Forum: WD 8.x

Topic: {Script} Controler WinAmp avec WinDev

Subject: Re: {Script} Controler WinAmp avec WinDev

PostÃ© par: Anonyme

Contribution le : 4/6/2004 10:43:21

Je vais faire quelques explications de mon code à la demande de R&B pour les plus débutant d'entre nous (Je le suis un peu donc si je dis des bétises ne pas hésiter à me corriger).
Tout d'abord jai repris du code qui était écrit en C++ sur un pluging pour un jeu qui permettais de piloter WinAmp par le jeu. En voici les sources :

```
// Winamp Ctrl.cpp : Defines the initialization routines for the DLL. //  #include "stdafx.h" #include
<afxdllx.h> #include "..plugin.h" #include "About.h"  #ifdef _DEBUG #define new DEBUG_NEW
#undef THIS_FILE static char THIS_FILE[] = __FILE__; #endif  sPlugin thisplug; CString m_WATitle;
bool flag_notify;  static AFX_EXTENSION_MODULE WinampCtrlDLL = { NULL, NULL };  extern "C"
int APIENTRY DllMain(HINSTANCE hInstance, DWORD dwReason, LPVOID lpReserved) {
 UNREFERENCED_PARAMETER(lpReserved);   if (dwReason == DLL_PROCESS_ATTACH)  {   if
(!AfxInitExtensionModule(WinampCtrlDLL, hInstance))    return 0;   new
CDynLinkLibrary(WinampCtrlDLL);  }  else if (dwReason == DLL_PROCESS_DETACH)  {
  AfxTermExtensionModule(WinampCtrlDLL);  }  return 1;   // ok }  lpsPlugin fnInit(char langue) {
 thisplug.nom="Winamp Ctrl";  thisplug.brief="Permet de controler Winamp sans sortir du jeu.";
 thisplug.flags=PLUGIN_FLAG_ON|PLUGIN_FLAG_ABOUT|PLUGIN_TYPE_OUT|PLUGIN_FLAG_
V2;  thisplug.version=TPP_VERSION;  thisplug.fnSendIn=NULL;  thisplug.fnSendOut=NULL;
 flag_notify=true;   return &thisplug; }  void fnAbout(void) {  CAbout dlgAbout;   dlgAbout.DoModal(); }
void Send_sysmsg(LPCTSTR msg,...) {  int  lg_msg;  CString message;  char *pak;   if
(thisplug.fnSendIn != NULL)  {   va_list args;   va_start(args, msg);   message.FormatV(msg, args);
  lg_msg=message.GetLength();   pak=new char[lg_msg+8];    *(unsigned short *)&pak[0]=0x3F00; //
Message systeme   *(unsigned short *)&pak[2]=0x1E00; // ???   *(unsigned short
*)&pak[4]=0x0300; // ???   pak[6]=(lg_msg/256)&0xff; // Longueur du message   pak[7]=lg_msg&0xff;
  strncpy(&pak[8],message,lg_msg);    // Envoie le message au client
  thisplug.fnSendIn(pak,lg_msg+8,false);   delete pak;  } }  int GetWinampTitle(char *str, int lg) {  char
*title,*p;   *str=0;   HWND hwndWinamp = FindWindow("Winamp v1.x",NULL);   if (hwndWinamp)  {
  title=str;    GetWindowText(hwndWinamp,title,lg);   p = title+strlen(title);   while (p >= title)   {    if
(!strnicmp(p,"- Winamp",8))    break;    p--;   }   if (p >= title)    p--;    while (p >= title && *p == &#039;
&#039;)    p--;    *++p=0;    lg=p-title;  } else   lg=0;   return lg; }  void *fnProcessOut(unsigned char
*buff, DWORD &taille, DWORD &param) {  int offset=0,type,commande=0;   type=*(short *)buff;  if
(type==0x1B00)   offset=13;  if (type==0x1E00)   offset=17;   param=PLUGIN_PAK_NOCHANGE;   if
(flag_notify)  {   CString newTitle;
  newTitle.ReleaseBuffer(GetWinampTitle(newTitle.GetBufferSetLength(2048), 2048));   if
(m_WATitle.Compare(newTitle))   {    m_WATitle=newTitle;    Send_sysmsg("Playing:
%s",m_WATitle);   } }   if (offset)  {   if (!strnicmp((char *)&buff[offset],"wa_play",7))
   commande=40045;   else if (!strnicmp((char *)&buff[offset],"wa_pause",7))    commande=40046;
  else if (!strnicmp((char *)&buff[offset],"wa_stop",7))    commande=40047;   else if (!strnicmp((char
*)&buff[offset],"wa_prev",7))    commande=40044;   else if (!strnicmp((char
*)&buff[offset],"wa_next",7))    commande=40048;   else if (!strnicmp((char
*)&buff[offset],"wa_plist",8))    commande=40154;   else if (!strnicmp((char
*)&buff[offset],"wa_runcd",8))    commande=40323;   else if (!strnicmp((char
*)&buff[offset],"wa_chgon",8))   {    flag_notify=true;    Send_sysmsg("Notification de changement de
```

```
piste activée");    param=PLUGIN_PAK_DELETE;   }   else if (!strnicmp((char
*)&buff[offset],"wa_chgoff",9))   {    flag_notify=false;    Send_sysmsg("Notification de changement de
piste désactivée");    param=PLUGIN_PAK_DELETE;   }   else if (!strnicmp((char
*)&buff[offset],"wa_title",8))   {
  m_WATitle.ReleaseBuffer(GetWinampTitle(m_WATitle.GetBufferSetLength(2048), 2048));
  Send_sysmsg(m_WATitle);    // Le pak de doit pas etre envoyé au serveur
  param=PLUGIN_PAK_DELETE;   }    if (commande)   {    HWND hwndWinamp =
FindWindow("Winamp v1.x",NULL);     if (hwndWinamp)
   PostMessage(hwndWinamp,WM_COMMAND, commande, 0); // preferable a SendMessage en
cas de plantage de winamp     param=PLUGIN_PAK_DELETE;   } }   return buff; }
```

Tout d'abord il faut décortiquer le message, c'est à dire répérer les fonctions qui opère sur WinAmp. Par exemple la fonction GetWinampTitle():

```
 int GetWinampTitle(char *str, int lg) {  char *title,*p;   *str=0;   HWND hwndWinamp =
FindWindow("Winamp v1.x",NULL);   if (hwndWinamp)  {   title=str;
 GetWindowText(hwndWinamp,title,lg);   p = title+strlen(title);   while (p >= title)   {    if (!strnicmp(p,"-
Winamp",8))     break;    p--;   }    if (p >= title)    p--;    while (p >= title && *p == &amp;#039; &amp;#039;)    p--;
 *++p=0;    lg=p-title;  } else   lg=0;   return lg; }
```

On peut voir directement cette ligne : HWND hwndWinamp = FindWindow("Winamp v1.x",NULL);
Cette fonction permet de rechercher une fenêtre. Elle ne fonctionne que sur les fenêtres principales (pas les filles des MDI).
lpClassName est le nom de la classe qui identifie la fenêtre (rarement connu et donc souvent égal à une chaîne vide).
lpWindowName est le titre (complet) de la fenêtre.
La fonctionne retourne alors le handle de la fenêtre, en cas d'échec, elle renvoie 0.
Cependant dans nôtre exemple on connait la classe (qui nous est renseigner dans la documentation SDK de WinAmp), mais pas le titre, il faut savoir qu'il changera à chaque musique, puisque le titre y est dedans ;)
Donc on va commencer par transcrire cette fonction en fonction Windev, pour cela on peut faire appel à plusieur chose soit avec AppelDLL32, soit Api, enfin avec toutes les fonctions de WinDev qui appel une fonction dans une Dll. Ensuite il nous faut trouver l'alias de la fonction pour pouvoir l'éxécuter par Windev; Dans mon cas j'ai était farfouiller dans la Doc de Windev ("API Windows, Fonctions de l'API Windows") en faisant un recerché sur la valeur : "FindWindow" et j'ai trouvé FindWindowA (donc visiblement les Alias sont tout bètement le nom suivit d'un A comme alias.)
Aprés reste plus qu'à l'écrire.

hwndWinamp est un entier = API("user32.dll","FindWindowA","Winampv1.x",Null)
Cette fonction nous fournira l'id de la fenêtre de WinAmp en cours d'éxécution, car si WinAmp n'est pas lancé on ne récupéra aucune id.

Aprés si on désire on pourrait récupérer le titre en cours, je le met de coté et le fairai plustard, on va partir sur le PostMessage, voici l'extrait de la documentation de Windev :
Envoie un message Windows à un champ ou à une fenêtre. Ce message est mis dans la file d'attente des messages Windows (l'appel de cette fonction n'est pas bloquant : le programme continue de s'exécuter). La fonction SendMessage permet d'envoyer directement le message.

Remarque : Cette fonction est un appel à l'API Windows PostMessage. Elle permet de réaliser des opérations de bas niveau sur les champs et les fenêtres. Pour connaître la signification de chaque message et de ses paramètres, il faut se reporter à l'API Windows (WIN32).

Attention : L'usage inapproprié de cette fonction peut conduire à des problèmes graves dans votre application et/ou à des dysfonctionnements de l'interface graphique.

Donc on voit tout dabord qu'il faut déclarer dans son projet ceci : **EXTERNE "WINCONST.wl"**
Ceci va nous servir à dire quel type de message on va envoyer, une commande, une clik sur un bouton ,ect...
Dans notre cas cela sera une commande.

operantion est un booléen =PostMessage(hwndWinamp, WM_COMMAND, commande, 0)
Donc comme vous pouvez voir on envoit au Handle de la fenêtre WinAmp sous forme de commande une donnée, c'est donnée nous sont renseigner par encore le documentation de WinAmp qui nous donne l'id pour faire les fonctions suivante, Jouer, next & cie...
Par exemple : WA_Pause est un entier = 40046
Donc si on envoit la commande 40046 WinAmp stopera sa zolie mélodie :)

J'espère que cela à était un minimum compréhensible ! Je suis pas professionel mais amateur ^^