

HYPERTEXTES MULTIMEDIA - 4 (2ème partie) Ouvertures

Pierre KESSLER

INCLURE DES FICHIERS MULTIMEDIA

La principale application de RegisterRoutine reste néanmoins la possibilité d'ajouter des capacités multimédia à vos hypertextes. Quatre types de fichiers vont ici être pris en compte : WAV, MID, AVI et FLI/FLC. Il n'y a toutefois aucune limitation réelle que celle des pilotes effectivement installés sur la machine.

Les fichiers Wav

Les fichiers *.WAV sont, comme chacun sait, des fichiers son de Windows. Vous pouvez, au moyen de ces fichiers, sonoriser vos hypertextes. Pour autant que l'utilisateur final dispose d'une carte son. Si tel n'est pas le cas, il ne se produira strictement rien à l'exécution, et en tout cas, aucun "plantage".

Dans GenHelp, la mise en place est très simple : la commande Insertion | Références Wav affiche la boîte ci-dessous.

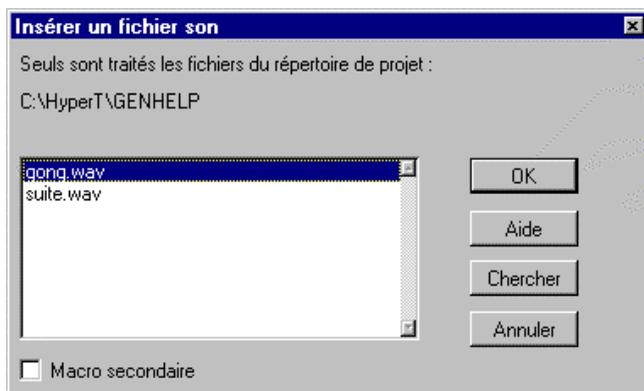


Figure 1 : Insertion d'un fichier Wav dans GenHelp

Comme tous les fichiers auxiliaires, les WAV doivent se trouver dans le répertoire de projet. Contrairement aux images, ils ne seront pas inclus dans l'hypertexte, et devront être joints au fichier HLP final.

Le bouton [Chercher] vous permet cependant de choisir un fichier extérieur au répertoire de projet, mais la macro se charge alors de l'y copier. Attention, il faut absolument avoir sélectionné les mots ou l'image servant de bouton avant de lancer cette commande.

L'option macro secondaire permet d'enchaîner plusieurs macros l'une derrière l'autre. Elle ne doit être utilisée qu'après une première utilisation de l'une des commandes à macro : appel de programme, de macro, de fichier son ou d'animation MCI.

Attention, si vous utilisez le bouton [Chercher], il faut d'abord valider l'option, car la boîte de dialogue ne sera plus réaffichée ultérieurement.

Ces actions automatisées correspondent à la mise en place de la macro :

```
RegisterRoutine("MMSystem","sndPlaySound","Su")
```

dans votre fichier de projet (HPJ).

Dans le corps de l'hypertexte, l'appel lui-même se fera par :

```
SndPlaySound("Nom de fichier wav",statut)
```

Le paramètre numérique *statut* aura l'une des valeurs décrites dans le tableau suivant, ou l'addition de plusieurs de ces valeurs.

Valeur	Nom Windows	Description
0x0000	SND_SYNC	Joue le fichier son "synchrone" et ne rend pas la main à l'utilisateur avant la fin de l'exécution.
0x0001	SND_ASYNC	Joue le fichier son "asynchrone". L'utilisateur reprend la main dès le début de l'exécution.
0x0002	SND_NODEFAULT	Ne joue pas le son par défaut indiqué dans WIN.INI. Si le fichier WAV n'est pas trouvé, l'hypertexte reste silencieux.

0x0008	SND_LOOP	Boucle en continu jusqu'à la prochaine commande sndPlaySound, avec un nom de fichier vide ('').A combiner obligatoirement avec SND_ASYNC.
0x0010	SND_NOSTOP	N'arrête pas le fichier son courant pour en jouer un autre.

Pour éviter toute erreur dans la combinaison des valeurs hexadécimales, servez-vous de la calculatrice Windows. Passez en affichage scientifique, et cliquez sur le bouton [HEX] avant d'additionner tout simplement vos hexadécimaux.

Les fichiers Midi

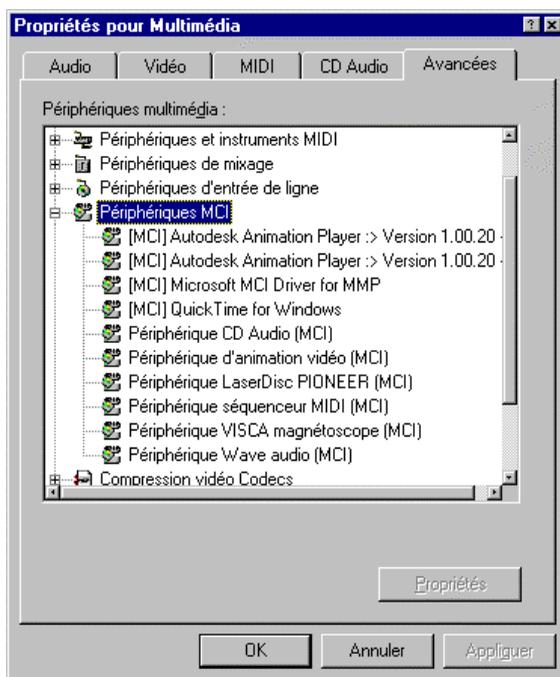


Figure 2 : Pilotes MCI installés sur une machine en Windows 95

Pour l'ensemble des fichiers multimédia, Windows utilise le MCI. (Multimédia Control Interface) Toutes les applications tournant sous Windows peuvent y faire appel. MCI est capable de gérer toutes sortes de périphériques, pour peu que les gestionnaires correspondants soient installés. Ceci se vérifie dans le panneau de configuration, icône multimédia.

MCI présente un nombre important de fonctions, des plus simples aux plus complexes. Nous avons vu plus haut `SndPlaySound`. Deux autres de ces fonctions méritent notre attention, `MciExecute` et `MciSendString`.

`MciExecute` est la plus simple d'emploi. Sa déclaration dans le fichier de projet se fait de la façon suivante :

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

ligne de code que vous avez rencontré plus d'une fois déjà dans cet article. Elle accepte une chaîne de caractères, en fait la commande à exécuter, comme unique paramètre.

Le lancement de la macro, dans le corps de votre hypertexte, suivra la syntaxe suivante :

```
MciExecute("Play NomAvi.Ext")
```

Pour nous, la commande passée à `MciExecute` sera toujours l'instruction `PLAY` suivie du nom d'un quelconque fichier MCI. Que ce soit un `.WAV`, un `.MID` ou `.AVI`. Cette fonction cependant nous laisse très peu de contrôle sur les opérations multimédia en cours, et je l'ai à peu près totalement abandonnée au profit de `MciSendString`.

La fonction MciSendString

La déclaration dans votre fichier de projet se fera sous la forme :

```
RegisterRoutine(`mmsystem.dll`,`mciSendString`,`SSuu`)
```

Et la lancement dans le corps de l'hypertexte suivra la syntaxe :

```
mciSendString(`chaîne de commande`,``,0,0)
```

Parmi les commandes destinées à `MciSendString`, voici quelques unes des plus usitées :

- `open` ouvre un fichier (ou un périphérique comme le CD-ROM-Audio).
- `play` démarre l'exécution du fichier (ou met en route le périphérique).
- `pause` arrête provisoirement l'exécution à la position courante.

- stop arrête l'exécution et revient à la position de départ.
- alias permet d'utiliser un nom quelconque pour désigner un fichier ou périphérique dans les commandes suivantes.
- wait attend que le fichier ou le périphérique ait terminé son exécution avant de rendre le contrôle à l'application appelante.
- close ferme le fichier ou le périphérique.

Pour régler l'exécution d'un fichier multimédia, les appels à cette macro seront toujours enchaînés, comme le montre l'exemple ci-dessous, qui joue le fichier midi canyon.mid, normalement présent dans votre répertoire \WINDOWS.

```
mciSendString( open canyon.mid alias
canyon',',0,0) ;mciSendString( play canyon
wait',',0,0) ;mciSendString( close canyon',',0,0)
```

Jetons maintenant un coup d'oeil à l'exemple élaboré ci-dessous, établi par l'expert en fichiers HLP, Paul Arnote (voir son URL en fin d'article). Il nous propose une interface graphique de type magnétophone à inclure dans un hypertexte pour jouer un fichier MIDI. Utilisez n'importe quelle image BMP à votre convenance.



Figure 3 : Interface pour fichiers MIDI proposé par Paul Arnote

Cliquer sur la flèche (verte) correspond à la fonction PLAY et lance l'audition du fichier. Les deux barres (bleues) vous donnent la PAUSE et le carré (rouge) correspond à une touche STOP.

Le fonctionnement est assuré par des macros liées à chacune des zones de la barre dans l'éditeur d'hypergraphiques SHED, comme indiqué dans mon précédent article.

- Charger le bitmap de votre barre de commandes dans Shed.
- Pour la fonction PLAY, sélectionner le premier bouton et le lier à la suite de macros suivante :

```
mciSendString( open canyon.mid alias
fichierSon',',0,0) ;mciSendString( play fichierSon',',0,0)
```

- Pour la fonction PAUSE, sélectionner le deuxième bouton et le lier à la suite de macros suivante :

```
IfThenElse(IsMark(`pause'), `DeleteMark(`pause') ;mciSendString(`play
fichierSon', `', 0, 0)', `SaveMark(`pause') ;mciSendString(`pause
fichierSon', `',0,0)')
```

- Pour la fonction STOP, sélectionner le dernier bouton et le lier à la suite de macros suivante :

```
mciSendString(`stop fichierSon', `',0,0) ;mciSendString(`close
fichierSon', `',0,0)
```

Vous recevrez pour chacune de ces chaînes de macros un message d'erreur (Invalid macro), car il ne s'agit pas de macros standard de WinHelp mais d'appels à des DLL. Pas de panique, c'est normal.

- Il ne vous reste plus qu'à sauvegarder votre barre de commandes au format SHG dans votre répertoire de projet, puis à l'inclure par référence (Insertion | Références images) dans votre hypertexte.

Lancer un fichier vidéo ou d'animation

GenHelp vous propose de réaliser cela très simplement par Insertion | Animation MCI. Cette commande permet d'insérer dans votre hypertexte tout type de fichier multimédia supporté par le diffuseur de médias MPLAYER.EXE. Il peut s'agir de fichiers WAV, MID, AVI, FLI ou FLC.

GenHelp vous permet en outre de choisir parmi les options ci-dessous dans quel type de fenêtre sera affichée votre fichier AVI ou FLC.



Figure 4 : choix de l'aspect d'une fenêtre d'animation dans Genhelp

Vous devez cependant vous assurer que les drivers MCI correspondants sont bien installés sur la machine de l'utilisateur final. En particulier, le driver pour les animations FLI et FLC n'est pas livré par défaut avec Windows, et doit être chargé depuis la disquette AAPLAY, runtime freeware de Autodesk Animator. Il est prudent de la joindre à votre création si vous utilisez de tels fichiers.

Les fichiers multimédia doivent comme toujours se trouver dans le répertoire de projet. Vous pouvez cependant choisir un fichier extérieur au répertoire de projet, mais la macro se charge alors de l'y copier.

Les fichiers FLI, FLC et AVI sont diffusés par l'enchaînement de macro-commandes ci-dessous :

```
mciSendString('open clouds.avi alias bldg style popup', '\', 0,
0) ;mciSendString('put bldg window at 160 160 0 0', '\', 0,
0) ;mciSendString('play bldg from 0 wait', '\', 0, 0) ;mciSendString('close
bldg', '\', 0, 0)
```

Les deux nombres 160,160 indiquent la position du coin supérieur droit de la fenêtre vidéo. Vous pouvez éventuellement les modifier à la main pour obtenir un emplacement différent. Mais attention aux changements de résolution d'écran.

LES FICHIERS DE MAPPING

L'insertion d'une section MAP dans le fichier de projet permet la mise en place effective des mécanismes d'aide contextuelle depuis votre hypertexte. Elle sert à affecter un "context number" à chaque référence de votre hypertexte. Cet identificateur numérique de contexte sera ensuite repris dans les propriétés des composants de votre application Windows, ou utilisé dans un appel explicite.

Vous pourrez construire une aide contextuelle, comme celle de GEN_HELP, pour vos programmes. Mais le mapping offre également la possibilité d'appeler votre hypertexte, non pas en bloc, à sa page d'introduction, mais à une rubrique précise, depuis un autre hypertexte, une simulation, un QCM... Il sera ainsi possible d'intégrer votre création à un véritable réseau d'hypertextes s'appelant les uns les autres.

Voici un extrait de fichier de projet contenant des indications de mapping :

```
[CONFIG]
```

```

RegisterRoutine("MMSystem","sndPlaySound","Su")
RegisterRoutine("MMSystem","MciExecute","S")
CBB("btn_contents","JI(^,`RF_Index')")
CB("btn_quitter", "&Quitter", "Exit()")
BrowseButtons()
CB("btn_fen2", "&Fenêtre d'Index", "JI(^POP_POP.HLP>glossary',
`RF_Index')")
[OPTIONS]
REPORT=ON
ROOT=C:\RENNES\POP_POP
BMROOT=C:\RENNES\POP_POP
COMPRESS=OFF
OLDKEYPHRASE=NO
TITLE=Les bateaux pop-pop
[FILES]
C:\RENNES\POP_POP\POP_POP.RTF
[WINDOWS]
main = "Les bateaux pop-pop", (0,0,650,977 ), (255,255,223),
(192,192,192)
glossary = "Index", (650,0,373,977 ), (255,255,223), (192,192,192)
[BUILDTAGS]
[ALIAS]
[MAP]
RF_INTRODUCTION 1
RF_CONTINUER 2
RF_ORIGINE 3
RF_COMBUSTIBLES 4
RF_REMPLISSAGE 5
RF_PRINCIPES 15
DF_POP_POP 16
DF_DROGUERIE_MARINE 17
DF_ST_MALO 18
DF_SALUT_JEAN_PIERRE 19

```

DF_PARI_TENU 20

[BITMAPS]

Les valeurs numériques sont indifférentes. Elles doivent simplement être uniques, et il est préférable en pratique qu'elles se suivent. Il est bien entendu nécessaire de compiler votre hypertexte pour que le mapping soit effectif. Et vous aurez ensuite à imprimer la liste des valeurs de mapping pour en tirer quelque chose...

Dans GenHelp, le fonctionnement est automatique. Une boîte de dialogue vous avertit de la fin de l'opération. Le mapping ne doit intervenir qu'à la fin de votre travail, les autres opérations achevées.

Une section [MAP] est créée dans votre fichier de projet, ainsi qu'un fichier [NomDeProjet].MAP reprenant les références de vos rubriques.

Même si vous n'avez pas dès maintenant l'utilisation du mapping, je vous engage fortement à le réaliser sur votre hypertexte, et à distribuer le fichier [NomDeProjet].MAP aux utilisateurs en plus du fichier [NomDeProjet].HLP.

Exemple d'utilisation dans le Test Poker

Le Test-Poker est un logiciel d'administration de et de création de QCM, à tendance nettement ludique. Il est conçu plus comme un outil anti-zapping que dans un but réel d'évaluation des connaissances. J'y utilise pleinement le mapping, puisque le fichier MAP créé par GenHelp est directement relu dans le module de création des questionnaires. Ainsi, l'enseignant peut assigner la référence d'une rubrique de son hypertexte à chacune des questions posées, et afficher à volonté une aide contextuelle. L'élève a ainsi la possibilité d'afficher le cours en version hypertexte avant de répondre à la question posée.

Voyez ci-dessous comment les choses se présentent dans le module *Maintenance des questionnaires* du Test-Poker. Si vous avez envie d'utiliser ce logiciel, il est également disponible gratuitement à la Bourse aux disquettes.

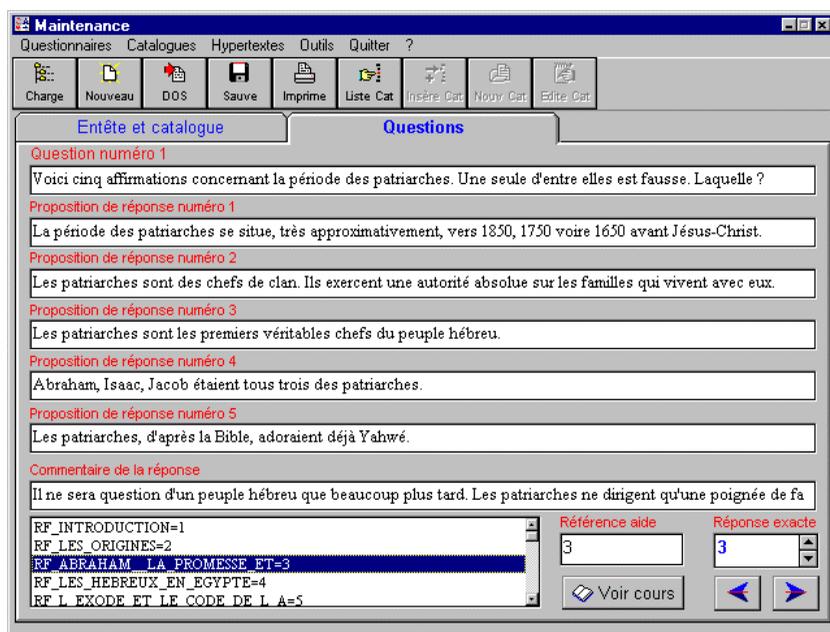


Figure 5 : Utilisation du mapping dans Test-Poker

APPELER L'AIDE WINDOWS DEPUIS UN LOGICIEL

Une question doit être toujours posée : vaut-il mieux appeler l'hypertexte depuis un logiciel, ou un logiciel depuis l'hypertexte. Deux facteurs vont principalement entrer en jeu : êtes-vous l'auteur de ce logiciel, et quelle en est la taille ? Pour tout logiciel réellement important par sa taille, dépassant le cadre des petits accessoires livrés avec Windows, ou de petits produits analogues, il me semble infiniment préférable d'aller de l'exécutable vers l'hypertexte, et non l'inverse.

Rappelez-vous au passage que vous n'êtes pas limités à la seule aide contextuelle sur la façon d'utiliser votre logiciel, mais que vous pouvez y adjoindre des hypertextes de cours sur tous les sujets. Le Test-Poker décrit plus haut est un exemple simple d'un tel produit. Je commence à travailler actuellement, dans le même ordre d'idée, sur un générateur de jeux de simulation simples, qui permettrait à partir d'un hypertexte de créer pour les élèves des aventures textuelles, un peu sur le modèle des livres dont vous êtes le héros.

Bien entendu, si vous n'êtes pas l'auteur du logiciel, il ne vous reste guère que la possibilité de travailler dans l'autre sens, c'est-à-dire d'appeler le logiciel depuis l'hypertexte, comme je le fais par exemple pour le petit logiciel de puzzle Winjig, livré dans le pack GenHelp. En souhaitant dans ce cas que votre application externe accepte des paramètres en ligne de commande...

DLL 16 bits et DLL 32 bits

L'appel à une aide contextuelle repose encore une fois sur les API. C'est la fonction WINHELP, située dans le fichier USER.EXE que nous allons solliciter. Certains langages de programmation comme Visual Basic ou Delphi intègrent directement cet appel, sans qu'il soit besoin de déclarer quoi que ce soit, en mettant simplement à jour une propriété HelpContext ou équivalente des différents composants. Dans d'autres produits, comme Winword ou Toolbook, il vous faudra faire une déclaration analogue à RegisterRoutine et un appel explicite à l'API.

Ceci suppose bien entendu que vous ayez compilé votre hypertexte avec une section mapping, comme je l'indique au paragraphe correspondant.

Les choses se compliquent un peu avec l'apparition de Windows 95. Ce dernier environnement dispose en effet de deux jeux complets de DLL et d'API, l'un pour les applications 16 bits et l'autre pour les applications 32 bits. Si vous essayez d'appeler, par exemple, une DLL 16 bits depuis la version 32 bits de Word, vous obtiendrez un superbe message d'erreur, vous indiquant que les logiciels 32 bits ne doivent appeler que des DLL 32 bits. Ce qui, au passage, explique aussi pourquoi nombre de DLL écrites spécialement pour apporter des fonctionnalités complémentaires à Winhelp ne fonctionnent plus sous Windows 95.

Sous 95, la plupart des fichiers contenant les API ont vu leur nom évoluer par l'adjonction d'un suffixe 32. USER.EXE devenant ainsi USER32.EXE. De plus, les noms des API elles-mêmes ont parfois changé, ainsi que les paramètres à leur passer. L'API de l'aide est nommée WINHELP en 3.11 et WINHELPA en 95... Comparez les différents exemples ci-dessous pour vous en faire une petite idée.

Sous Word 7

```
REM Macro AideContextuelle
REM Gen_Help version 7.0 pour Word 7
REM Dernière modification le 25/1/96
```

```

Declare Function WinHelpA Lib "USER32"(hWnd As Integer,
lpHelpFile As String, wCmd As Integer, dwData As Integer) As
Integer
Declare Function GetActiveWindow Lib "USER32"() As Integer
Sub MAIN
End Sub
Function GetHelp(HelpId)
    hWnd = GetActiveWindow
Select Case HelpId
Case 77
    fichierAide$ = inilib.windir$ + "\G_HELP7.HLP>TUTORIEL"
Case Else
    fichierAide$ = inilib.windir$ + "\G_HELP7.HLP>MAIN"
End Select
GetHelp = WinHelpA(hWnd, fichierAide$, 1, HelpId)
End Function

```

Sous Word 6

```

REM Macro AideContextuelle
REM Gen_Help version 6.6 pour Word 6
REM Dernière modification le 25/1/96
Declare Function WinHelp Lib "USER.EXE"(hWnd As Integer,
lpHelpFile As String, wCmd As Integer, dwData As Long) As Integer
Declare Function GetActiveWindow Lib "USER.EXE"() As Integer
Sub Main
End Sub
Function GetHelp(HelpId)
    hWnd = GetActiveWindow
    fichierAide$ = inilib.windir$ + "\G_HELP6.HLP>MAIN"
    GetHelp = WinHelp(hWnd, fichierAide$, 1, HelpId)
End Function

```

Et un exemple en DELPHI

```

procedure TPokerMain.CoursBtnClick(Sender : TObject) ;
var
n : byte ;
HelpOK : Bool ;
data : LongInt ;
fichier_aide_appli, chemin_aide : string ;
begin
fichier_aide_appli := application.HelpFile ;
application.helpFile := PrgQuestions.entete.aide ;
chemin_aide := ExtractFilePath(PrgQuestions.entete.aide) ;
n := length(chemin_aide) ;
if chemin_aide[n] = '\' then delete(chemin_aide,n,1) ;
chdir(chemin_aide) ;
try
with PrgQuestions do
data := Questions[question_courante].HelpId ;
if data > 0 then HelpOK := Application.HelpContext(data)
else if data = 0 then HelpOK := Application.HelpCommand(HELP_CONTENTS,data)
else if data < 0 then HelpOK := Application.HelpCommand(HELP_CONTEXTPOPUP,Abs(data)) ;
finally
application.helpFile := Fichier_aide_appli ;
chdir(pokerDir) ;
end ;
end ;

```

CHOISIR SON OUTIL DE CRÉATION HYPERTEXTE

Une des caractéristiques du programmeur amateur est que, bien souvent, il ne maîtrise qu'un seul langage, qu'il essaye d'utiliser pour toutes les tâches qu'il entreprend. C'est évidemment la source de bien des déboires et de difficultés inutiles. Si l'aide de Windows est un outil

agréable à utiliser, elle ne permet pas, et de loin, de venir à bout de toutes les réalisations multimédia.

Rappelez-vous que, malgré les extensions que nous avons vues ensemble, il reste impossible d'utiliser de véritables variables dans vos fichiers HLP. Si vos projets impliquent une personnalisation réelle, des calculs, une grande interactivité, il faudra certainement abandonner le logiciel Winhelp pour vous tourner vers des outils plus performants, comme Toolbook, par exemple.

HLP ou HTML

En outre, je m'interroge sur l'avenir à terme du format HLP. L'explosion d'Internet conduit à un développement du langage d'hypertextes HTML, qui lui est propre. Les premiers balbutiements me laissaient croire à une durable supériorité de l'aide Windows, tant les possibilités d'HTML étaient limitées. Les actuelles extensions du langage l'amènent maintenant à un niveau intéressant, et les outils de création disponibles progressent de façon sidérante. Toutes les nouvelles suites bureautiques et jusqu'aux petits intégrés proposent ou vont proposer en standard la sauvegarde des textes au format HTML.

Microsoft a d'ailleurs à l'une ou l'autre occasion laissé entendre de façon officieuse que les applications destinées à la prochaine version de Windows se verraient dotées d'une aide au format HTML et non plus HLP. J'ai déjà devant moi des CD-Rom réalisés entièrement en HTML...

Il n'en reste pas moins que l'aide Windows n'a pas besoin d'un logiciel de visualisation à rajouter sur la machine. Les pages HTML sont parfois lourdes à porter d'une machine à l'autre : les images sont externes, et il ne faut pas oublier de les recopier toutes. La multiplication des fichiers pose souvent des problèmes d'arborescence. L'absence de compilation est un plus, en terme de facilité de création, et un moins pour l'efficacité du produit.

Nous verrons bien. Une chose est sûre, Internet conduit à la banalisation absolue de l'hypertexte et des outils multimédia. Et si vous savez créer des séquences pédagogiques intéressantes avec l'aide de Windows, vous ferez aussi de petites merveilles avec des pages HTML.

Pierre KESSLER

Service de formation informatique
CRDP - 23 rue du Mal. Juin
67007 Strasbourg Cedex