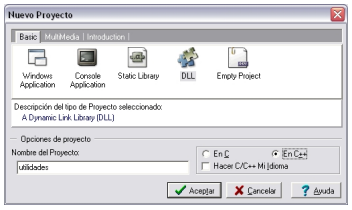
En esta ocasión vamos a ver como crear una dll sencilla y luego como podemos usarla en un programa de ejemplo.

Vamos a usar el compilador Dev-C++ 4.9.9.2, si no lo tienes descárgalo primero de la sección Compiladores de este sitio.

Empecemos, como siempre paso a paso:

Creación de la dll

1. Primero que nada crearemos una carpeta de nombre utilidades en C:Dev-Cppbin en la cual iremos guardando todos los archivos del proyecto. Abrimos el Dev-C++ y vamos a Archivo
->
Nuevo
->
Proyecto
, seleccionamos
DLL
y el nombre del proyecto será
utilidades
tal como se muestra:




2. Guardamos todos los ficheros en la carpeta creada anteriormente y dejamos los nombres por defecto para los ficheros, vemos que se han creado los ficheros: dllmain.cpp y dll.h como se muestra:

3. Ahora cambiamos el código del fichero dll.h por el siguiente:

```
codeDivStart()/* dll.h */

#ifndef _DLL_H_
#define _DLL_H_

#if BUILDING_DLL
# define DLLIMPORT __declspec (dllexport)
#else /* Not BUILDING_DLL */
# define DLLIMPORT __declspec (dllimport)
#endif /* Not BUILDING_DLL */



#endif /* _DLL_H_ */
```

Este es un código que por lo general hay que dejarlo así

4. Ahora cambiamos el código del fichero dllmain.cpp por el siguiente:

```
codeDivStart()/* dllmain.cpp */

#include "dll.h"
#include <windows.h>

extern "C" _declspec(dllexport)
int suma(int a, int b)
{
    return a + b;
}

int longitud(char *cadena)
{
    int len = 0;
    for(int i=0; ; i++)
    {
        if(cadena[ i ] == '\0')
          break;

        len ++;
    }
```

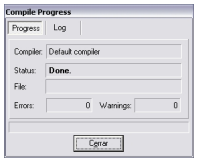
```
    return len;
}

extern "C" _declspec(dllexport)
int es_palindroma(char *cadena)
{
   int i;
   char invertida[50];

   /* Halla la cadena invertida */
   for(i=0; i            invertida[ i ] = cadena[longitud(cadena)-i-1];

   /* Coloca caracter de fin de cadena */
   invertida[ i ] = '\0';

   /* Realiza la comparacion */
   if(!strcmp(cadena, invertida))
      return 1; /* es un palindromo */

   return 0;  /* no es un palindromo */
}


BOOL APIENTRY DllMain (HINSTANCE hInst     /* Library instance handle. */ ,
                       DWORD reason        /* Reason this function is being called. */ ,
                       LPVOID reserved     /* Not used. */ )
{
    switch (reason)
    {
      case DLL_PROCESS_ATTACH:
        break;

      case DLL_PROCESS_DETACH:
        break;

      case DLL_THREAD_ATTACH:
        break;

      case DLL_THREAD_DETACH:
        break;
    }

    /* Returns TRUE on success, FALSE on failure */
    return TRUE;
}
```

Como podemos mirar, las funciones suma y es_palindroma están precedidas por la siguiente línea de código:

codeDivStart()extern "C" _declspec(dllexport)

Mediante esta línea decimos que las funciones definidas a continuación serán públicas, es decir que pueden ser usadas por cualquier que tenga la dll resultante. La función longitud es privada, es decir que sólo puede ser usada al interior de la dll, se trata de una función de utilería.

5. Bien, ahora vamos a compilar, para ello, simplemete presionamos CTRL+F9 y podemos ver el resultado de la compilación:

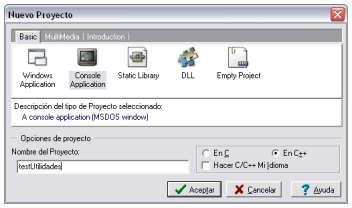



6. Bien con esto ya hemos creado nuestra dll. Si vamos al Explorador de Windows a nuestra carpeta de trabajo, podemos ver que se han creado algunos ficheros, de todos estos, la dll es obviamente : utilidades.dll, los demás ficheros tienen el fuente de la dll. Si queremos distribuír nuestra dll sólo hace falta entregar este fichero y ningún otro más.

Uso de la dll

1. Ahora que tenemos la librería o dll resultante, vamos a crear un programa de prueba. Primero que nada crearemos una carpeta de nombre testUtilidades en C:Dev-Cppbin en la cual iremos guardando todos los archivos de este proyecto de prueba. Copiamos el archivo utilidades.dll
en la carpeta
testUtilidades
. Abrimos el Dev-C++ y vamos a
Archivo

->
Nuevo
->
Proyecto
, seleccionamos
Console Application
y el nombre del proyecto será
testUtilidades
tal como se muestra:




Reemplazamos el código inicial por el siguiente:

```
codeDivStart()#include <cstdlib>
#include <iostream>
#include <windows.h>

using namespace std;

int suma_int(int, int);
int es_cadena_palindroma(char *);

typedef int (WINAPI*cfunc1)(int, int);
cfunc1 ConexionDLL1;

typedef int (WINAPI*cfunc2)(char *);
cfunc2 ConexionDLL2;

int main(int argc, char *argv[])
{
    int a, b, resultado;
    char cadena[50];
    cout > a;
    cout > b;
    resultado = suma_int(a, b);
    cout
```