

VbNews 2 Fichier d'aide au développement sous Visual Basic

[Ce fichier](#)

[Les autres fichiers](#)

[A Venir](#)

Rubriques :

[Bugs](#)

[Fichiers](#)

[Sharewares](#)

[Trucs & astuces](#)

[Utilisation des bibliothèques dynamiques](#)

[Windows](#)

Ce fichier d'aide

Visual Basic est un langage de programmation qui permet de réaliser des applications sous Windows. Sa simplicité d'apprentissage, sa puissance, son orientation "objet" en font le langage indispensable du programmeur Windows.

Qu'est ce que VB-News ? :

Vb-News est un journal, en français !, entre ceux qui s'intéressent à ce langage de programmation.

Que contient t'il ? :

Une foule de rubriques pour vous aider à programmer avec Vb

- Des codes sources documentés à tous les coins de rubriques.
- Une liste (non exhaustive) des "bugs" connus.
- Une sélection de "sharewares" actuellement disponible (autour de Vb bien sûr).
- Une rubrique "trucs & astuces", ça peut servir.

Mais encore ? :

[Enregistrer](#)

[Diffuser](#)

[Erratum](#)

Enregistrement

Si ce programme d'aide vous est utile vous devez l'enregistrer. Trois modes d'enregistrements sont possibles :

- 1 - Obtenir des infos régulières par courrier sur ce journal (c'est gratuit).
- 2 - Etre enregistré et recevoir une mise à jour de Vb-News. Faites parvenir un chèque d'un montant de 90 Frs.
- 3 - Apporter mes propres idées. Envoyez votre disquette (au format 3 " 1/2 , ces disquettes étant moins fragiles). Après tests de vos créations , vous recevrez gratuitement une mise à jour de ce journal, et vous y figurerez!

Où enregistrer ? :

Remplissez le coupon réponse obtenu en imprimant le fichier VBNEWS.ENR et faites le parvenir à :

Jamel BENNADJA
VB-NEWS
93 rue du général LECLERC
95320 ST LEU LA FORÊT

Diffusion

Ce produit est un "shareware". Vous pouvez le diffuser autour de vous , toutefois vous devez fournir sans modification, les fichiers suivants :

- VBNEWS.ENR coupon réponse (enregistrement de ce logiciel).
- VBNEWS.HLP ce fichier
- VBNEWS.LIR fichier texte d'accueil.

Les répertoires :

- SHARE Contenant les "sharewares" décrits.
- SRC Contenant des fichiers source.
- TXT Contenant des fichiers au format texte.

Ce fichier d'aide

Malgré tout le soin apporté à ce journal, ni les auteurs cités, ni moi même ne pouvons assurer que les codes sources fournis sont exempts de "bugs", tenez moi au courant afin de réparer au plus vite les éventuelles erreurs. Les conseils sont aussi les bienvenus!. Merci d'envoyer un courrier à l'adresse suivante:

Jamel BENNADJA
VB-NEWS
93 rue du général LECLERC
95320 ST LEU LA FORÊT

Autres fichiers

Avec VbNews vous trouverez :

- Un répertoire "SRC" où vous trouverez des fichiers exemple sur les thèmes abordés dans Vb-News. Depuis Visual Basic 1.0 le format d'enregistrement change à chaque version, les exemples sont au format Visual Basic 2.0, cette version semblant être la première vraiment exploitable.
- Un répertoire "SHARE" où vous trouverez les "sharewares" décrits.
- Un répertoire "TXT" où vous trouverez des fichiers au format texte du "bloc-note" de Windows. Ces fichiers décrivent essentiellement des parties du fichier CONSTANT.TXT fourni avec Visual Basic.

Rubriques à venir

- Le DDE ou comment parler à d'autres applications.
- Suite de la dissection de CONSTANT.TXT.

Bugs

dates

fonction "Trim\$"

dates

Tout comme avec les anciens Basics, la date système est renvoyée au format anglais, nécessitant des routines de substitution de chaînes.

Fonction Trim\$

Trim\$ doit enlever les espaces contenus devant et derrière une chaîne en fait parfois au retour de la fonction il reste un espace à droite de la chaîne. Utilisez plutôt (même si c'est plus long) :

```
Ltrim$(Rtrim$(Chaine$))
```

Fichiers



Les fichiers d'initialisation (.INI)



Les fichiers daide (.HLP)



Le fichier CONSTANT.TXT

Fichiers .ini

L'accès aux fichiers d'initialisation (.ini) ne peut se faire que par l'intermédiaire des "dll" Windows. Ce concept est très intéressant car il peut vous permettre de sauvegarder des paramètres de votre application sans avoir recours à des "Open", "Get" et autres commandes Vb sur les fichiers.

On peut distinguer deux groupes de commandes :

- Celles qui concernent le fichier d'initialisation Win.ini.
- Celles qui concernent les fichiers d'initialisations personnels.

La structure d'un fichier ini est toujours la même :

```
[ Nom_D'une_entrée ]  
Paramètre_1 = 300  
Suite des paramètres .....  
[ Nom_D'une_entrée ]  
Paramètre_1 = Novembre  
Suite des paramètres .....  
etc ...
```

La syntaxe d'appel et les paramètres à passer lors de l'appel sont rigoureusement les mêmes, seuls les noms d'appels changent. Microsoft conseille l'utilisation des fichiers "privés" plutôt que Win.ini, en effet celui-ci est lu à chaque chargement de windows, un trop grand nombre de paramètres personnels l'encombre inutilement et ralentit le lancement de Windows.

Le code source fourni contient les déclarations des "dll" de lecture et d'écriture d'un fichier d'initialisation personnel, en remarque le nom de l'équivalent pour Win.ini, une procédure pour lire une entrée, une pour écrire une entrée.

Création d'un fichier d'aide

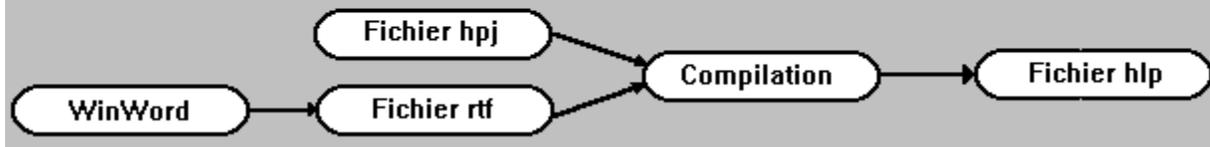
Généralités

Les fichiers d'aides sous Windows sont des fichiers structurés dit "hypertexte" accessibles directement à partir d'une application c'est la raison pour laquelle beaucoup de logiciels utilisent cette possibilité, la lecture d'une notice se révélant vite lassante. De plus grâce à son architecture un fichier d'aide peut être consulté directement en fonction d'un thème, d'une action à effectuer.

Création d'un fichier d'aide

Appel à partir de Visual Basic

Création d'un fichier d'aide



Comme décrit ci-dessus il faudrait disposer dans l'idéal :

- D'un traitement de texte capable de générer un fichier au format "RTF" Microsoft, bien sûr, en propose un c'est WinWord (ça c'est contournable).
- Du compilateur hc.exe fourni en autre dans le "SDK" .

Phases de création

1 - Fichiers au format "RTF"

Ce format trouve son utilité lorsque l'on souhaite échanger des fichiers entre diverses machines Le texte étant entièrement décrit (fontes, liens hypertexte...). C'est le format d'entrée du compilateur. Plusieurs logiciels "sharewares" se proposent de générer des fichiers rtf à partir d'un éditeur ASCII ou ANSI (ANSI étant la norme utilisée par Windows), parmi ceux-ci, vous en trouverez un parmi les "sharewares" fournis c'est "QdHelp.exe".

2 - Fichiers "HPJ"

C'est un fichier texte qui va indiquer au compilateur quels sont les éléments à inclure (fichier rtf, bitmaps ...) sa structure est la suivante:

Entrée [OPTION]

INDEX= Indiquez là le nom de la fenêtre principale.

COMPRESS= true compression du fichier, false sans compression.

TITLE= Titre qui sera affiché dans la barre de menus.

Entrée [FILES]

Indiquez là le nom du fichier rtf à compiler.

Entrée [MAP]

Cette entrée est décrite dans le chapitre "Appel à partir de Visual Basic"

Entrée [BITMAPS]

Indiquez là le nom du ou des fichiers bitmaps à inclure (extension .bmp)

3 - Compilation

Le fichier hpj étant correctement rempli, il suffit, au prompt du DOS, d'appeler le compilateur par la commande :

hc [Nom du fichier .hpj]

Si tout va bien le fichier d'aide est créé.

Appel à partir de Visual Basic

Vous trouverez la déclaration à ajouter au module global en cliquant sur "[, > >](#)"
Le fichier Aide.Txt contient les constantes suivantes :

HELP_HELPONHELP : Appel de l'aide de l'aide.

HELP_INDEX : Appel de la fenêtre de départ du fichier d'aide.

mais surtout :

HELP_CONTEXT : Appel de l'aide sur un sujet précis.

En effet beaucoup de contrôles ont une propriété HelpContextID. C'est là qu'intervient l'entrée [MAP] du fichier .hjp. Un Item de menu peut notamment avoir une valeur HelpContextID. Donnez une valeur supérieure à 100 exemple 101. Dans l'entrée [MAP] insérer un nom de rubrique (commande /topic dans QdHelp) suivi d'un espace puis de 101 dans notre exemple. Ainsi lorsque l'utilisateur "clicquera" sur ce menu l'aide sera lancée et la fenêtre correspondant à l'indice 101 sera directement affichée.

Voir répertoire src.Projet aide.mak

Le fichier Constant.Txt

Rappel sur les constantes :

Une constante, comme son nom l'indique, reste invariable pendant toute la durée de l'exécution de votre programme. Les fonctions Windows doivent être souvent appelées avec un nombre importants de paramètres, dont beaucoup sont des constantes. Plutôt que d'avoir à se souvenir que, par exemple, la touche "Return" a pour valeur "&HD" autant déclarer une fois pour toutes :

```
CONST RETURN = &HD
```

C'est pourquoi Visual Basic est livré avec un fichier contenant un maximum de constantes Windows. Avec toutefois quelques inconvénients :

- Impossible de le charger avec votre application (fichier trop long).
- Très peu documenté.

Avec VbNews vous trouverez dans le répertoire "TXT" quelques constantes regroupées en fonction d'un thème. Ainsi si votre projet utilise des constantes relatives à l'aide, il vous suffira de copier le contenu du fichier aide.txt dans le module global de votre application.

Les fichiers TXT disponibles actuellement sont :

Couleurs.txt : Utilisation des couleurs système
Aide.txt : Si vous utilisez l'aide Windows

Trucs & astuces

[Ascenceurs](#)

[Raccourcis clavier](#)

[Lenteur de chargement](#)

[Lenteur de chargement \(suite\)](#)

[Lenteur de chargement \(suite et fin\)](#)

[Application en cours ?](#)

[Palette de couleurs](#)

Ascenceurs

A l'exécution, lorsque l'utilisateur fait défiler un ascenseur celui-ci se met ensuite à clignoter, donnant une visualisation désagréable. Afin d'éviter cela donner le focus (commande setfocus) à une image ou à une étiquette présente dans la fenêtre et ce à la fin de la procédure Change de l'ascenseur, cela provoquera l'arrêt du clignotement.

exemple : `Form1.Image1.setfocus`

Raccourcis clavier

Vous pouvez aller notablement plus vite lors de la programmation en apprenant des raccourcis clavier, en voici quelques uns :

Ctrl + Y coupe la ligne où se trouve le curseur et la met dans le presse-papier.

Maj + Ins place le contenu du presse-papier sur la ligne courante

Ctrl+N crée une ligne vide au dessus du curseur.

A l'exécution dans l'environnement en cas d'erreur avant de "cliquer" sur "Ok" tapez sur "F1" et vous aurez une aide plus complète sur l'erreur survenue.

Vous pouvez copier plusieurs contrôles, images, zones de textes ...etc et ce d'un seul coup. Pour cela sélectionnez les éléments à copier en cliquant dessus tout en maintenant la touche Ctrl du clavier enfoncée. Lorsque vous avez fini la sélection taper sur Ctrl+Ins, l'ensemble est mis dans le presse-papier, il suffit de faire Maj + Ins dans la nouvelle fenêtre et tout est copié!.

Lenteur de chargement

Dès que l'application que vous avez créée devient conséquente le temps de chargement de celle-ci devient insupportable. De plus, pendant le temps de chargement le curseur passe alternativement de la forme sablier à la forme flèche.

Même s'il n'y a pas pour l'instant de véritable solution à ce problème on peut tout de même essayer l'astuce suivante.

Le but consiste à faire patienter l'utilisateur pendant le chargement de votre programme pour cela créez une fenêtre contenant juste deux éléments :

- 1 - Une étiquette avec un message du style "Chargement en cours"
- 2 - Une minuterie avec la propriété Interval à 1

Pour la fenêtre pas de Max ni de Min button, le borderstyle à aucun et pas de controlbox. Designez la comme feuille de démarrage.

A l'évènement Timer de la minuterie dévalidez le timer, faites un load de la feuille principale de votre application et donnez sa propriété visible à faux. Positionnez le curseur souris à l'intérieur de la fenêtre puis mettez sa forme en sablier.

A l'évènement Timer de la minuterie dévalidez le timer, faites un load de la feuille principale de votre application et donnez sa propriété visible à faux.

A l'évènement load de la feuille réalisez toutes les affectations de variables et initialisations diverses (vous pouvez tenir l'utilisateur informé grâce à l'étiquette de la fenêtre de démarrage). Faites un Unload de la première feuille, donnez la propriété visible à vrai, redonnez la forme standard

Et voilà le tour est joué !

Vb-Tips Janvier 1992

Lenteur de chargement (suite)

la principale lenteur d'exécution d'une application Visual-Basic est due au chargement préalable de la bibliothèque "VBRUNX00.DLL" (plus de 400 K-Octets avec la version 3). Un des moyens de contourner cet inconvénient est de charger la bibliothèque en mémoire et ce dès le lancement de Windows.

Pour cela :

- Lancez un nouveau projet, supprimer la fenêtre "Feuille1.frm" et autre "Vbx"
- Créez un nouveau module : en principe "module1.bas"
- Dans ce module ajoutez :

a) Dans la section "déclarations"

```
Declare Function LoadLibrary Lib "Kernel" (ByVal lpLibFileName As String) As
```

Integer

b) Dans la procédure "main" ajoutez le code suivant :

```
Out% = LoadLibrary ("VbRunX00.Dll") ' avec "X"représentant la version de
```

Visual-Basic

```
End
```

Après avoir défini le module "main" comme étant la fenêtre de démarrage de votre projet Il suffit d'ajouter le programme après "création d'un exe" à votre groupe de démarrage de Windows

Lenteur de chargement (suite et fin)

Vous pouvez aussi ajouter la commande "Show" au début de la procédure "Form_load" de votre projet

Cela obligera l'affichage de la fenêtre dès que possible

Application en cours ?

Il peut être intéressant de savoir si l'application que l'utilisateur tente de charger n'est pas déjà en cours d'exécution.

Pour cela au début de la procédure Form_Load saisissez le code suivant:

```
If App.PrevInstance = True Then
    ' L'application est déjà en cours d'exécution.
End
End If
```

Palette de couleur

Afin d'avoir rapidement une palette de couleurs, appeler la fenêtre des propriétés (F4), puis "double cliquez" sur une propriété Foreground ou Background une palette de couleurs apparaîtra à l'écran.

Utilisation des D.L.L

Les bibliothèques externes permettent d'accéder à des fonctions non disponibles dans un langage de programmation, l'inconvénient majeur était qu'il fallait les intégrer à chaque programme compilé et ce lors de l'édition de liens. Sous Windows deux fonctionnalités importantes ont été ajoutées :

- Plus besoin d'ajouter la bibliothèque il suffit de déclarer correctement la fonction que l'on souhaite utiliser.
- Lorsque la bibliothèque est chargée en mémoire elle devient accessible à tous les programmes en cours d'exécution.

Avec Vb vous pouvez appeler les trois dll fournies en standard sous Windows qui sont **GDI**, **USER** et **KERNEL** les déclarations devant impérativement tenir sur une seule ligne et figurer dans la section "Global" de votre projet.

En voici quelques unes regroupées par thème:



Aide

1 - Appeler un fichier d'aide.

```
Declare Function WinHelp Lib "User" (ByVal hWnd as Integer, ByVal lpHelpFile As String,  
ByVal_  
wCommand as Integer, ByVal dwData as Long) As Integer
```

Appel : Out% = WinHelp (MaFenêtre.hWnd,"Toto.hlp",HELP_HELPINDEX, 0)

Out% est vraie si cela c'est bien passé, Faux dans l'autre cas. Tapez " ≤ <" pour plus d'infos.

Fenêtres

1 - Imposer votre fenêtre, pas de changement à l'aide de [ALT TAB] ou [CTRL ESC]

Declare Function SetSysModalWindow Lib "User" (ByVal hWnd As Integer) As Integer

Appel : Out% = SetSysModalWindow(hWnd) ' hWnd renvoie le numéro d'ordre de votre fenêtre

note : Votre fenêtre devient "SystemModal", vous n'avez plus accès à une autre fenêtre que celle la. Attention en cours de développement ,en cas de "bug" vous n'avez plus qu'à "rebooter".

Fichiers

1 - Ecrire dans un fichier d'initialisation privé.

```
Declare Function GetPrivateProfileString Lib "Kernel" (ByVal lpApplicationName As String, _  
ByVal KeyName As String, ByVal lpDefault As String, ByVal lpReturnedString As String, _  
ByVal nSize As Integer, ByVal lpFileName As String) As Integer
```

Appel : Voir Répertoire src.Projet ini.mak

2 - Ecrire dans un fichier d'initialisation privé.

```
Declare Function WritePrivateProfileString Lib "Kernel" (ByVal lpApplicationName As String, _  
lpKeyName As String, ByVal lpString As String, ByVal lpFileName As String) As Integer
```

Appel : Voir répertoire src.Projet ini.mak

3 - Lire une entrée dans Win.ini

```
Declare Function GetProfileString Lib "Kernel" (ByVal lpApplicationName As String, _  
ByVal KeyName As String, ByVal lpDefault As String, ByVal lpReturnedString As String, _  
ByVal nSize As Integer,) As Integer
```

Appel : Voir répertoire src.Projet ini.mak

4 - Ecrire une entrée dans Win.ini.

```
Declare Function WriteProfileString Lib "Kernel" (ByVal lpApplicationName As String, _  
lpKeyName As String, ByVal lpString As String) As Integer
```

Appel : Voir répertoire src.Projet ini.mak

5 - Charger une bibliothèque en mémoire.

```
Declare Function LoadLibrary Lib "Kernel" (ByVal lpLibFileName As String) As Integer
```

Appel : Out% = LoadLibrary ("VbRun300.dll") ' charge vbrun300 en mémoire

Out%=0 pas assez de mémoire.Out%=2 fichier non trouvé

Note : Voir lenteur de chargement (suite)

Souris

1 - Positionner la souris.

Declare Sub SetCursorPos Lib "user" (ByVal x%, ByVal y%)

Appel : SetCursorPos 320, 240 ' souris positionnée au centre de l'écran (VGA)

2 - Montrer ou cacher le curseur souris.

Declare Function ShowCursor Lib "user" (ByVal bShow As Integer)

Appel : ShowCursor 0 ' décrémente le compteur de visibilité de la souris (si ShowCursor a été appelé 2 fois avec une valeur différente de 0 le curseur sera toujours visible). Le compteur est fixé à zéro si une souris est installée et à -1 dans le cas contraire.

Windows

1 - Quitter Windows à partir de Visual Basic

```
Declare Function ExitWindows Lib "user" (ByVal dwReserved, ByVal wReturncode As Long) _  
As Integer
```

```
Appel : Out% = ExitWindows (0, 0)
```

Windows

Couleur des fichiers d'aides

Couleurs système

Couleurs dans les fichiers d'aide

Par défaut les rubriques (mots soulignés) apparaissent en vert. Avec une carte graphique affichant 16 couleurs c'est lisible, il en va tout autrement avec une carte 256 couleurs.

Vous pouvez changer cette couleur par une couleur de votre choix et ce d'une manière très simple :

- A l'aide du bloc-notes éditez le fichier "Win.ini"
- Rendez vous à la section [Windows Help]
- Créez ou modifiez l'entrée JumpColor

,exemple : `JumpColor=255 0 0` donnera des caractères rouges (valeur RGB)

Couleurs système sous Visual Basic

Lorsque vous fixez la couleur d'un contrôle ou d'une feuille à l'aide de la palette des couleurs, Visual Basic code cette couleur à l'aide d'un "long integer". Ce codage représente une valeur "RGB" ou "RVB" en français. Ainsi le bleu le plus intense aura la valeur "&H0000FF" (notez que cette valeur est exprimée en hexadécimal ce qui donne en théorie plus de 16 millions de couleurs possibles). En définissant les couleurs par cette méthode lorsque l'utilisateur changera les couleurs système via le panneau de configuration votre application ne suivra pas cette modification. Si vous souhaitez que les couleurs de votre application respectent les couleurs définies dans le panneau de configuration de Windows, vous devez définir vos couleurs par référence. Les couleurs ne sont plus décrites sur 6 chiffres mais sur 8 le premier étant le chiffre 8 indiquant la valeur "système". Ajouter en dernier un chiffre correspondant au type de contrôle.

exemple : `Menu.BackColor = &H80000004` donnera la couleur de fond à une barre de menu

Le fichier Couleurs.Txt provenant de Constant.txt vous donne toutes les déclarations de constantes possibles. Ainsi:

`Feuille1.BackColor = FOND_FENETRE` ou
`Feuille1.BackColor = WINDOW_BACKGROUND` si vous utilisez les constantes en anglais.

Fera que la feuille créée aura toujours la couleur de fond choisie dans le panneau de configuration. Ainsi lorsque l'utilisateur changera les couleurs système votre application respectera ce choix.

Sharewares



Titre : Ico2bmp

Auteur : George CAMPBELL

Coût : \$0

But : Transformer un fichier de type icône (.ico) en fichier image (.bmp).Ça peut servir, la preuve : les bitmaps utilisés dans ce fichier ont été réalisés avec ce programme.



Titre : FileBox

Auteur : Thomas KIEHL

Coût : \$0

But :Obtenir une boîte de dialogue de type "fichier,ouvrir" nettement meilleure que celle obtenue en utilisant les contrôles Vb,fourni avec code source.



Titre : FrameVB

Auteur : Nelson FORD

Coût : \$0

But :Permet d'enrober des images ou listes de fichier à la façon 3d et ce sans "dll" ,fourni avec code source.



Titre : QDHelp

Auteur : Phil ALLEN

Coût : \$20

But : permettre de générer des fichiers au format "rtf" sans disposer d'un traitement de texte coûteux.Bien fait malgré son nom "Quick and Dirty Help".



Titre : StsBar

Auteur : Ed STAFFIN

Coût : \$10

But :Démo sur la "dll" status qui comme son nom l'indique vous ajoute un contrôle personnalisé permettant de créer une barre de status à la mode Windows 3.1 ce qui ne gâche rien.Notez que l'exemple fourni n'a pas l'air de fonctionner ce qui n'enlève rien à StsBar.



Titre : VbCards

Auteur : Richard R SANDS

Coût : \$0

But : Créer un jeu utilisant des cartes n'est pas chose facile. Ce programme est une démo montrant la puissance de VBCARDS.DLL. Les 52 cartes sont bien sûr disponibles, des dos de cartes (avec Marilyn en plus!), le joker et d'autres images.

de l'anglais "to share : partager" et "ware : pour faire joli"

Microsoft © Visual Basic

Dynamic Link Library

Ritch Text Format

Software Development Kit de Microsoft ®

