

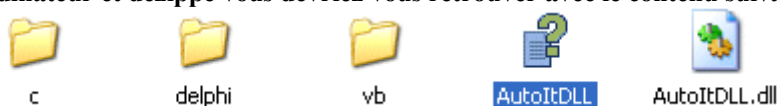
## Automatiser des Applications en Delphi avec la DLL AutoIt

### DESCRIPTION DE CETTE BIBLIOTHEQUE

Je vais vous présenter dans cette fiche pratique un outil fort sympathique, qui permet de prendre le contrôle d'une application externe. Cette librairie est un freeware et elle est développée par la société [HiddenSoft](#). La version DLL permet d'intégrer directement les fonctions dans les fichiers programmes de votre application. Sinon AutoIt existe sous forme de langage de scripts et la version téléchargeable est la version 2. Une version 3 plus aboutie est à l'étude qui permet de réaliser sa propre interface graphique (A voir absolument). Celle-ci est un outil permettant d'automatiser un certain nombre d'actions, comme l'appui sur une touche de clavier, les événements liés à la souris (clique, double clique, etc...), mais aussi la gestion des fenêtres (minimiser, agrandir, remplissage des zones de texte, gestion du temps d'attente, etc...) . Donc cette bibliothèque ne se limite pas à lancer un exe, c'est un outil assez complet. Je vais vous expliquer maintenant comment cette bibliothèque fonctionne sous Delphi (eh oui! il existe bien une version Delphi, et bien sûr une version C++).

### LE TELECHARGEMENT DE LA BIBLIOTHEQUE

Avant de commencer il faut télécharger la bibliothèque pour pouvoir l'utiliser. Pour ce faire il faut se rendre sur le site de Hiddensoft. Il vous faut juste la version DLL. Donc rendez-vous directement sur la page de téléchargement, et téléchargez juste le fichier zippé AutoItDII. Une fois transféré sur votre ordinateur et dézippé vous devriez vous retrouver avec le contenu suivant:



Maintenant que vous avez tous les fichiers nécessaires pour travailler il vous faut maintenant déplacer les



fichiers suivants : AutoItDLL.dll AutoItDLL

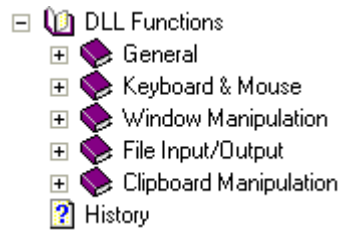
dans le répertoire où se trouve les fichiers sources de votre application. Une fois cette opération effectués, il faut inclure le fichier AutoItDII.pas dans un fichier unité .pas de votre application. Pour ce faire il faut utiliser ceci :

**uses AutoItDII;**

Ceci fait il vous est possible d'entreprendre les travaux à effectuer pour automatiser une application. Je vais faire un petit tour d'horizon pour vous présenter les fonctions clé de cette bibliothèque. Pour plus de détails, reportez vous au fichier d'aide qui donne de très nombreuses informations sur l'utilisations de cette bibliothèque.

### LES FONCTIONS ESSENTIELLES DE CETTE BIBLIOTHEQUE

Je vais vous faire un inventaire des fonctions que j'ai le plus utilisé personnellement. Mais pour avoir une vision plus large de la richesse de cette bibliothèque, il est important de consulter le fichier d'aide associé. Pour les reconnaître ce n'est pas très compliqué, elle porte toutes le préfixe AUTOIT\_. Pour ce faire, c'est simple il suffit d'ouvrir le fichier d'aide associé AutoItDII.chm et de regarder au niveau de DII fonctions :



Comme vous pouvez le constater les fonctions sont rangées par thèmes, ce qui est plus aisé pour s'y retrouver.

## UTILISATION DES FONCTIONS

Tout d'abord avant de pouvoir utiliser la bibliothèque de fonctions, il faut lancer l'application "cible". Ceci se fait tout simplement en utilisant la fonction WinExec dont je vous rappelle la syntaxe :

**WinExec(nom\_application, options);**  
où l'on a:  
**nom\_application** : nom de l'exécutable  
**options** : dans notre cas on va utiliser l'option SW\_SHOW

Il reste une petite opération à réaliser pour que l'on puisse avoir le focus sur la nouvelle fenêtre application que l'on vient de lancer. Pour ce faire il faut utiliser la fonction suivante :

**FindWindowA(titre\_application, 0);**  
où **titre\_application** est le titre qui apparaît dans la barre de titre

Donc on peut directement employer les fonctions de la bibliothèque sur la fenêtre application que l'on vient d'ouvrir. Nous allons voir maintenant comment il est possible, grâce à cette quirielle de fonctions, de naviguer dans l'application cible et d'en prendre ainsi le contrôle. Pour pouvoir passer d'un objet à l'autre, il faut se souvenir que dans la plupart des cas, cela se fait par l'intermédiaire de la touche de tabulation. Pour envoyer ce type de caractère spécial on va utiliser la fonction AUTOIT\_Send qui permet comme son nom l'indique d'envoyer un caractère à l'application cible et dont la syntaxe est la suivante:

**AUTOIT\_Send(\*Char szLine);**

Comme on peut le constater on va envoyer comme paramètre à cette fonction un pointeur sur une chaîne de caractères. Avec cette fonction il est possible d'envoyer des caractères spéciaux de la façon suivante :

**AUTOIT\_Send(PChar('{TAB}'));** pour simuler le caractère spécial de tabulation  
**AUTOIT\_Send(PChar('{ENTER}'));** pour simuler le caractère spécial de validation  
**AUTOIT\_Send(PChar('{CAPS\_LOCK}'));** pour forcer l'écriture en majuscule/minuscule.

Ce qui est particulièrement intéressant avec celle-ci c'est que l'on peut remplir des objets de l'application cible, comme par exemple les zone de texte, les barres de titres. Cela se fait toujours par l'intermédiaire de la fonction AUTOIT\_Send avec en paramètre un pointeur sur la chaîne de caractère. Comme par exemple :

**AUTOIT\_Send(PChar('DEBUT'));**

Donc nous avons vu que l'on pouvait réaliser un certain nombre de tâche par l'intermédiaire de la fonction AUTOIT\_Send. Cependant cette bibliothèque ne se limite pas à cette seule fonction. En effet il est possible

de réaliser une temporisation pour, par exemple, attendre qu'une application soit complètement chargée. Pour réaliser ceci, il faut utiliser la fonction `AUTOIT_Sleep(temps)` où temps est le temps en millisecondes.

**`AUTOIT_Sleep(5000);`**

Vous possédez maintenant les bases nécessaires pour utiliser cette bibliothèque pour vos propres applications. Pour plus d'informations lisez le fichier d'aide fourni avec le zip, et allez consultez le site de l'auteur.