



Forum: WD 10

Topic: PB d'utilisation d'une DLL

Subject: Re: PB d'utilisation d'une DLL

Posté par: Madinico

Contribution le : 23/5/2007 8:45:26

D'accord, je te donne tout d'abord le code WD d'appel de la DLL.

Code_retour est un entier

hInst est un entier

```
hInst = ChargeDLL("C:Dev-CppUtilitaireBCP.dll")
```

```
SI hInst = 0 ALORS
```

```
  Erreur("Erreur lors du chargement")
```

```
SINON
```

```
  Code_retour= AppelDLL32("UtilitaireBCP.dll", "exportBCP@40",  
  "DSN", "Serveur", "base", "user", "password", "C:\Fichier_SAV\SAV\export.txt", "C:\Fichier_SAV\SAV\error.txt", "SELECT * FROM table", "", "")
```

```
SI Code_retour<0 ALORS
```

```
  Erreur("Erreur du BCP", Code_retour, ErreurInfo())
```

```
SINON
```

```
  Info("BCP réussie")
```

```
FIN
```

```
DéchargeDLL(hInst)
```

```
FIN
```

FinProgramme()

AppelDLL32 et API donne le même résultat.

Voici une partie du code de la DLL.

```
#include "UtilitaireBCP.h"
```

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
#include <windows.h>
```

```
#include <sql.h>
```

```
#include <sqlext.h>
```

```
#include <odbcss.h>
```

```
#include <odbcinst.h>
```

```
#include <string.h>
```

```
BOOL WINAPI DIIMain (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;
}

int WINAPI __stdcall exportBCP(char* DSN,char* server,char* database,char* user,char*
password,char* exportFile,char* errorFile,char* request,char* separator,char* endOfLine){
    SQLHENV env;
    SQLHDBC dbc;
    RETCODE retcode;
    DBINT    nRowsProcessed;
    // Résultat de la configuration des sources de données ODBC
    bool resultat;
    char* temp2;
    char* temp;
    int mlen;
    temp=new char[500];
    temp2=new char[100];

    //Construction de la chaine de création de la source de données
    sprintf(temp,"DSN=%s* Server=%s* Database=%s*
Trusted_Connection=No*",DSN,server,database);
    mlen = strlen(temp);
    for (int i=0; i<mlen; i++)
    {
if (temp[i] == '*')
        temp[i] = ' ';
    }

    //Création d'une source de données ODBC
    resultat=SQLConfigDataSource(NULL,ODBC_ADD_DSN,"SQL Server",(LPCSTR)temp);
    if(resultat==true){
        //printf("Creation DSN reussien");
    }
}

```

```

else{
    //printf("Echec de la creation DSNn");
    return -1;
}

//Description du Handle d'environnement
retcode = SQLAllocHandle(SQL_HANDLE_ENV,NULL,&env);
if (retcode==SQL_SUCCESS || retcode==SQL_SUCCESS_WITH_INFO){
    /* Set the ODBC version environment attribute */
    retcode =
SQLSetEnvAttr(env,SQL_ATTR_ODBC_VERSION,(SQLPOINTER)SQL_OV_ODBC3,SQL_IS_INTEGER);
    //Description du Handle de connexion
    retcode = SQLAllocHandle( SQL_HANDLE_DBC, env, &dbc );
    if (retcode==SQL_SUCCESS || retcode==SQL_SUCCESS_WITH_INFO){
        retcode = SQLSetConnectAttr(dbc, SQL_COPT_SS_BCP,(SQLPOINTER)SQL_BCP_ON
,SQL_IS_INTEGER);
        //Connexion à une base de données SQL
        retcode =
SQLConnect(dbc,(SQLCHAR*)DSN,SQL_NTS,(SQLCHAR*)user,SQL_NTS,(SQLCHAR*)password,
SQL_NTS);
        if (retcode==SQL_SUCCESS || retcode==SQL_SUCCESS_WITH_INFO){
            //printf( "Successfully connectedn" );
            SQLDisconnect(dbc);
        }
        else
        {
            return -2;
        }
    }
    //Libération mémoire du Handle de connexion
    SQLFreeHandle( SQL_HANDLE_DBC, dbc );
}
//Libération mémoire du Handle d'environnement
SQLFreeHandle( SQL_HANDLE_ENV, env );
sprintf(temp2,"DSN=%s*",DSN);
mlen = strlen(temp2);
for (int i=0; i<mlen; i++)
{
if (temp2[i] == '*')
    temp2[i] = ' ';
}
//Suppression de la source de données ODBC
resultat=SQLConfigDataSource(NULL,ODBC_REMOVE_DSN,"SQL Server",(LPSTR) temp2);

if(resultat==true){
    //printf("Suppression DSN reussien");
}
else{

```

```
    return -7;
    //printf("Echec de la suppression DSNn");
}
return 0;
}
```

Voilà, il y a tout ce qu'il faut pour simuler le comportement. Il faut juste recréer la DLL. Donc réécrire un fichier header de dll et la recompilée pour avoir le fichier.
L'erreur s'effectue sur l'instruction SQLConnect.

Merci de votre aide.

Cordialement,
Nicolas.