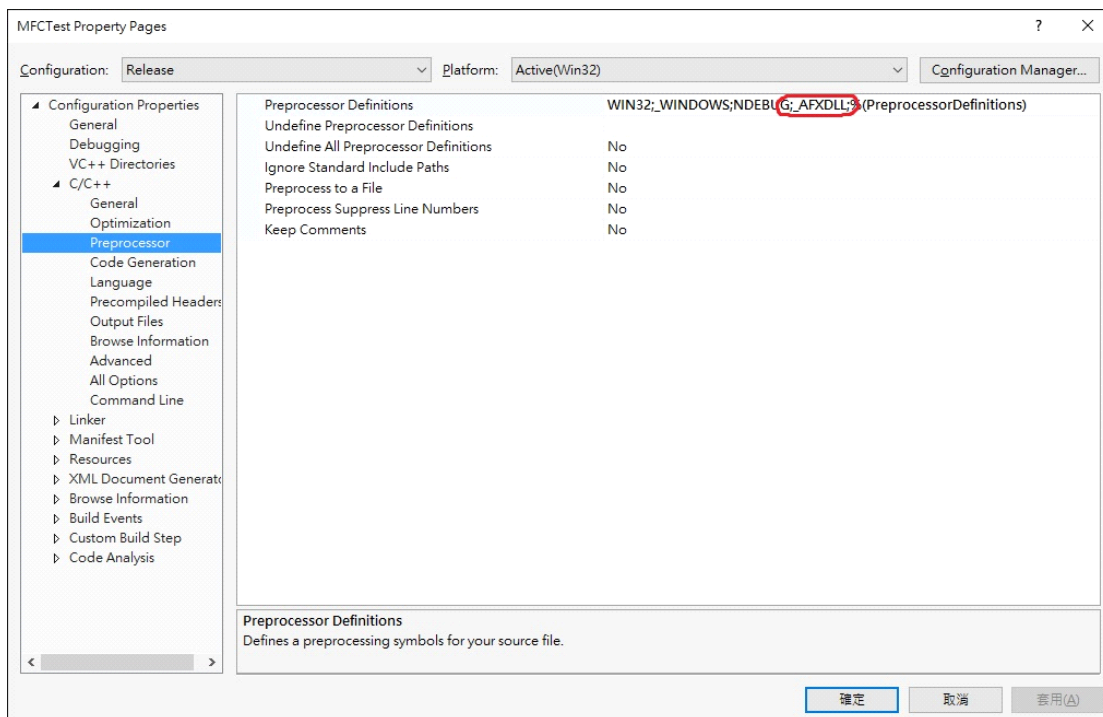
3. Under **VC++ Directories**, add the path to the C# .dll file(s) to Reference Directories.



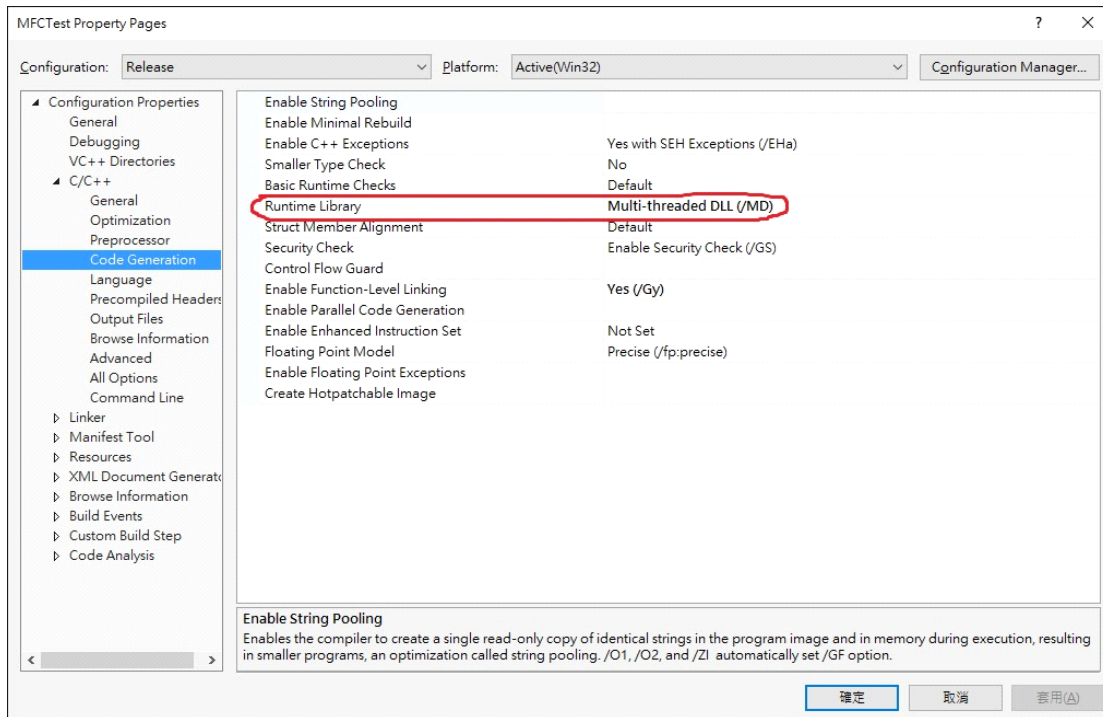
4. Under **C/C++ General**, set Common Language Runtime Support to "Common Language Runtime Support (/clr)."



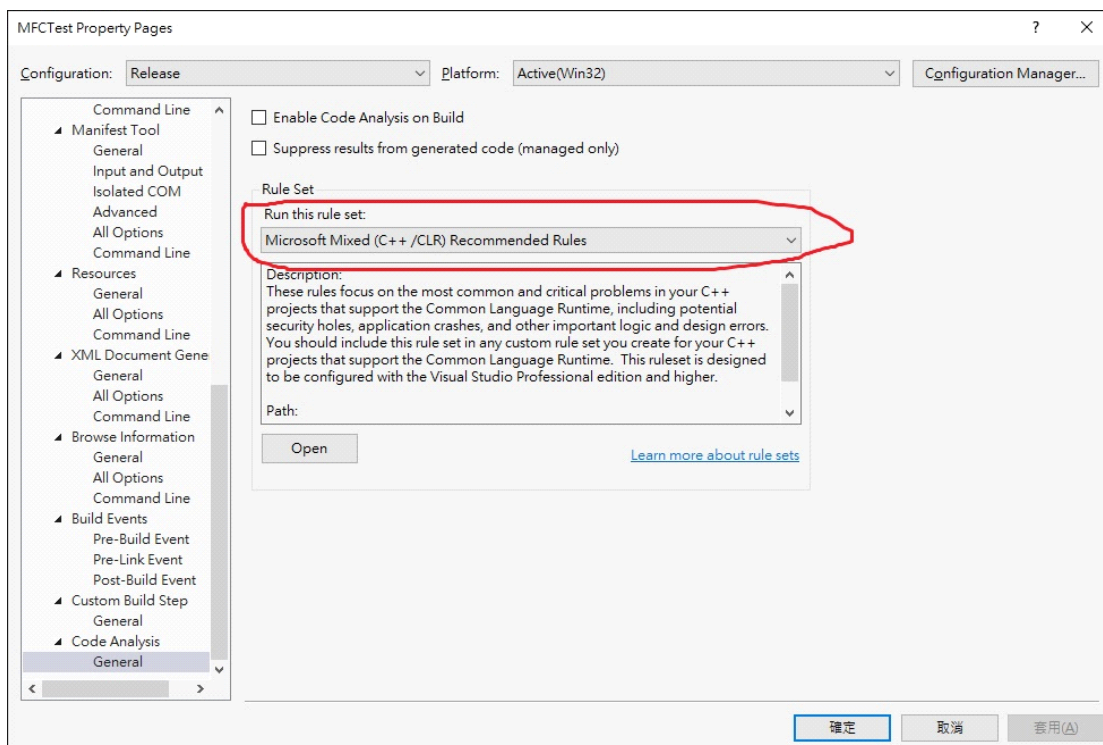
5. Under **C/C++ Preprocessor**, add `_AFXDLL` to Preprocessor Definitions.



6. Under **C/C++ Code Generation**, change Runtime Library to "Multi-threaded DLL (/MD)."



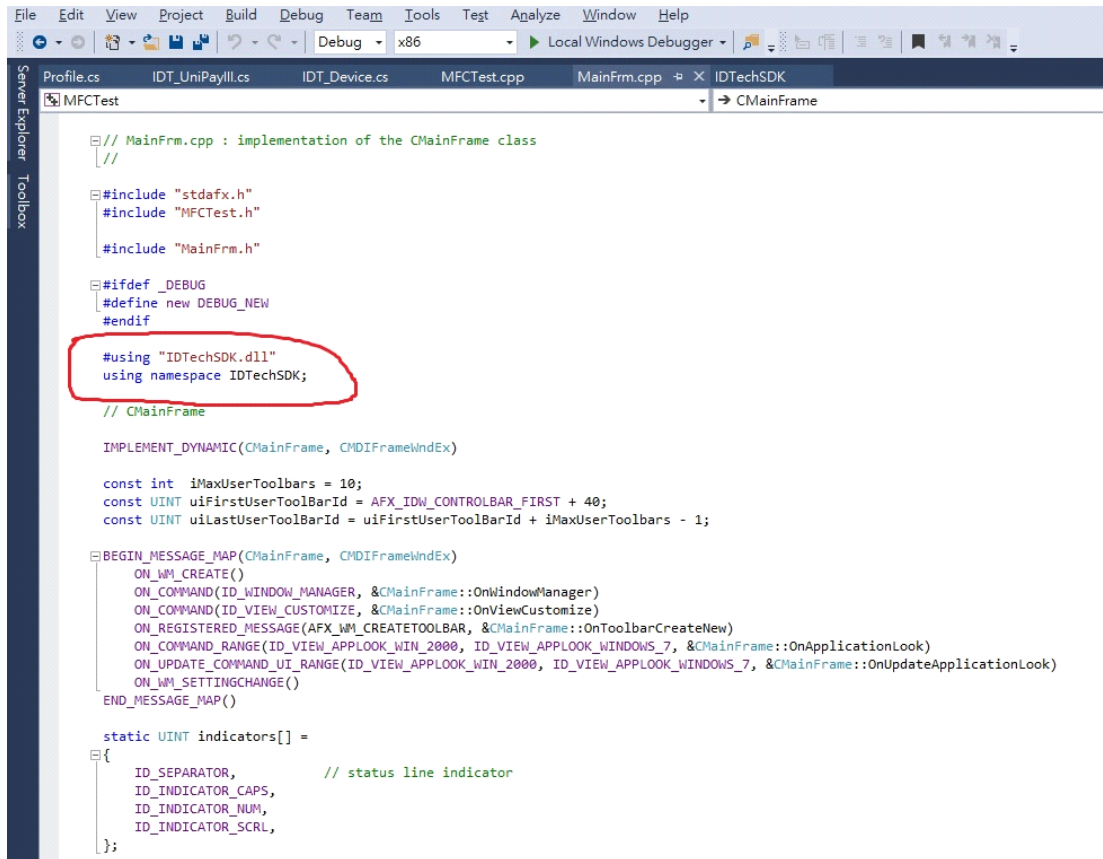
7. Under **Code Analysis General**, change Rule Set to "Microsoft Mixed (C++ /CLR) Recommended Rules."



8. Use **IDTechSDK.dll** in your .cpp file.

a. Open a .cpp file in the Visual C++ project (MainFrm.cpp, for example).

b. Add #using "IDTechSDK.dll" and using namespace IDTechSDK below any #include and #define statements.



```
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug x86 Local Windows Debugger
Profile.cs IDT_UniPayIll.cs IDT_Device.cs MFCTest.cpp MainFrm.cpp IDTechSDK
MFCTest CMainFrame
// MainFrm.cpp : implementation of the CMainFrame class
//
#include "stdafx.h"
#include "MFCTest.h"
#include "MainFrm.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

using "IDTechSDK.dll"
using namespace IDTechSDK;

// CMainFrame

IMPLEMENT_DYNAMIC(CMainFrame, CMDIFrameWndEx)

const int iMaxUserToolbars = 10;
const UINT uiFirstUserToolBarId = AFX_IDW_CONTROLBAR_FIRST + 40;
const UINT uiLastUserToolBarId = uiFirstUserToolBarId + iMaxUserToolbars - 1;

BEGIN_MESSAGE_MAP(CMainFrame, CMDIFrameWndEx)
    ON_WM_CREATE()
    ON_COMMAND(ID_WINDOW_MANAGER, &CMainFrame::OnWindowManager)
    ON_COMMAND(ID_VIEW_CUSTOMIZE, &CMainFrame::OnViewCustomize)
    ON_REGISTERED_MESSAGE(AFX_WM_CREATETOOLBAR, &CMainFrame::OnToolBarCreateNew)
    ON_COMMAND_RANGE(ID_VIEW_APPLOOK_WIN_2000, ID_VIEW_APPLOOK_WINDOWS_7, &CMainFrame::OnApplicationLook)
    ON_UPDATE_COMMAND_UI_RANGE(ID_VIEW_APPLOOK_WIN_2000, ID_VIEW_APPLOOK_WINDOWS_7, &CMainFrame::OnUpdateApplicationLook)
    ON_WM_SETTINGCHANGE()
END_MESSAGE_MAP()

static UINT indicators[] =
{
    ID_SEPARATOR, // status line indicator
    ID_INDICATOR_CAPS,
    ID_INDICATOR_NUM,
    ID_INDICATOR_SCRL,
};
```

c. Declare an object and call the functions in IDTechSDK.dll.

```
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug x86 Local Windows Debugger
Server Explorer Toolbox
Profile.cs IDT_UniPayIII.cs IDT_Device.cs MFCTest.cpp MainFrm.cpp* X IDTechSDK
MFCTest CMainFrame
if (!m_wndMenuBar.Create(this))
{
    TRACE0("Failed to create menubar\n");
    return -1;    // fail to create
}

m_wndMenuBar.SetPaneStyle(m_wndMenuBar.GetPaneStyle() | CBRS_SIZE_DYNAMIC | CBRS_TOOLTIPS | CBRS_FLYBY);

// prevent the menu bar from taking the focus on activation
CMFCPopupMenu::SetForceMenuFocus(FALSE);

if (!m_wndToolBar.CreateEx(this, TBSTYLE_FLAT, WS_CHILD | WS_VISIBLE | CBRS_TOP | CBRS_GRIPPER | CBRS_TOOLTIPS | CBRS_FL
!m_wndToolBar.LoadToolBar(theApp.m_bHiColorIcons ? IDR_MAINFRAME_256 : IDR_MAINFRAME))
{
    TRACE0("Failed to create toolbar\n");
    return -1;    // fail to create
}

CString strToolBarName;
bNameValid = strToolBarName.LoadString(IDS_TOOLBAR_STANDARD);
ASSERT(bNameValid);
m_wndToolBar.SetWindowText(strToolBarName);

//test
IDTechSDK::IDT_Device^ device;
IDTechSDK::IDT_DEVICE_Types type = IDTechSDK::IDT_DEVICE_Types::IDT_DEVICE_UNIPAYIII_USB;
device->setDeviceType(type);

CString strCustomize;
bNameValid = strCustomize.LoadString(IDS_TOOLBAR_CUSTOMIZE);
ASSERT(bNameValid);
m_wndToolBar.EnableCustomizeButton(TRUE, ID_VIEW_CUSTOMIZE, strCustomize);

// Allow user-defined toolbars operations:
InitUserToolbars(NULL, uiFirstUserToolBarId, uiLastUserToolBarId);

if (!m_wndStatusBar.Create(this))
{
    TRACE0("Failed to create status bar\n");
    return -1;    // fail to create
}

m_wndStatusBar.SetIndicators(indicators, sizeof(indicators)/sizeof(UINT));
```

9. Finally, clean and build the Visual C++ project. Copy IDTechSDK.dll and all the other provided SDK .dll files to Debug and Release folders in the Visual C++ project.