

HYPertextes MULTIMEDIA - 4 (1ère partie)

Ouvertures

Pierre KESSLER

Cet article ¹ est le dernier de la série consacré à l'aide de Windows et à ses documents hypertextes. Il vise à vous initier aux techniques permettant de donner une ouverture "sur l'extérieur" à vos HLP. Enrichir vos productions de macros, utiliser les DLL de Windows, lancer des aides contextuelles depuis vos logiciels, lier des hypertextes entre eux, pour augmenter votre créativité.

Les réalisations pratiques font référence à mon générateur d'hypertextes GenHelp. Il est disponible gratuitement à la bourse aux disquettes de l'EPI. Nous travaillons actuellement avec une version GenHelp 6.6b pour Word 6 et une version 7.0b pour Word 7.

LES MACROS

Winhelp dispose d'une cinquantaine de macro commandes. Ces macros sont présentées de façon extrêmement détaillée dans HAG, le *Help Authoring Guide*, document de référence hypertexte de Microsoft, livré avec GenHelp. En fait, je n'utilise de façon courante qu'une poignée de ces macros.

Voici la boîte de dialogue Insertion | Lien macro de GenHelp. La liste des macros ne donne que des exemples de syntaxes. Les paramètres sont à modifier manuellement dans la boîte de dialogue.

¹ NB : La suite de cet article paraîtra dans le n° 83 de septembre et traitera des applications de RegisterRoutine : insertion à vos hypertextes d'une section MAP et de fichiers multimédias WAV, MID, AVI, FLI ou FLC.

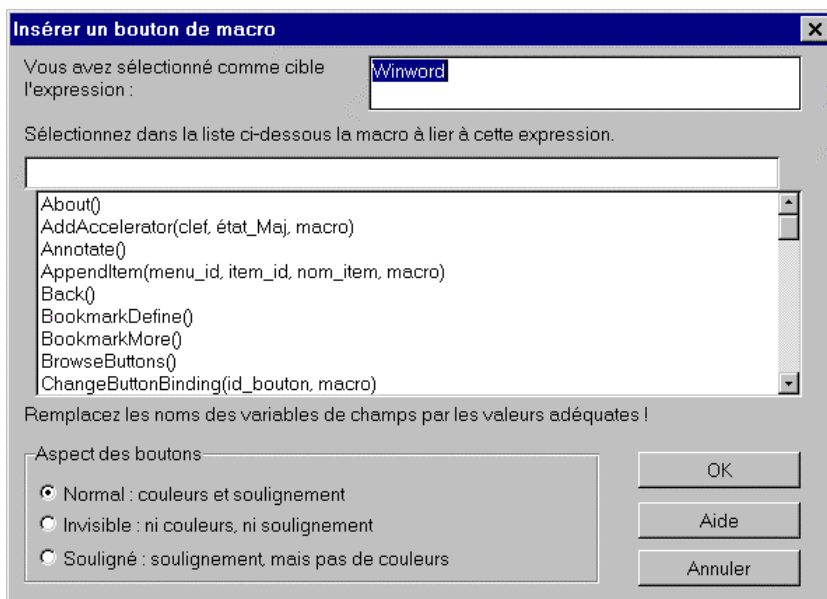


Figure 1 : L'insertion d'une macro dans GenHelp

Eléments généraux de syntaxe

Tous les paramètres (noms de fichiers, de procédures...) doivent être mis entre parenthèses. Si une macro n'a pas de paramètres, il faudra néanmoins lui mettre une paire de parenthèses vides.

Il est parfaitement possible d'enchaîner des macros successives, en les séparant par un point-virgule. Cependant, la longueur totale d'une macro ou d'une chaîne de macros doit toujours rester inférieure à 255 caractères. Voyez l'exemple ci-dessous :

```
ChangeButtonBinding("btn_contents", "JumpID('hgcd.hlp',
`acc_idx_hg`)" ); EnableButton("btn_up") ;
ChangeButtonBinding("btn_up", "JumpID(`cdcd.hlp', `hlpidx_idx_card`)" )
```

Notez enfin que les macros de Winhelp n'ont absolument rien à voir avec celles de Winword.

Les chaînes de caractères

Toutes les chaînes de caractères doivent être entourées de guillemets. Guillemets hauts, doubles, ("chaîne de caractères") auxquels nous

sommes accoutumés, ou "guillemets" simples américains qui vont nous poser quelques problèmes. (autre chaîne')

Vous remarquerez en effet que si le guillemet simple fermant correspond à notre apostrophe, sur la même touche que le chiffre 4, le guillemet simple ouvrant ne figure pas sur notre clavier AZERTY. Je l'obtiens par la combinaison de touches ALT+96.

L'utilisation conjointe des deux types de guillemets ci-dessus va se révéler indispensable dès lors que nous aurons à traiter des paramètres chaîne emboîtés, ce qui arrive sans arrêt dès qu'il va s'agir de lancer un logiciel externe.

Considérons l'exemple ci-dessous. Cette macro crée un bouton appelé Chrono dans la barre, qui utilise la macro ExecProgram comme paramètre. Au clic sur le bouton, le petit logiciel horloge, des accessoires de Windows, sera lancé. Comme la macro ExecProgram exige elle aussi un paramètre chaîne, celui-ci devra être encadré de guillemets américains simples. Si des guillemets doubles encadrent une macro, les chaînes que comprend celle-ci doivent être encadrés de guillemets simples.

```
CreateButton("btn_time", "&Chrono", "ExecProgram('clock', 0)")
```

Utiliser des caractères particuliers dans les chaînes

Un certain nombre de caractères ne peuvent pas être utilisés directement dans les chaînes passées comme paramètres aux macros. Il s'agit en particulier des guillemets doubles ("), du guillemet simple ouvrant ('), de l'apostrophe (') et de l'antislash (\). Tous ces caractères devront être précédés d'un antislash. L'antislash par exemple devra être redoublé. Le chemin \\ROOT\PROJECT\SUBDIR\MYHELP.HLP d'un hypertexte s'écrira donc dans le corps d'une macro :

```
JumpContents("\\\\root\\project\\subdir\\myhelp.hlp")
```

De même pour un nom de bouton contenant une apostrophe :

```
CreateButton("btn_fin", "&Quitter l'application", "Exit()")
```

Valeurs numériques

Help accepte les nombres indifféremment sous forme décimale ou hexadécimale. Utilisez simplement le préfixe 0x pour indiquer un paramètre hexadécimal. Ainsi, les nombres ci-dessous représentent tous deux le décimal 64 :

64

0x40

Où placer une macro ?

Les macros peuvent être placées en différents endroits d'un hypertexte, selon les effets à obtenir. Certaines des macros n'ont d'ailleurs de sens qu'à l'un ou l'autre seulement de ces emplacements. Nombre d'entre elles ne fonctionnent que si elles sont exécutées dans une rubrique affichée en fenêtre principale, ni dans une fenêtre secondaire, ni dans une note pop-up de définition.

Consultez la référence complète des macros dans le chapitre 15 du HAG pour voir où il est effectivement possible d'utiliser la macro qui vous intéresse.

Macro exécutées à l'ouverture du fichier d'aide

Si vous placez une macro dans la section [CONFIG]² de votre fichier de projet (*.HPJ) cette macro sera exécutée dès que l'hypertexte est lancé. Vous pouvez bien entendu mettre là plusieurs macros. Cette opération se fait dans GenHelp avec la commande Fichier | Charge projet, qui vous permet d'éditer votre HPJ. Vous constaterez alors que ce fichier contient déjà par défaut un certain nombre d'appels de macros. L'exemple ci-dessous vous montre la section [CONFIG] typique générée par GenHelp.

```
[CONFIG]
RegisterRoutine("MMSystem","MciExecute","S")
RR("MMSystem","sndPlaySound","Su")
RR("MMSystem","MciSendString","SSuu")
RR(`user.exe',`SetWindowText',`US')
CreateButton("btn_quitter", "&Quitter", "Exit()")
CB("btn_fen_index", "&Fenêtre navigation",
"JI(>glossary,RF_INDEX)")
BrowseButtons()

InsertItem(`mnu_glossaire',`fl_glossaire',`Glossaire',JI(>glossaire',`RF_GLOSSAIRE'),2)
```

Vous remarquerez d'abord que chaque macro a un nom sous deux formes, longue et abrégée. Ainsi, RegisterRoutine peut s'écrire aussi RR,

² Voir le deuxième article de cette série *Naviguer dans l'hypertexte* pour davantage d'explications sur les fichiers de projet.

et `CreateButton`, `CB`. La macro `InsertItem` a déjà été commentée dans l'article *Naviguer dans l'hypertexte*, et je vous y renvoie donc. `RegisterRoutine` prépare les appels de DLL. Nous en parlerons plus loin.

CreateButton ou CB

Cette macro crée un nouveau bouton dans la barre d'outils standard de l'aide Windows. Vous ne pouvez créer que 16 de ces boutons personnalisés en tout. Aucune des macros de boutons ne fonctionne dans une fenêtre secondaire.

La syntaxe de base : `CreateButton("identificateur du bouton","nom affiché","macro lancée")`

- L'identificateur du bouton est un nom que Windows va utiliser de façon interne. Il n'est jamais apparent pour l'utilisateur. A enserrer de guillemets.
- Le nom affiché est le texte qui prendra effectivement place sur le bouton. Il peut compter jusqu'à 29 caractères, mais pour des raisons d'esthétique des écrans, restez-en à des noms très courts. Ce nom doit lui aussi être entouré de guillemets. Vous pouvez faire de l'une des lettres de ce nom un raccourci-clavier en la faisant précéder du signe `&`.
- La macro lancée est une chaîne de caractères indiquant l'action que va provoquer un clic sur le bouton. Ce peut être une macro de `WinHelp` ou un appel à une DLL externe. Cette chaîne de caractères peut d'ailleurs contenir plusieurs macros successives, séparées de points-virgules. Là encore, entourez votre paramètre de guillemets.

BrowseButtons ou BB

Cette macro affiche dans la barre d'outils de l'hypertexte les deux boutons permettant une lecture linéaire de vos rubriques. La syntaxe est simplissime : `BrowseButtons ()`. Pas de paramètres, mais n'oubliez pas les parenthèses.

Rechercher	Rubriques d'aide	Précédent	Imprimer	Quitter
≤<	≥>	Fenêtre navigation		

Figure 2 : Barre d'outils modifiée par les macros `CB` et `BB`

Macros exécutées depuis une note de bas de page

Si une macro est incluse dans une note de bas de page, elle sera exécutée dès que l'utilisateur affiche la rubrique concernée. Pour créer une telle macro, il suffit d'insérer dans la rubrique voulue une note de bas de page appelée par un point d'exclamation (!). Dans le volet des notes de bas de page, les choses se présentent ainsi :

! Annotate()

Cet exemple provoquerait l'ouverture de la fenêtre d'annotation à chaque affichage de la rubrique. Notez l'unique espace entre le point d'interrogation et le texte de la macro lui-même.

Dans GenHelp, ce processus est automatisé par Insertion | Note automacro. Ce processus ne fonctionne hélas pas pour les définitions, seulement pour les rubriques.

Ainsi placées, j'utilise effectivement les macros suivantes dans mes hypertextes :

- `JumpId('MONPROJET.HLP>glossary', 'RF_Index')` correspond à la commande Outils | Ouverture index auto de GenHelp et provoque l'affichage de la fenêtre d'index dès l'ouverture de l'hypertexte.
- `Print()` me permet de lancer l'impression d'une rubrique dès qu'elle est affichée. J'utilise quasi-systématiquement cette macro pour les rubriques contenant une grille de mots croisés. Je donne alors bien évidemment le forme *Imprimer une grille de mots-croisés* à mon entrée dans l'index.
- `SndPlaySound("carillon.wav",1)` lance l'exécution d'un fichier son. J'y reviendrai en détail un peu plus bas. Elle me permet de faire entendre la prononciation d'un nom de rubrique étranger ou inhabituellement difficile, dès l'affichage.

Macro exécutées depuis un appel de rubrique ou de définition

C'est ce que dans GenHelp j'appelle un lien macro. Je les utilise de deux façon différentes : un lien hypertexte placé sur un bouton pour déclencher la macro, pour appeler la calculette de Windows, par exemple. Ou un lien appelant une macro depuis un mot de l'hypertexte.

Pour lancer une macro depuis un bouton, ou n'importe quelle image d'ailleurs, il vous suffit dans GenHelp de sélectionner le code d'insertion de l'image, en prenant bien soin d'inclure les deux accolades, avant d'utiliser la commande adéquate : Insertion | Lien macro, Inser

tion | Références Wav, Insertion | Références programme ou Insertion | Animation MCI. Voici un exemple du code obtenu.

```
{bmc Wav.bmp}!sndPlaySound("carillon.wav",1)
```

Créer un bouton dans un titre

Un effet très intéressant consiste à créer un bouton dans un titre de rubrique. Il pourra appeler une note-planche, un logiciel extérieur, un fichier sonore, une animation AVI... Le but est de rendre cette commande disponible indépendamment du défilement de la fenêtre principale. Le résultat obtenu sera celui de la copie d'écran ci-dessous.



Figure 3 : Un bouton dans un titre

Pour ce faire, vous devrez mettre en place votre bouton à la fin du titre de rubrique, juste avant le caractère de fin de paragraphe. Utilisez la commande Insertion | Bouton standard.

Liez le bouton créé à la définition de la note-planche, ou à la macro lançant un programme externe. Le code doit alors se présenter comme ci-dessous.

```

$#+K.Le-récit↵
{bmc-LOGO.BMP}CLEOPATRE-PARLE-AU-MOINS-SIX-LANGUES.
RF_CLEOPATRE_PARLE_AU_MOINS_{BML:
C:\MSOFFICE\MODELES\GENHELP\IMAGES\BT_GLOB2.WMF}DF_PL_CARTEBMP||

```

Si vous insérez le bouton ailleurs qu'à l'extrême fin de la rubrique de titre, juste avant la marque de fin de paragraphe, le lien avec une rubrique ou une note ne se fera pas correctement.

Des boutons sonores

Ne pas confondre avec les boutons destinés à l'écoute d'un fichier WAV. Il s'agit de faire entendre un bruitage en plus d'une action quelconque. De tiroir-caisse pour une note de prix, de moteur photo pour l'affichage d'une note-planche, à votre guise.

Les boutons sonores ne sont qu'un cas particulier de l'emploi de liens hypertextes créés par macro, permettant d'enchaîner plusieurs actions lors du clic de l'utilisateur. En effet, le lien hypertexte standard permet uniquement de se rendre à une autre rubrique ou d'afficher une définition dans une fenêtre pop-up. Deux macros se prêtent à ce type de liens, il s'agit de `PopupId()` pour les définitions et `JumpId()` pour les rubriques.

L'opération se fait en deux temps :

Inclure par référence l'image du bouton, éventuellement avec Insertion | Bouton standard.

Affecter au bouton en question un lien macro double suivant le modèle ci-dessous :

```
!PopupId("nom hypertexte","identificateur
définition");sndPlaySound("nom fichier son",1)
```

Remarquez le point-virgule pour enchaîner deux macros. Cette opération est faite automatiquement si vous validez l'option *lien par macro* à la création d'un appel de définition normal. Vous pouvez évidemment utiliser `JumpId()` pour sauter à une rubrique au lieu de faire apparaître une définition.

GEN_HELP contient un unique bouton sonore prêt à employer, avec ses macros déjà codées sous SHED. Le fichier en est BT_SUITE.SHG, associé au fichier son SUITE.WAV.

APPELER UN LOGICIEL DEPUIS L'AIDE WINDOWS

Une des macros est spécialement efficace pour ouvrir vos applications vers l'extérieur, c'est `ExecProgram` (ou EP en version abrégée). Comme son nom l'indique, elle sert à lancer un logiciel.

La syntaxe de base en est simple : `ExecProgram("commandes", aspect)`

- Le paramètre "commandes", à inclure entre guillemets va comprendre le nom de l'application à lancer, avec éventuellement son chemin et ses paramètres. Par exemple "C : \WINDOWS \WRITE.EXE EXO01.WRI".

Si les paramètres avec lesquels vous lancez votre logiciel doivent eux-mêmes être définis comme chaîne de caractères, vous arriverez à la syntaxe tordue ci-dessous :

`ExecProgram('commandes "chaîne de caractères"', 0)`

- Le paramètre numérique aspect va définir l'état de la fenêtre de l'application à lancer. Avec une valeur 0, fenêtre normale ; une valeur 1, l'application est lancée en icône et avec une valeur 2, en plein écran.

Attention, tous les logiciels n'acceptent pas les indications données par ce deuxième paramètre, nombreux sont ceux qui l'ignorent superbement. Faites ce test simple : lancez l'application à essayer en maintenant la touche majuscule enfoncée pendant que vous double-cliquez sur l'icône. Si le logiciel démarre en icône, tout devrait bien se passer aussi avec `ExecProgram`.

En pratique, le problème va être la localisation de l'application à lancer. Vous devez être certain qu'elle est bien présente sur la machine de l'utilisateur, et à l'endroit requis de surcroît. Winhelp, dans l'ordre, la cherchera dans les répertoires suivants

- Le répertoire courant
- Le répertoire de l'hypertexte d'origine
- Le répertoire MS-DOS (Windows 3.x) ou `WINDOWS\COMMAND` (Windows 95)
- Le répertoire `WINDOWS`
- Le répertoire `WINDOWS\SYSTEM`
- Les répertoires listés dans la variable `PATH`
- Les répertoires spécifiés dans le fichier `WINHELP.INI`

Sachez aussi que la macro `ExecProgram` ne peut en aucun cas procéder à un changement de répertoire courant. Je vous conseille donc de vous limiter soit à des logiciels faisant partie de Windows, comme `WRITE` ou la calculette, ou des logiciels que vous installez automatiquement en même temps que votre hypertexte.

CRÉER DES LIENS ENTRE PLUSIEURS HYPERTEXTES

Pour dépasser le niveau du simple hypertexte "individuel", il convient de pouvoir gérer des ensembles de documents multimédia, qui

feront des références les uns aux autres, avec des liaisons non plus seulement internes, mais interfichiers.

Lien interfichier

Le lien interfichier est identique au lien hypertexte normal, à ceci près que la rubrique de destination est prise dans un autre hypertexte.

Ces liens sont gérés automatiquement dans la dernière version de GenHelp. Une première boîte de dialogue vous permet de choisir le fichier INI correspondant à l'hypertexte contenant la rubrique de votre choix. Attention à ne pas choisir n'importe quel fichier INI !

Le choix de la rubrique s'effectue ensuite comme normalement.

Ces liens interfichier se traduisent par un code semblable aux liens ordinaires, à ceci près que le nom du fichier d'aide externe est indiqué après une esperluette (@) comme dans l'exemple ci-dessous.

La France est une démocratieDF_DEMOCRATIE@DICOPOLI.HLP

Cliquer sur le mot démocratie appelle ici l'affichage d'une définition sous forme de fenêtre pop-up. Le texte de cette définition ne sera cependant pas pris dans l'hypertexte courant, mais dans un autre hypertexte, ici DICOPOLI.HLP.

Le principal problème sera de retrouver l'emplacement de l'hypertexte cible sur la machine de l'utilisateur final. Il ne faut en aucun cas utiliser de chemin absolu, qui conduirait à des problèmes insolubles, mais vérifier que l'hypertexte en question se trouve à l'un des emplacements suivants :

- Le répertoire courant
- Le répertoire de l'hypertexte d'origine
- Le répertoire MS-DOS (Windows 3.x) ou WINDOWS\COMMAND (Windows 95)
- Le répertoire WINDOWS
- Le répertoire WINDOWS\SYSTEM
- Le répertoire WINDOWS\HELP (Windows 95)
- Les répertoires listés dans la variable PATH
- Les répertoires spécifiés dans le fichier WINHELP.INI

Les dictionnaires

Le dictionnaire est un hypertexte qui, en dehors de sa première rubrique de titre, normalement jamais affichée, ne contient que des définitions. Le but est de constituer, dans un champ pédagogique, un corpus de vocabulaire technique courant, qu'il ne serait plus besoin de réintroduire dans chacun de vos hypertextes. Tous les appels à définitions se faisant uniquement par des liens interfichiers, au moyen de la commande Insertion | Lien définition Interfichier.

Vous pourriez d'ailleurs constituer de la même façon un atlas, un recueil de graphiques ou de schémas techniques en créant un hypertexte rempli seulement de notes-planches ³.

ENRICHIR L'AIDE : API ET DLL

Les API (Application Programming Interface) sont des fonctions que Windows met à la disposition des programmeurs pour accéder aux différentes possibilités de cet interface graphique. Elles sont contenues habituellement dans des portions de code utilisables par plusieurs logiciels en même temps, les DLL ou Dynamic Links Library.

De nombreuses lacunes de Winhelp peuvent ainsi être comblées par un appel direct à Windows. Tous les bons manuels des langages de programmation donnent des explications détaillées sur la façon d'utiliser ces routines. GEN_HELP en fait un usage fréquent. Vous trouverez un certain nombre d'hypertextes en circulation, nommés souvent WINAPI.HLP et décrivant par le menu l'ensemble de ces API. Malheureusement, la syntaxe est toujours, ou presque, donnée pour le langage C et nécessite quelques modifications pour une utilisation dans les aides HLP.

Les principes que nous allons voir ici sont transposables à d'autres langages et logiciels. J'utilise les mêmes appels aux API pour créer l'aide contextuelle dans les boîtes de dialogue de GenHelp, en WordBasic cette fois ou donner des possibilités multimédia à la version simple de Toolbook.

³ Voir l'article précédent *Les Images*.

Register Routine

Tout appel aux API passe par une déclaration préalable, incluse dans le fichier de projet [NOMDEPROJET].HPJ, à la section [CONFIG]. La macro RegisterRoutine ou RR permet d'effectuer cette déclaration.

La ligne ci-dessous permet par exemple d'utiliser la fonction multimédia très simple de Windows MciExecute, localisée dans le fichier MMSystem.DLL et d'inclure dans nos hypertextes vidéo ou animation.

```
RegisterRoutine("MMSystem","MciExecute","S")
```

Syntaxe

RegisterRoutine("Nom de DLL", "Nom de fonction", "Format des paramètres")

Le nom de DLL est une chaîne précisant le nom de la DLL appelée. S'il s'agit effectivement d'un fichier *.DLL, l'extension peut être omise. Elle devra être spécifiée dans le cas d'un EXE. A priori, il faut s'interdire tout nom de chemin. Se référer à la table des répertoires de recherche pour savoir où placer ou trouver ses fichiers. En pratique, j'utilise essentiellement dans mes hypertextes MMSYSTEM.DLL, pour disposer de capacités multimédia.

Le Nom de fonction est le nom exact de l'API que vous comptez utiliser comme macro.

Format des paramètres est une chaîne de caractères spécifiant les paramètres à passer à la fonction appelée. Les types de paramètres valides sont les suivants :

Caractère à inclure	Type de données	Equivalent Windows
u	Entier court non signé	UINT, WORD, WPARAM
U	Entier long non signé	DWORD
i	Entier court signé	BOOL (type C int ou short)
I	Entier long signé	LONG, LPARAM, LRESULT
s	Pointeur Near sur une chaîne terminée par un caractère Null (type C)	PSTR, NPSTR
S	Pointeur Far sur une chaîne terminée par un caractère Null (type C)	LPSTR, LPCSTR

v	Void (absence de tout type, utilisé pour les retours de valeurs uniquement)	Equivalent du Type Void en C
---	---	------------------------------

Remarquez que les formats de paramètres font la différence entre minuscules et majuscules. Ils sont simplement aboutés pour composer la chaîne de formats de paramètres. Ainsi dans l'exemple suivant

```
RegisterRoutine('user.exe', 'SetWindowText', 'US')
```

nous passons un entier long et une chaîne de caractères en guise de paramètres.

Variables internes

Winhelp met à votre disposition un certain nombre de variables internes que vous pouvez utiliser comme paramètres à passer aux API déclarées avec RegisterRoutine.

Nom de Variable	Format de paramètre	Description
hwndApp	U	Handler de la fenêtre principale de votre hypertexte
hwndContext	U	Handler de la fenêtre courante de votre hypertexte
qchPath	S	Chemin complet de votre hypertexte
qError	S	Pointeur vers une structure d'erreur
lTopicNo	U	Numéro du "topic" (rubrique ou définition) courant
hfs	U	Handle (lien) vers le système de fichier de votre hypertexte
coForeground	U	Couleur d'avant plan dans la fenêtre active de votre hypertexte
coBackground	U	Couleur d'arrière-plan dans la fenêtre active de votre hypertexte

Vous pouvez facilement "planter" votre système par des appels inconsidérés aux DLL de Windows. Commencez par lire les chapitres 14 et 15 de HAG.

Quelques exemples d'utilisations des API

Changer le titre de la fenêtre principale

Un changement de titre de fenêtre ne peut être obtenu que par un appel à une fonction Windows contenue dans le fichier USER.EXE.

• Ouvrir le fichier de projet (Fichier Charge projet) Déclarer la fonction suivante dans la section [CONFIG] de votre fichier de projet :

```
RegisterRoutine(`user.exe', `SetWindowText', `US')
```

- Refermer le fichier de projet.
- Dans le corps de votre hypertexte, le code à insérer dans un lien macro est :

```
SetWindowText(hwndApp, `texte à afficher dans la barre')
```

• Compiler votre travail. Dans GenHelp, ne pas oublier de cocher l'option "ne pas modifier le fichier de projet" dans la boîte de dialogue de compilation. Sans quoi, la re-crédation automatique du fichier HPJ effacerait vos modifications manuelles.

Changer le titre d'une fenêtre secondaire

Pour changer le titre d'une fenêtre secondaire, vous devez procéder comme ci-dessus, mais en utilisant cette fois deux lignes Register Routine (RR) dans votre fichier de projet HPJ.

```
RR(`user.exe', `FindWindow', `U=SS')
```

```
RR(`user.exe', `SetWindowText', `US')
```

Insérer ensuite comme note automacro :

```
SetWindowText(FindWindow(`MS_WINTOPIC_SECONDARY', `ancien titre'), `nouveau titre')
```

Afficher une boîte de dialogue simple

Vous pouvez utiliser une boîte de dialogue Windows standard dans votre hypertexte, pour afficher un message exigeant une réponse de l'utilisateur. Une limitation importante de ce procédé est qu'il vous sera impossible, sauf à écrire une DLL "maison" d'exploiter des boîtes à plusieurs boutons, car WinHelp n'est pas capable de traiter les codes de retour. Vous serez donc limité à des boîtes à bouton unique, demandant

par exemple à l'utilisateur de vérifier que son imprimante est prête avant de cliquer sur [OK].

Incluez dans votre fichier de projet la ligne RegisterRoutine suivante :

```
RegisterRoutine(`user.exe',`MessageBox',`USSu')
```

L'appel de la macro proprement dite se fera selon la syntaxe suivante :

```
MessageBox(Hwnd,`texte du message',`titre de la boîte',icône)
```

La valeur de "hwnd" sera la variable interne hwndContext si vous lancez la macro depuis une rubrique affichée ou hwndApp si l'appel de macro est placé dans la section [CONFIG] du fichier de projet.

Pour les deux chaînes de caractères, n'oubliez pas l'antislash si vous voulez inclure une apostrophe. Pour forcer un saut de ligne dans le texte du message, incluez la séquence `\'0A` à l'endroit voulu.

Le paramètre numérique icône peut prendre les valeurs suivantes :

Valeurs hexadécimales	Nom Windows	Description
0x000		Pas d'illustration
0x0040	MB_ICONASTERISK, MB_ICONINFORMATION	Un i minuscule dans un cercle
0x0030	MB_ICONEXCLAMATION	Point d'exclamation
0x0010	MB_ICONHAND, MB_ICONSTOP	Panneau STOP
0x0020	MB_ICONQUESTION	Point d'interrogation

L'exemple ci-dessous :

```
MessageBox(hwndContext,`Avez vous mis l'imprimante sous  
tension',`Mots fléchés',0x0010)
```

affiche une boîte titrée *Mots fléchés*, avec le message *Avez-vous mis l'imprimante sous tension*, un panneau STOP et le bouton OK.

PLUS LOIN

Pour aller plus loin dans la découverte des possibilités de Winhelp, je vous renvoie avant tout aux diverses sources disponibles sur Internet,

au regard de la pauvreté des documents papier effectivement disponibles en France. Voici donc les sites que j'ai retenus dans mon *bookmark*.

<http://www.ac-strasbourg.fr>

Tout d'abord, l'adresse du serveur Web de l'académie de Strasbourg, qui devrait être accessible au moment où vous lirez ces lignes. Je compte bien pouvoir y donner des conseils de création multimédia concernant aide Windows, livrets Toolbook et pages HTML, et vous proposer des sites à visiter et des outils à télécharger.

<ftp://ftp.win.net/winnet/infosoft/winhelp/>

<ftp://ftp.onyxgfx.com/pub/winhelp/> Onyx Winhelp

Deux sites ftp où vous pourrez trouver nombre d'utilitaires et d'exemples de très beaux hypertextes. J'apprécie tout spécialement le premier qui, s'il n'est pas très abondant, contient des fichiers récents et d'un grand intérêt.

<http://www.sky.net/~parnote/>

<http://www.sky.net/~parnote/helpful.htm>

Un site majeur pour tous les "helpeurs", les pages Web de Paul Arnote qui méritent un détour régulier. La première adresse propose surtout des "tuyaux" et des conseils de réalisation. La seconde, un panorama des autres sites concernant l'aide Windows, très régulièrement mis à jour et commenté.

<http://www.primenet.com/~wai/text/universe.html>

The Windows Help Universe

<http://www.r2m.com/windev/winhelp.html>

WinHelp Developer Internet Resouces

<http://www.hyperact.com/winhelp/>

WinHelp WWW Index

<news:comp.os.ms-windows.programmer.winhelp>

Un newsgroup où vous pourrez poser toutes vos questions et faire un tour d'horizon des problèmes actuels et de leurs solutions.

Pierre KESSLER

Service de formation informatique

CRDP - 23 rue du Mal. Juin

67007 Strasbourg Cedex