

# 程序猿的程式寫作記錄

```
CODE { display: block; /* fixes a strange ie margin bug */ font-family: Courier New; font-size: 8pt; overflow:auto; background: #f0f0f0
url(http://klcintw.images.googlepages.com/Code_BG.gif) left top repeat-y; border: 1px solid #ccc; padding: 10px 10px 10px 21px; max-height:200px; line-height:
1.2em; }
```

2016年6月5日 星期日

## [BCB] 創建dll和使用dll方法

DLL的原文為Dynamic Link Library，中文稱為動態連結函式庫。

使用dll的好處可減少程式佔用記憶體空間，當程式有需要時才會將dll所需要的資源載入到memory中。除了記憶體的優勢外，當開發大型程式時，可將程式所需的功能拆成一個個component，將這些component的輸入和輸出定義出來後，即可分發給各個工程師撰寫完成。未來若發現某個component有問題時，維護也只需針對該component做維護。

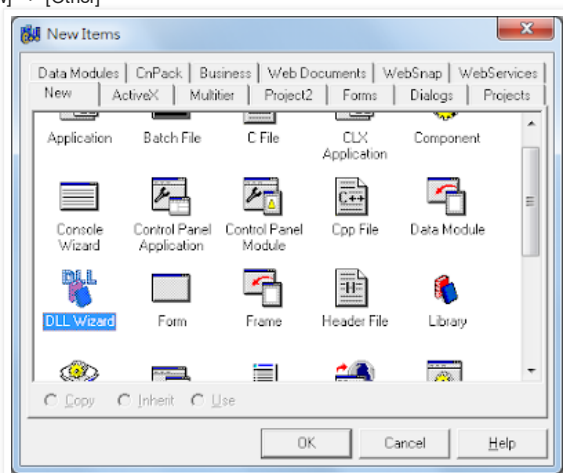
開發dll需要注意的事情是你要如何寫[entry point]，意思是程式呼叫dll檔案時的切入點。而用BCB來開發動態函式庫時有兩種entry point寫法，一種為使用BCB的entry point；另外一種則是用visual studio C++的entry point。

底下會針對上述 2種entry point的方式分別做說明。

[BCB Type entry point]

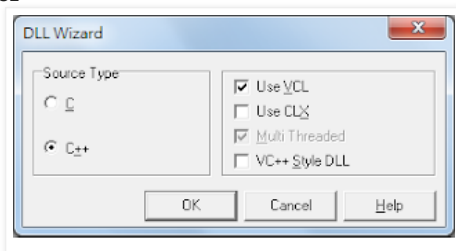
1. 建立一個dll的開發專案

[File] -> [New] -> [Other]



選擇DLL Wizard

2. 選擇c++和Use VCL



按[OK]繼續。

3. BCB自動生成以下程式碼

```
//-----
#include
#include
#pragma hdrstop
//-----
// Important note about DLL memory management when your DLL uses
// static version of the RunTime Library:
//
// If your DLL exports any functions that pass String objects (o:
// classes containing nested Strings) as parameter or function r:
//
```

追蹤者

追蹤者 (0)

網誌存檔

▼ 2016 (6)

► 七月 (1)

▼ 六月 (4)

[視覺2點定位概述](#)

[\[BCB\] 靜態函式庫的開發](#)

[\[BCB\] 如何在dll中建立一個windows form 程式](#)

[\[BCB\] 創建dll和使用dll方法](#)

► 五月 (1)

## 4. 撰寫程式碼

```

long __declspec(dllexport) __stdcall RPC_ADDRESS_CHANGE_FN(void *)
{
    long sum = 0;

    for(m; m<=n; m++)
    {
        sum += m;
    }

    return sum;
}

```

\_\_declspec(dllexport)是為了讓compiler知道我們現在要寫的是一個DLL 檔案。  
\_\_stdcall則是一種函式的呼叫方式。

## 5. 編譯程式碼。

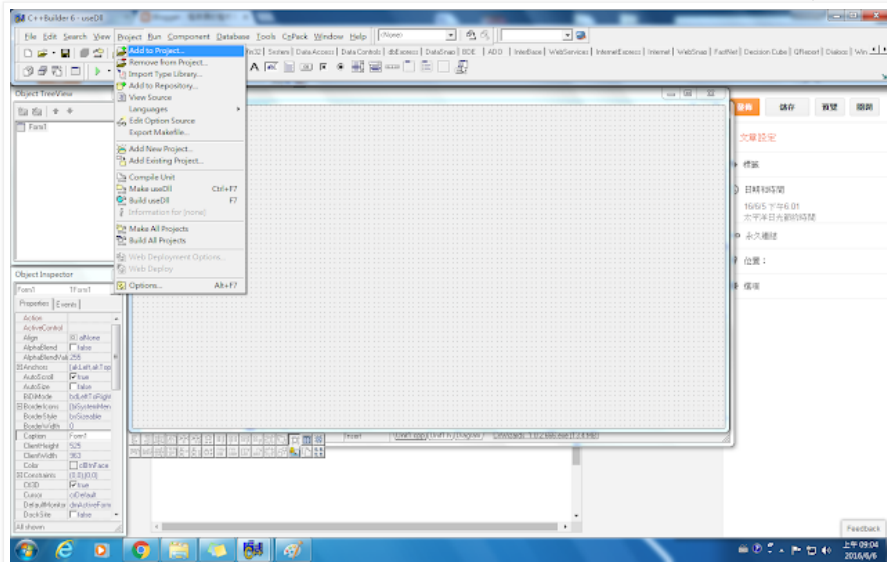
編譯時會依據專案名稱產生對應的dll和lib檔案。以該範例吾人想產生 1個dllexample的dll，則此時必須把專案名稱要更改為dllexample。  
 按compiler後，則可在專案目錄下找到dllexample.dll和dllexample.lib兩個檔案。  
 目前為此已完成dll的製作工作了，接下來就是要如何使用這個dll檔案。

=====

## [使用dll]

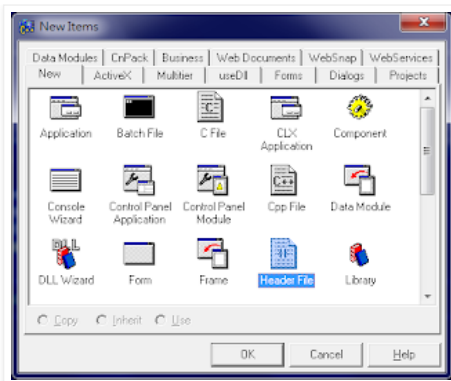
1. 創建一個專案名為useDll。
2. 將剛才製作的dllexample.dll和dllexample.lib檔案複製到該專案下。
3. 將lib檔案加入到該專案中。

[Project] -> [Add Project]



選擇檔案類型為lib，並選擇dllexample.lib檔案後按[開啟舊檔]來將該libray加到專案中。

4. 產生 1個header file定義dll檔案內的函式。  
 [File] -> [New] -> [Other]



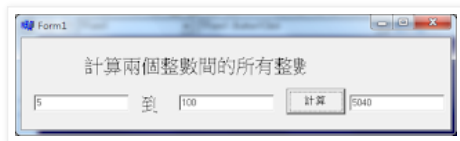
並將檔案儲存為user.h。

5. 在user.h中定義dll 函式。

```
int __declspec(dllexport) __stdcall BCBType_count(int m, int n);
```

6. 在專案中記得要加入#include "user.h"

7. 撰寫專案界面。



usedll.h

```
//-----
#ifndef Unit1H
#define Unit1H
//-----
#include
#include
#include
#include
#include "user.h"
//-----
```

usedll.cpp

```
//-----
#include
#pragma hdrstop
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
```

張貼者：Tommy 於 下午6:01



沒有留言:

張貼留言

輸入您的留言...

留言身分： Google 帳戶 ▼

發布

預覽

[較新的文章](#)

[首頁](#)

[較舊的文章](#)

訂閱： [張貼留言 \(Atom\)](#)

技術提供：[Blogger](#).