

Native (Delphi) callbacks in .NET (C#) COM assembly



م. عصام علي

<http://issamsoft.com>

مقدمة

لنفترض لدينا الحالة التالية:

- تريد كتابة إجراء في بيئة الدوت نت و تجعله متاحاً للغات البرمجة الأصلية (native) مثل دلفي.
- هذا الاجراء يأخذ تابع منادى (callback) كأحد بارامتراتهِ/التابع في هذه الحالة سيكون مكتوباً بلغة أصلية كدلفي/.
- تريد استدعاء تابع الدوت نت هذا من برنامج دلفي.

قبل البدء

بداية ما هو التابع المنادى (callback function) وبالمناسبة هذه هي ترجمتي لأنني لم أجد ترجمة عربية أخرى معتمدة لهذا المصطلح الانكليزي. حسب [الويكيديا](#) التابع المنادى هو عبارة عن مرجعية لقطعة من النص المصدري يمرر كمتحول إلى نص مصدري آخر وهي تقنية شائعة جداً في لغات البرمجة الأصلية مثل سي بلس بلس و دلفي أما في دوت نت فالتقنية الأقرب لهذا المفهوم هي تقنية التوكيل delegate. التعامل مع هذا النوع من التوابيع قد يكون سهلاً عندما تبقى ضمن نطاق اللغات الأصلية مثل سي و دلفي وعندما يبقى استخدامها وتبادلها ضمن النطاق نفسه لكن كيف يمكن تمرير مرجعية لنص مصدري موجود في الذاكرة الحقيقية إلى نص مصدري آخر موجود في الذاكرة المدارة و هي هنا بيئة الدوت نت؟ هذا ما سأحاول تبينه في هذا المقال.

أمر آخر سيواجهنا في السيناريو السابق و هو كيف نجعل تابع دوت نت متاحاً لاستخدامه في لغات البرمجة الأصلية، من المعروف أن هذا الأمر يمكن تحقيقه عن طريق جعل مكتبة الدوت نت net assembly. متاحة لعناصر الكوم- COM visible وهنا لا بد من الأخذ بعين الاعتبار الخطوات التالية:

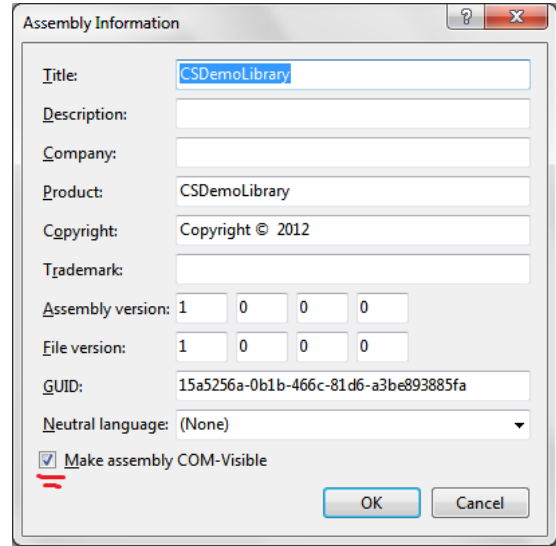
- تسجيل مكتبة الدوت نت كعنصر كوم.
- إضافة المكتبة إلى ذاكرة المكتبات العامة (Global Assembly Cache) GAC في حال أردنا استدعاء المكتبة من مسارات مختلفة.
- يجب إعطاء المكتبة اسماً فريداً (قويّاً) في حال أردنا إضافتها إلى GAG .

قسم الدوت نت

سأستخدم فيجوال ستديو 2010 و سي شارب لبناء هذا القسم.

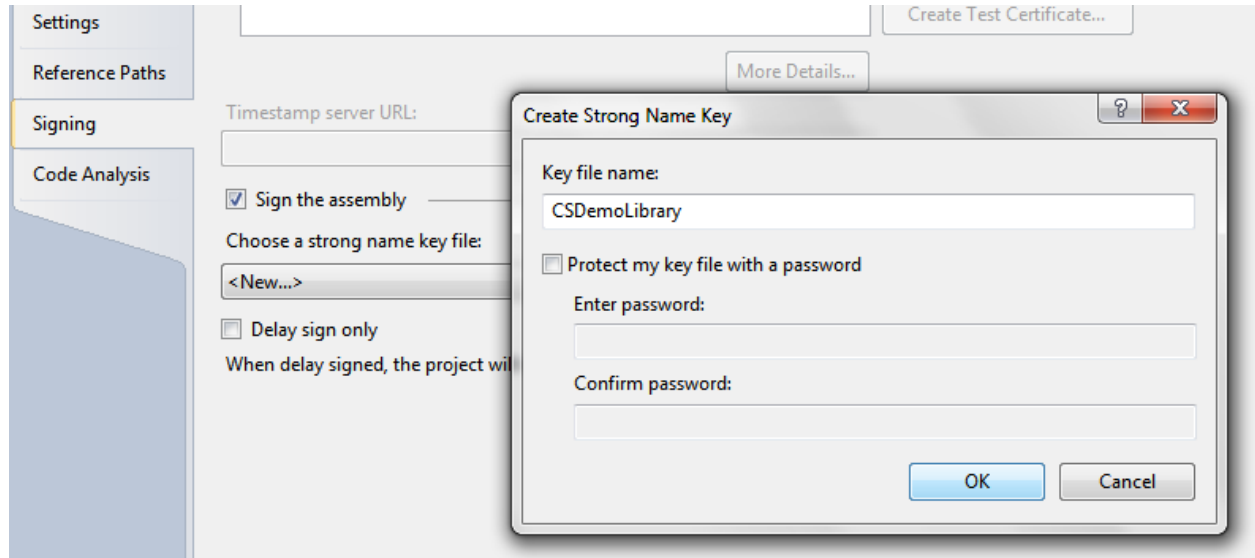
في فيجوال ستوديو قم بإنشاء مشروع مكتبة صنف جديد و سمّها CSDemoLibrary.

من ضمن خصائص المشروع وفي مربع معلومات المكتبة assembly information قم بتفعيل خيار COM-Visible.

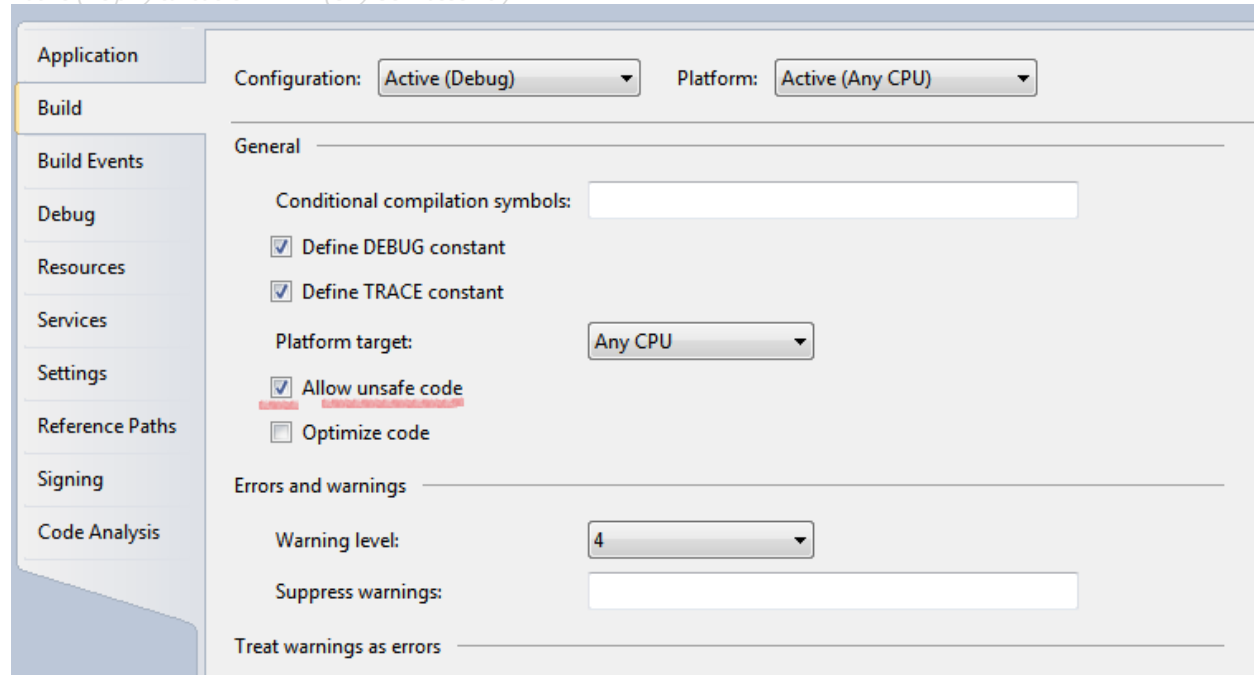


نحن بحاجة إلى إعطاء المكتبة اسماً فريداً إذا كنا سنضيفها إلى الـ GAC لاستخدامها من مسارات مختلفة -يمكنك تجاهل هذه الخطوة إذا كنت ستحفظ هذه المكتبة في نفس المسار لتطبيق دلفي- ولإعطاء المكتبة هذا الاسم الفريد:

من لسان التويب signing اختر (Sign The Assembly) ثم من القائمة المنسدلة اختر جديد و ليكن اسم الملف CSDemoLibrary.



ولأننا نريد تمرير مؤشر من برنامج دلفي إلى هذه المكتبة فهناك خطوة إضافية لابد منها وهي (السماح بالشفرة غير الآمنة) unsafe code وذلك بتفعيل الخيار (Allow unsafe Code) في لسان التويب Build مزيد من المعلومات عن المؤشرات في الدوت نت [في هذا المقال](#)



الآن أصبحنا جاهزين لكتابة الكود، احذف الصف الافتراضي في المشروع و أضف صف جديد و سمّه MyClass. من أجل التبسيط سوف نعتمد في هذا المثال تابع منادى بسيط يأخذ بارامترين فقط. في دلفي سيكون شكل هذا التابع كما يلي:

```
procedure callback(intParam: Integer; strParam: pChar); stdcall;
```

المكافئ لهذا التابع في الدوت نت سيكون توكيل delegate يأخذ الشكل التالي:

```
public delegate void NativeCallback(Int32 intParam,  
[MarshalAs(UnmanagedType.LPWSTR)] string strParam);
```

لقد استخدمت وسم [MarshalAs(UnmanagedType.LPWStr)] للبارمتر النصي لتحويله إلى يونيكود عند بدله مع اللغات الأصلية native ويمكنك إغفال هذا الوسم في حال أردت التعامل مع ال Anstring لكن عندها يجب تغيير نوع البارمتر النصي في تابع الدلفي ليصبح Pansichar.

الآن نستطيع كتابة تابع الدوت نت، مع مراعاة أننا يجب أن نجعل الصف MyClass متاحاً لـ COM وذلك بوسمه بالوسم ComVisible(true) مع نص guid جديد وبما أنه سيكون فريداً سيشكل ما يشبه البصمة الخاصة بعنصر الكوم هذا. الأمر الآخر المهم هنا ولكون هذا التابع سيتعامل مع المؤشرات فيجب وسمه بالوسم unsafe (راجع مقالي السابق) وفي مايلي النص الكامل للصف MyClass:

```
using System;  
using System.Runtime.InteropServices;  
using System.Threading;  
  
namespace CSDemoLibrary  
{  
    public delegate void NativeCallback(Int32 intParam,  
[MarshalAs(UnmanagedType.LPWSTR)] string strParam);  
  
    [ComVisible(true),  
    GuidAttribute("3A65D04A-3F2F-4CB3-B65A-8D402B8C64CE")]
```

```

public class MyClass
{
    public unsafe int Process(
        int intValue,
        IntPtr callbackPointer,
        int intParam,
        string strParam)
    {
        IntPtr ptr = new IntPtr(callbackPointer);
        NativeCallback callbackMethod =
        (NativeCallback)Marshal.GetDelegateForFunctionPointer(ptr,
        typeof(NativeCallback));

        callbackMethod(intParam, strParam);
        Thread.Sleep(1000);

        callbackMethod(25, "الأولى المرحلة");
        Thread.Sleep(1000);
        callbackMethod(50, "الثانية المرحلة");
        Thread.Sleep(1000);
        callbackMethod(75, "الثالثة المرحلة");
        Thread.Sleep(1000);
        callbackMethod(100, "الرابعة المرحلة");

        return intValue * 10;
    }
}

```

في النص السابق: التابع Process يحاكي عملية تستغرق فترة طويلة من الزمن (مثل جلب كمية كبيرة من البيانات من الانترنت) مع إعلام التطبيق الطالب بشكل مستمر بحالة العملية أو بمقدار التقدم. في مثالنا هذا ومن أجل التبسيط فقط سنقوم بتوقيف التنفيذ لمدة ثانية واحدة في كل مرة وذلك لمحاكاة عملية تستغرق وقتاً طويلاً و تعيد في نهايتها رقم كنتيجة للعملية. في الكود السابق لدينا callbackPointer هو مؤشر للتابع المنادى، ولدينا (intParam, strParam) هي بارامترات ذلك التابع، هذين البارامترين يمكن استخدامهما كقيم ابتدائية وفي معظم السيناريوهات لن يكون هناك حاجة لقيم ابتدائية ولكي ابقيت عليهم هنا لشرح كيفية تمرير بارامترات التوابع من البيئة الأصلية إلى تابع الدوت نت.

GetDelegateForFunctionPointer هو المفتاح في هذه المسألة كلها حيث يقوم هذا التابع بتحويل مؤشر لتابع أصلي إلى التوكيل delegate المكافئ له وهناك مزيد من المعلومات عن هذا التابع يمكن أن تجدها هنا <http://msdn.microsoft.com/en-us/library/system.runtime.interopservices.marshal.getdelegateforfunctionpointer%28v=vs.85%29.aspx>

تسجيل مكتبة الدوت نت

لتجنب اشكالية المسارات في الدوس سأستخدم هنا (Visual Studio Command Prompt) وهناك اختصار لهذه الاداة تجده في المسار التالي:

(start menu-> All programs-> Microsoft Visual Studio 2010-> Visual Studio Tools)

باستخدام هذه الاداة يمكنك تسجيل مكتبة الدوت نت كعنصر كوم باتباع الخطوات التالية:

شغل Visual Studio Command Prompt /as administrator/ وانتقل إلى المجلد الذي يحتوي المكتبة CSDemoLibrary.dll باستخدام تعليمات الدوس.

إذا كنت ستستخدم CSDemoLibrary.dll من نفس مسار برنامج دلفي فقط فلا داعي لاضافة المكتبة الى GAC فيما عدا ذلك يجب إضافة المكتبة الى الـ GAC ولتحقيق هذا سنستخدم الاداة gacutil مع البارامتر -i كما يلي:

```
gacutil -i CSDemoLibrary.dll
```

لتسجيل المكتبة كعنصر COM سنستخدم أداة regasm كما يلي:

```
regasm CSDemoLibrary.dll
```

ملاحظة:

لإزالة المكتبة من الـ GAC نستخدم الأمر التالي:

```
gacutil -u CSDemoLibrary
```

لإلغاء تسجيل المكتبة نستخدم الامر:

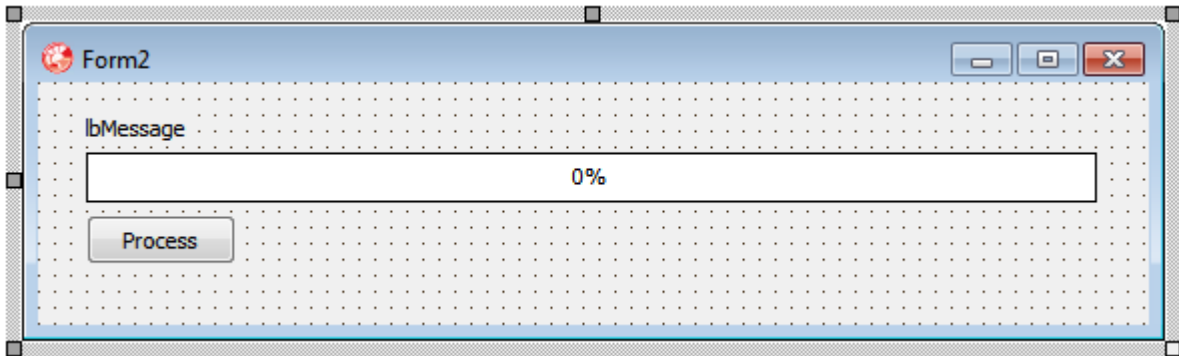
```
regasm -u CSDemoLibrary.dll
```

قسم الدلفي:

سأستخدم في هذا القسم دلفي 2010.

في دلفي 2010 انشئ مشروع تطبيق VCL جديد

أضف (TLabel, TButton, TGauge) إلى النموذج الرئيسي



في الكود خلف النموذج الرئيسي أضف ComObj إلى قسم uses

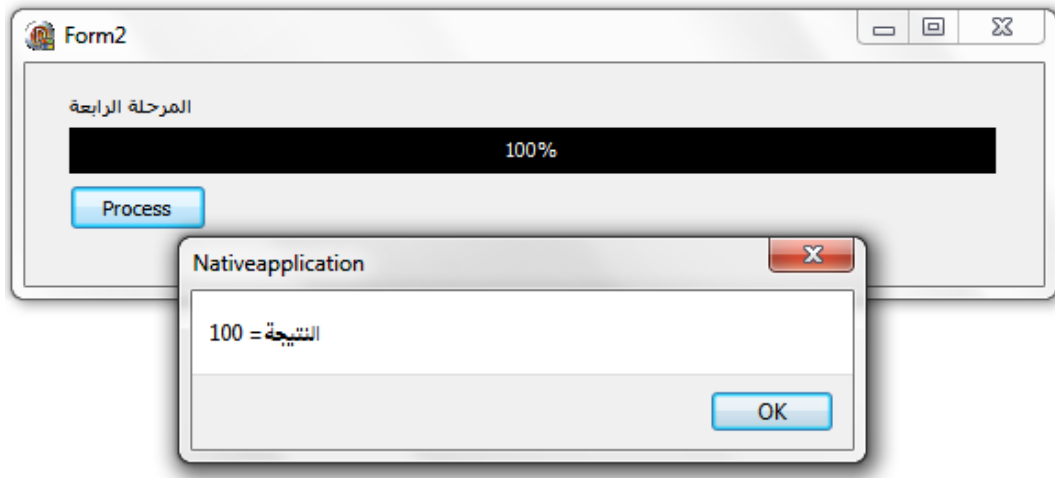
الآن لنقوم بإنشاء تابع بسيط في دلفي سيكون هو التابع المنادى الذي سنرسله (بالأحرى نرسل مرجعيته) إلى مكتبة الدوت نت لتقوم بدورها باستدعائه لإظهار وتحديث حالة العملية للمستخدم.

```
procedure callback(intParam: Integer; strParam: pChar); stdcall;  
Begin  
  Form2.Gauge1.Progress := intParam;  
  Form2.lbMessage.Caption := strParam;  
  Application.ProcessMessages;  
End;
```

سأستخدم في هذه المشروع تقنية الاسناد المتأخر (late binding) في انشاء عنصر الكوم, لذلك سنغير كود الضغط على الـ btnProcess كمايلي:

```
procedure TForm2.btnProcessClick(Sender: TObject);
var
  oleObject: OleVariant;
begin
  try
    oleObject := CreateOleObject('CSDemoLibrary.MyClass');
    ShowMessage('النتيجة= ' + IntToStr(oleObject.Process(10, LongInt(@callback),
0, 'القيمة الابتدائية')));
  except on E: Exception do
    ShowMessage('COM Error: ' + #13 + #10 + e.Message);
  end;
end;
```

وبهذا يصبح لدينا تطبيق دلفي يستدعي عملية طويلة في الدوت نت التي بدورها تقوم بتنبيه تطبيق دلفي حول حالة العملية و مقدار التقدم الحاصل باستخدام التابع المنادى المكتوب في دلفي.



مراجع

http://en.wikipedia.org/wiki/Callback_%28computer_programming%29

<http://edn.embarcadero.com/article/32754>

<http://msdn.microsoft.com/en-us/library/system.runtime.interopservices.marshal.getdelegateforfunctionpointer%28v=vs.85%29.aspx>

ملاحظة:

نشرت النسخة الانكليزية لهذا المقال في موقع [codeproject](http://codeproject.com)