



FAQ WINDEV

*Version 0.1
(Février 2004)*

Frédéric Blanc
<http://www.tutorea.tk/>
tutorea@wanadoo.fr



TABLE DES MATIERES

I. CATEGORIE : WINDEV API 4

1.	WINDEV5.5 RESSOURCES BASSES	4
2.	SIMULER UN CLIC SUR LE BOUTON D'UNE AUTRE APPLICATION	4
3.	MODIFIER LA RESOLUTION DE L'ECRAN	4
4.	LISTER LES PROCESSUS NT	7

II. CATEGORIE: WINDEV CODES 9

1.	HIMPORTETEXTE() SANS DELIMITEUR DE CHAINES	9
2.	FICHER EXTERNE : LIRE LE CONTENU HEXADECIMAL	9
3.	CONVERSION OCTET – BINAIRE	10
4.	IMPRIMER UNE PAGE DE SITE INTERNET	11
5.	LIRE UN CHAMP APPARTENANT A UNE AUTRE FENETRE.....	11
6.	APPELER UN TRAITEMENT PAR UNE TOUCHE	12
7.	REBOOTER UN PROJET EN EXECUTION	12
8.	CONVERSION DECIMAL – HEXADECIMAL	12
9.	INACTIVER UNE TOUCHE CLAVIER	13
10.	SUPPRIMER UNE LIGNE DANS UN FICHER TEXTE	13
11.	MONGROUPE..ETAT NE CONTIENT PAS LA VALEUR PREVUE.....	13
12.	..VISIBLE AVEC UNE FENETRE COMPORTANT DES PLANS	14
13.	COLONNES DE TABLES QUI DEVIENNENT INEXISTANTES EN EXECUTION	14
14.	SUPPRIMER UNE SOUS CHAINE DANS UNE CHAINE	15

III. CATEGORIE: WINDEV CODES HF 15

1.	CACHER ET RECUPERER DES ENREGISTREMENTS.....	15
2.	AFFICHER LES COMPOSANTES DES CLES COMPOSEES	16
3.	SQL INSERT : RECUPERER LE DERNIER IDENTIFIANT AUTO (HF & MYSQL).....	16

IV. CATEGORIE : WINDEV ETATS 17

1.	ETAT SUR UN TABLEAU DYNAMIQUE.....	17
2.	BOOLEENS, DE BELLES COCHES ET PLUS DE CARRES VIDES	17
3.	APPLIQUER LES MODIFICATION IPARAMETRE() POUR CHANGER DE BAC	17

V. CATEGORIE : FENETRES..... 18

1.	ONGLETS, ORDRE DE NAVIGATION ET TOUCHE ENTREE	18
2.	REPERTOIRES DANS UN TREEVIEW	18
3.	INFO / ERREUR / OUI/NON PERSONNALISES	21
4.	SUPPRIMER UN CHAMP CREE AVEC CHAMPCLONE()	21
5.	CALCULER LE CRC 16 UTILISE DANS LE PROTOCOLE XMODEM	21
6.	COMMENT SAVOIR SI UNE FENETRE EST OUVERTE OU FERMEE ?	22
7.	COMMENT AFFICHER UN DOCUMENT PDF DANS UNE FENETRE ?	22
8.	COMMENT DONNER LE FOCUS A UN CHAMP PARTICULIER ?.....	23
9.	COMMENT DONNER LE FOCUS A UNE LIGNE D'UNE TABLE LORS D'UN CLIC DROIT ?	23

VI. CATEGORIE : WINDEV PROJET..... 23

1.	IDENTIFICATEUR XXX INCONNU OU INACCESSIBLE ICI.....	23
2.	ICONE PERSONNALISEE DANS LES BARRES DE TITRE.....	23

3. CONNAITRE LE REPERTOIRE DES DONNEES EN INSTALLATION RESEAU 24

VII. CATEGORIE : WINDEV DIVERS..... 24

1. WINDEV ET LES CARTES SOUNDBLASTER..... 24
2. TROUVER LE PILOTE ODBC HYPERFILE..... 24

VIII. CATEGORIE: WINDEV GABARITS/DESIGN 25

1. MAGENTA DES BMP ET BOUTONS 25

Remarques diverses : WinDev, WebDev et HyperFile sont des marques déposées par [PCSoft](http://www.pcsoft.com).
Cette FAQ est inspirée des publications parues dans les forums du site <http://rbesset.net>. Ce document est sous licence GNU.

I. CATEGORIE : WINDEV API

1. WINDEV5.5 Ressources Basses

Emis par Kristian Paradis sur le forum daussy.org

"Bonjour a tous,

Je suis sur que je ne suis pas le seul qui lorsqu'il démarre l'application en mode test se fait souvent dire par WinDev qu'il reste moins de 25% de ressources gdi et user blablabla....

J'ai essayé cette astuce que j'ai trouvée sur un site de hardware (<http://www.inpact-hardware.com/>) et ça semble bien fonctionner. (je n'ai pas eu le message depuis) dans win.ini section Windows ajouter la ligne:

GDIMaxHeapSize=**32767**

ATTENTION! 32768 a la valeur UserMaxHeapSize rend impossible le démarrage de Windows... même en mode sans échec!!!

Ne pas mettre plus, j'ai essayé et Windows refuse de démarrer GDI.exe et on me demande de réinstaller Windows 🤔(même chose avec 32768).

J'espère que ça réglera les ennuis de plusieurs 😊

Bon développement

Kristian Paradis

2. Simuler un clic sur le bouton d'une autre application

Question postée sur le forum de daussy.org

réponse de freddyp@wanadoo.fr :

```
lpBName est chaine asciiz de 128
lpBClass est chaine asciiz de 128

lpBName='Titre Fenetre' // Titre de la fenetre contenant le bouton
lpBClass='#32770' // Classe de la fenetre
NumFen est entier
long=Appeldll32('USER32','FindWindowExA',Null,Null,&lpBClass,&lpBName)
si NumFen>0 alors
    lpBName='Bouton' // Nom du bouton
    lpBClass='Button' // Class Button
    NumBtn est entier
    long=Appeldll32('USER32','FindWindowExA',NumFen,null,&lpBClass,&lpBName)
    si NumBtn>0 alors
        Post3 =postmessage(NumBtn,245,null,null) // BM_CLICK
    fin
fin
fin
```

3. Modifier la résolution de l'écran

Auteur du code : Philippe RAMIN

Déclaration de la structure DEVMOD dans le code du projet

```

DEVMOEA est une structure
dmDeviceName est une chaîne fixe sur 32 //Type C : BYTE
dmSpecVersion est un entier sur 2 octets //Type C : WORD
dmDriverVersion est un entier sur 2 octets //Type C : WORD
dmSize est un entier sur 2 octets //Type C : WORD
dmDriverExtra est un entier sur 2 octets //Type C : WORD
dmFields est un entier //Type C : DWORD
dmOrientation est un entier sur 2 octets //Type C : short
dmPaperSize est un entier sur 2 octets //Type C : short
dmPaperLength est un entier sur 2 octets //Type C : short
dmPaperWidth est un entier sur 2 octets //Type C : short
dmScale est un entier sur 2 octets //Type C : short
dmCopies est un entier sur 2 octets //Type C : short
dmDefaultSource est un entier sur 2 octets //Type C : short
dmPrintQuality est un entier sur 2 octets //Type C : short
dmColor est un entier sur 2 octets //Type C : short
dmDuplex est un entier sur 2 octets //Type C : short
dmYResolution est un entier sur 2 octets //Type C : short
dmTTOption est un entier sur 2 octets //Type C : short
dmCollate est un entier sur 2 octets //Type C : short
dmFormName est une chaîne fixe sur 32 //Type C : BYTE
dmLogPixels est un entier sur 2 octets //Type C : WORD
dmBitsPerPel est un entier //Type C : DWORD
dmPelsWidth est un entier //Type C : DWORD
dmPelsHeight est un entier //Type C : DWORD
dmDisplayFlags est un entier //Type C : DWORD
dmDisplayFrequency est un entier //Type C : DWORD
dmICMMethod est un entier //Type C : DWORD
dmICMIntent est un entier //Type C : DWORD
dmMediaType est un entier //Type C : DWORD
dmDitherType est un entier //Type C : DWORD
dmICCManufacturer est un entier //Type C : DWORD
dmICCModel est un entier //Type C : DWORD
dmPanningWidth est un entier //Type C : DWORD
dmPanningHeight est un entier //Type C : DWORD
FIN

CONSTANTE DM_BITSPERPEL=262144
CONSTANTE DM_PELSWIDTH=524288
CONSTANTE DM_PELSHEIGHT=1048576
CONSTANTE DM_DISPLAYFREQUENCY=4194304

CONSTANTE CDS_TEST=2
CONSTANTE CDS_UPDATEREGISTRY=1
CONSTANTE CDS_TEMP=0

```

Listage des résolutions dans un tableau

```

// code de la fenetre :
dm est une DEVMOEA
iModeNum est un entier
TANTQUE API('USER32','EnumDisplaySettingsA',Null,iModeNum,&dm)
TableAjouteLigne(TableResolutions,dm:dmDeviceName,dm:dmBitsPerPel,dm:dmPelsW
idth,dm:dmPelsHeight,dm:dmDisplayFrequency)
  iModeNum++
FIN

```

Bouton1 = Check si la résolution est Ok (ne modifie rien)

```
SI TableSelectOccurrence(TableResolutions)=0 ALORS RETOUR
dmtemp est une DEVMODEA
dmtemp:dmBitsPerPel=TableResolutions.dmBitsPerPel
dmtemp:dmPelsHeight=TableResolutions.dmPelsHeight
dmtemp:dmPelsWidth=TableResolutions.dmPelsWidth
dmtemp:dmDisplayFrequency=TableResolutions.dmDisplayFrequency
dmtemp:dmFields=OUBinaire(dmtemp:dmFields,DM_BITSPERPEL)
dmtemp:dmFields=OUBinaire(dmtemp:dmFields,DM_PELSWIDTH)
dmtemp:dmFields=OUBinaire(dmtemp:dmFields,DM_PELSHEIGHT)
dmtemp:dmFields=OUBinaire(dmtemp:dmFields,DM_DISPLAYFREQUENCY)
dmtemp:dmSize=Dimension(dmtemp)
SI API('user32','ChangeDisplaySettingsA',&dmtemp, CDS_TEST)<>0 ALORS
  Erreur(ErreurInfo())
FIN
```

Bouton2 = Modification temporaire de la résolution

```
// Idem que le bouton 1 mais mettre le flag à 0 //

SI TableSelectOccurrence(TableResolutions)=0 ALORS RETOUR
dmtemp est une DEVMODEA
dmtemp:dmBitsPerPel=TableResolutions.dmBitsPerPel
dmtemp:dmPelsHeight=TableResolutions.dmPelsHeight
dmtemp:dmPelsWidth=TableResolutions.dmPelsWidth
dmtemp:dmDisplayFrequency=TableResolutions.dmDisplayFrequency
dmtemp:dmFields=OUBinaire(dmtemp:dmFields,DM_BITSPERPEL)
dmtemp:dmFields=OUBinaire(dmtemp:dmFields,DM_PELSWIDTH)
dmtemp:dmFields=OUBinaire(dmtemp:dmFields,DM_PELSHEIGHT)
dmtemp:dmFields=OUBinaire(dmtemp:dmFields,DM_DISPLAYFREQUENCY)
dmtemp:dmSize=Dimension(dmtemp)
SI API('user32','ChangeDisplaySettingsA',&dmtemp, CDS_TEMP)<>0 ALORS
  Erreur(ErreurInfo())
FIN
```

Bouton3 = Modification permanente (maj de la base de registre)

```
// Idem que le bouton 1 mais mettre le flag à 1 //

SI TableSelectOccurrence(TableResolutions)=0 ALORS RETOUR
dmtemp est une DEVMODEA
dmtemp:dmBitsPerPel=TableResolutions.dmBitsPerPel
dmtemp:dmPelsHeight=TableResolutions.dmPelsHeight
dmtemp:dmPelsWidth=TableResolutions.dmPelsWidth
dmtemp:dmDisplayFrequency=TableResolutions.dmDisplayFrequency
dmtemp:dmFields=OUBinaire(dmtemp:dmFields,DM_BITSPERPEL)
dmtemp:dmFields=OUBinaire(dmtemp:dmFields,DM_PELSWIDTH)
dmtemp:dmFields=OUBinaire(dmtemp:dmFields,DM_PELSHEIGHT)
dmtemp:dmFields=OUBinaire(dmtemp:dmFields,DM_DISPLAYFREQUENCY)
dmtemp:dmSize=Dimension(dmtemp)
SI API('user32','ChangeDisplaySettingsA',&dmtemp, CDS_UPDATEREGISTRY)<>0
ALORS
  Erreur(ErreurInfo())
FIN
```

Conclusion

Voilà, un grand merci à Philippe Ramin pour cette trouvaille

4. Lister les processus NT

Merci à DrCharly93 pour ce post sur le NG PCSoft:

Comment lister les processus et modules en cours
Voici un bout de code trouvé il y a longtemps sur Internet

```
Procédure ProcesNT()  
// Liste des process et modules sous NT  
// Ce module pour fonctionner correctement doit être lancer dans une  
// fenêtre disposant d'une table 'TABLE1' composer de 5 colonnes  
// id          numérique entier (index dans la liste)  
// cProcessID  texte  
// cProcessName texte  
// cModuleName texte  
// cPathName   texte  
// Un interrupteur 'INTMODULE' permet de n'afficher  
// que les process (INTMODULE=0)  
// ou les process et modules correspondants (INTMODULE=1)  
  
//  
LOCAL  
  idTabProcess  est un tableau de 200 entiers longs  
  Cb            est un entier long=800 // taille en octets du tableau idTabProcess  
  CbNeeded      est un entier long  
  NumElements   est un entier long  
  
  hTabModule    est un tableau de 200 entiers longs  
  Cb2           est un entier long=800 // taille en octets du tableau hTabModule  
  CbNeeded2     est un entier long  
  NumModules    est un entier long  
  Pname         est une chaine  
  
  FileName      est une chaine fixe sur 260  
  BaseName      est une chaine fixe sur 260  
  nSize         est un entier long=260 // taille de FileName et BaseName  
  
  lret,l1,l2    est un entier long  
  hProcess      est un entier long  
  i,i2          est un entier long  
  NumDll        est un entier long  
  
  NumDll=chargedll('PSAPI.DLL')  
  si NumDll=0 alors  
    erreur('La dll PSAPI n'a pas été chargée')  
    renvoyer faux  
  fin  
  
  //  
  // The EnumProcesses function retrieves the process identifier  
  // for each process object in the system.  
  //  
  // BOOL EnumProcesses(  
  //   DWORD * lpidProcess, // array to receive the process identifiers  
  //   DWORD cb,           // size of the array  
  //   DWORD * cbNeeded    // receives the number of bytes returned  
  // );  
  Lret=appeldll32('PSAPI.DLL','EnumProcesses',&idTabProcess,Cb,&CbNeeded)  
  NumElements=CbNeeded/4 // Taille renvoyée divisée par la taille d'un  
  entier long (4 octets)
```

```

Pour i=1 a NumElements
    //
    // The OpenProcess function returns a handle to an existing process object.
    //
    // HANDLE OpenProcess(
    //     DWORD dwDesiredAccess, // access flag
    //     BOOL bInheritHandle,   // handle inheritance flag
    //     DWORD dwProcessId      // process identifier
    // );
    //

HProcess=appeldll32('KERNEL32.DLL','OpenProcess',oubinaire(1024,16),0,idTabP
rocess[i])
si HProcess<>0 alors // On a un processus. On va rechercher ses modules.
    //
    // The EnumProcessModules function retrieves a handle for each module
    // in the specified process.
    //
    // BOOL EnumProcessModules(
    //     HANDLE hProcess, // handle to the process
    //     HMODULE * lphModule, // array to receive the module handles
    //     DWORD cb, // size of the array
    //     LPDWORD lpcbNeeded // receives the number of bytes returned
    // );
    // Process_Query_Information=1024
    // Process_All_Access=16
    //

Lret=Appeldll32('PSAPI.DLL','EnumProcessModules',Hprocess,&hTabModule,Cb2,&C
bNeeded2)
si Lret<>0 alors
    NumModules=CbNeeded2/4
    SI INTMODULE=0 alors NumModules=1
    Pour i2=1 a NumModules
        // The GetModuleBaseName function retrieves the base name of the
        // specified module.
        //
        //DWORD GetModuleBaseName(
        //     HANDLE hProcess, // handle to the process
        //     HMODULE hModule, // handle to the module
        //     LPTSTR lpBaseName, // buffer that receives the base name
        //     DWORD nSize // size of the buffer
        //);
        L1=Appeldll32('PSAPI.DLL','GetModuleBaseNameA',HProcess,hTabModule[i2]
....
        ,&BaseName,nSize)
        SI L1=0 alors BaseName=''
        Si i2=1 alors Pname=minuscule(BaseName[[1 à L1]])
        //
        // The GetModuleFileNameEx function retrieves the fully qualified path
        // for the specified module.
        //
        // DWORD GetModuleFileNameEx(
        //     HANDLE hProcess, // handle to the process
        //     HMODULE hModule, // handle to the module
        //     LPTSTR lpFilename, // buffer that receives the path
        //     DWORD nSize // size of the buffer
        // );
        //
        //
L2=Appeldll32('PSAPI.DLL','GetModuleFileNameExA',HProcess,hTabModule[i2] ...
        ,&FileName,nSize)

```

```

    Si L2=0 alors FileName=''
    TABLEAJOUTE('TABLE1',i+i2/1000+TAB+idTabProcess[i]+TAB+PName+...
                TAB+Minuscule(BaseName[[1 à L1]])+...
                tab+Minuscule(FileName[[1 a L2]]))
    fin
  fin
  LRet=Appeldll32('KERNEL32.DLL','CloseHandle',HProcess)
FIN
fin

si NumDll<>0 alors
  dechargedll(NumDll)
fin

```

II. CATEGORIE: WINDEV CODES

1. hImporteTexte() sans délimiteur de chaînes

Souvent les fichiers à importer ne comportent que des délimiteurs de colonnes et de lignes. Les chaînes ne sont pas encadrées par un délimiteur.

Pour exploiter ces fichier avec la hImporteTexte (plus performante qu'un parcours), il faut définir la chaîne des délimiteur comme suit :

```

cSep est une chaîne
cSep=TAB+caract(127)+RC
// intégration d'un fichier avec TAB en séparateur de colonnes et RC pour saut
de lignes
//Ensuite utiliser l'option hImpSansDélimiteur :
hImporteTexte(fichier,rubriques,cSep,hImpSansDélimiteur)

```

Ainsi les colonnes seront correctement interprétées

2. Fichier externe : lire le contenu hexadécimal

La lecture de la chaîne hexa d'un fichier binaire peut être réalisée par la fonction flit() et moyennant un petit subterfuge dans la lecture des caractères. en effet flit() est conçue nativement pour une lecture de fichier texte et propose directement une conversion en code ascii des caractères lus.

L'idée est alors de d'utiliser la résolution automatique des types et un entier non signé sur 1 octet (0...255) facile à convertir en hexa.

Ensuite, on peut faire ce que l'on veut...
Bon dev.

```

cFichier est une chaîne // contenu hexa du fichier
// Ouverture du fichier
nIdfic est un entier = fOuvre(MonFichier, foLecture)
SI nIdfic=-1 ALORS
  Erreur('L'ouverture du fichier a échoué')
SINON
  // récupère le nombre d'octets à lire
  nbcar est un entier = fPositionne(nIdfic,0,fpFin)
  ncar est un entier
  fPositionne(nIdfic,0,fpDébut)

```

```

sCar est un entier sans signe sur 1 octet

fLit(nIdfic,1,&sCar)    // nouvelle syntaxe : par adresse
TANTQUE ncar<nbcars
    ncar++
    // traitement de l'octet
    // conversion de l'entier en valeur hexa
    cfichier+=NumériqueVersChaine(sCar,'02X')
    // octet suivant
    fLit(nIdfic,1,&sCar)
FIN
// Fermeture du fichier
fFerme(nIdfic)
FIN

```

3. Conversion Octet – Binaire

Après lecture d'un fichier en hexadécimal, voici la conversion des octets en binaire.

Bon dev.

```

nA,nB sont des entiers sans signe sur 1 octet
nA=64
nB=255
Info(Octet2Bin(nA),Octet2Bin(nB))

```

Donne 01000000 et 11111111
avec

```

PROCEDURE Octet2Bin(sCar est un entier sans signe sur 1 octet)
// conversion de l'octet en hexa
sHexa est une chaîne fixe sur 2 = NumériqueVersChaine(sCar,'02X')
// conversion de l'octet (hexa) en binaire
sres est une chaîne fixe sur 8
i est un entier
BOUCLE
    i++
    SELON sHexa[[i]]
        CAS 0:
            sres[[ (4*(i-1))+1 A (4*i) ]]='0000'
        CAS 1:
            sres[[ (4*(i-1))+1 A (4*i) ]]='0001'
        CAS 2:
            sres[[ (4*(i-1))+1 A (4*i) ]]='0010'
        CAS 3:
            sres[[ (4*(i-1))+1 A (4*i) ]]='0011'
        CAS 4:
            sres[[ (4*(i-1))+1 A (4*i) ]]='0100'
        CAS 5:
            sres[[ (4*(i-1))+1 A (4*i) ]]='0101'
        CAS 6:
            sres[[ (4*(i-1))+1 A (4*i) ]]='0110'
        CAS 7:
            sres[[ (4*(i-1))+1 A (4*i) ]]='0111'
        CAS 8:
            sres[[ (4*(i-1))+1 A (4*i) ]]='1000'
        CAS 9:
            sres[[ (4*(i-1))+1 A (4*i) ]]='1001'

        CAS 'A':

```

```

                sres[[(4*(i-1))+1 A (4*i)]]='1010'
CAS 'B':
                sres[[(4*(i-1))+1 A (4*i)]]='1011'
CAS 'C':
                sres[[(4*(i-1))+1 A (4*i)]]='1100'
CAS 'D':
                sres[[(4*(i-1))+1 A (4*i)]]='1101'
CAS 'E':
                sres[[(4*(i-1))+1 A (4*i)]]='1110'
CAS 'F':
                sres[[(4*(i-1))+1 A (4*i)]]='1111'

FIN
SI i=2 ALORS SORTIR
FIN
REVOYER sres

```

4. Imprimer une page de site Internet

Merci à Firetox pour cette soumission :

pour ceux qui comme moi on cherché un moment comment imprimer une page Web sans passer par un activeX.

L'intérêt étant d'utiliser en WD7.5 le iapercu(ihtml,MonFic.htm) et ensuite transférer ce fichier .

Mais un beau jour vouloir l'imprimer sans qu'aucune question soit posée etc.

voici le bout de code :

```

NomFic est une chaine = 'D:\TestRTF.HTM'
monIe est un objet ole dynamique
si monie = null alors monie = allouer un objet OLE
'InternetExplorer.application'
si monie <> NULL ALORS
monie>>visible=olefaux
// on charge le document Word
monie>>navigate(nomfic)//,olefaux,olevrai)
tantque monIe>>busy
// attente de la fin du chargement de la page
FIN
monIe>>ExecWB(6,2)
FIN

```

Penser à libérer l'objet dynamique après impression....

```
libérer MonIe
```

5. Lire un champ appartenant à une autre fenêtre

Question récurrente s'il en est... et la solution est pourtant si simple.

Prenons deux fenêtre (fen1 et fen2), la première contient un table dont je veux récupérer des valeurs dans du code de la seconde.

Les deux fenêtres doivent évidemment être ouvertes.

Pour utiliser des champs d'autres fenêtres actives il faut utiliser l'indirection (consulter l'aide sur le sujet, c'est une des notions puissantes de WinDev) :

```

// code dans FEN2
// syntaxe 1 : accès direct en utilisant le 'chemin' du champ
MaVal = fen1.table.colonne[indice]

```

```
// syntaxe 2 : on construit le nom du champ (son 'chemin') à partir d'une
// chaîne, les signes {} activent l'indirection
MaVal = {'fenl.table.colonne['+indice+'],'indChamp}
```

NB : la seconde syntaxe interprète la chaîne pour construire le nom du champ. On peut donc construire le nom d'un champs avec des variables... très utile pour économiser du code.

6. Appeler un traitement par une touche

Il est parfois nécessaire de ne rendre disponible des traitements que par une combinaison de touches.

Pour ce faire, la méthode la plus simple est de mettre le code dans un bouton, lequel est simplement positionné en dehors de l'espace visible de la fenêtre. Evidemment il ne faut pas omettre, dans la description du bouton d'affecter la lettre d'appel correspondante (les ALT et CTRL en combinaison sont utilisables).

La disponibilité de la touche de fonction est équivalente à celle du bouton. Attention donc au multifenêtrage et aux plans.

Pour les plans, si le bouton est affecté à un plan, la touche ne sera disponible que lorsque le plan sera actif...

7. Rebooter un projet en exécution

Comment faire pour que l'application se relance ?

En effet, les modifications de la configuration par un panneau de configuration intégré doivent être prise en compte par le relancement de l'exé (lecture de la configuration dans initialisation du projet).

Pour ce faire c'est très simple :

l'application doit être multi-instance (pas testé si non instance)

On lance une autre session via LanceAppli() en spécifiant le nom de l'exé courant disponible avec exeInfo(). Puis, on termine la session en cours avec FinProgramme().

Ainsi l'utilisateur se voit invité dans une autre session du même logiciel... avec les nouveaux paramètres de configuration pris en compte.

```
SI PAS LanceAppli(gProjetExe,exeActif,exeNonBloquant) ALORS
    Erreur(ErreurInfo(errMessage))
SINON
    FinProgramme()
FIN
```

8. Conversion Décimal – Hexadécimal

Conversions par Michel Fages

Decimal en hexa

```
hex = NumToString(dec, 'x')
```

Hexa en décimal

```
dec = Val(hex, 16)
```

9. Inactiver une touche clavier

Merci au post de Firetox qui permet de répondre à cette question récurrente :

voici donc un exemple de code pour inactiver la touche échap.
Le principe est simple, on intercepte l'évènement correspondant à cette touche et on transforme le résultat obtenu pour qu'il ne soit pas interprété par l'application.
Pour les autres touches, il suffit de trouver les valeurs Windows de l'évènement correspondant.
Notez la présence du fichier WinDev7x/Personnel/Externe/KeyConst.WL qui peut être inclus à votre projet pas la commande Externe 'KeyConst.WL' en début de projet.

Un évènement et une procédure :

```
// ouverture de la fenêtre
GLOBAL
WMKeydown est un entier long = 256
evenement('Annul_Echap', '*', '*', WMKeydown)

// procédure Annul_Echap
Procédure Annul_Echap()
si _eve.wparam = 27 alors
// ou si _eve.wparam = VK_ESCAPE alors
_eve.retour = 0
FIN
```

10. Supprimer une ligne dans un fichier texte

Il est impossible d'effectuer cette manipulation directement dans le fichier original. Reste que rien n'empêche de passer par un fichier temporaire.

Il suffit alors simplement de copier dans un fichier temporaire les parties à conserver du fichier d'origine et sauter les lignes à supprimer.

Faire suivre le procédé par un fRenomme() et le tour est joué.

11. MonGroupe..Etat ne contient pas la valeur prévue

Attention :

Il est possible que la propriété ..Etat d'un groupe de champs ne corresponde pas à la dernière valeur qui lui a été fixée.

En effet, si vous affectez indépendamment l'état de certains champs du groupe, la propriété du groupe peut se voir affectée sans que les autres champs se voient modifiées.

Cause :

Aide de la propriété ..Etat

Citation :***Etat d'un groupe de champs***

Connaître l'état d'un groupe de champs : l'état renvoyé correspond à l'état du premier champ du groupe.

La modification de l'état de ce premier champs (selon l'ordre des tabstops) affecte la propriété ..etat du groupe sans que l'état des autres champs soient modifiés.

Une solution consiste alors simplement à ajouter un champs invisible au groupe : CHPGP_ et ensuite remplacer

```
si <NomDuGroupe>..etat=Actif alors
```

par

```
si CHPGP_<NomDuGroupe>..etat=Actif alors
```

pour être sûr de tomber juste dans l'évaluation l'état du groupe.

12...Visible avec une fenêtre comportant des plans

La propriété..visible et la gestion des plans se recouvrent.

Ainsi monchamp..visible retourne faux même si le champs est visible quand le champs n'appartient pas au plan courant de la fenêtre.

La solution, pour les fenêtres qui comportent des plans

```
// le plan du champ devient le plan courant  
Mafenêtre..plan=Monchamp..plan  
// récupère la visibilité effective du champ  
info(Monchamp..visible) // retourne alors vrai
```

13.Colonnes de tables qui deviennent inexistantes en exécution

L'utilisation de ConstruitTableFichier() pour alimenter une table à partir du résultat d'une requête doit être accompagnée de précaution.

Extrait de l'aide :Citation :

*-Les colonnes de la table définies sous l'éditeur sont supprimées.
- Les modes de saisie définis sous l'éditeur pour la table sont conservés.*

...

Les colonnes créées portent les noms "_COL1", "_COL2", "_COL3", etc. Ces colonnes sont accessibles par programmation via l'opérateur d'indirection { }.

Ainsi, après cette fonction, il ne faut utiliser plus que l'indirection et _COL1... pour manipuler les colonnes de votre table. Ne plus utiliser les noms des colonnes définies avec l'éditeur de fenêtre.

14. Supprimer une sous chaîne dans une chaîne

La procédure suivante supprime la première occurrence d'une sous chaîne dans une autre :

```
PROCEDURE ChaîneSupprime(ChaîneInitiale,ChaîneASupprimer)
pos est un entier
len est un entier = Taille(ChaîneASupprimer)
c est une chaîne
pos = Position(ChaîneInitiale,ChaîneASupprimer)
SI pos<>0 ALORS
    c = ExtraireChaîne(ChaîneInitiale,1,ChaîneASupprimer)
    ChaîneInitiale = c + ExtraireChaîne(ChaîneInitiale,2,ChaîneASupprimer)
FIN
```

Exemple :

ChaîneSupprime("111222333","222") => "111333"

III. CATEGORIE: WINDEV CODES HF

1. Cacher et récupérer des enregistrements

Il peut s'avérer utile de 'cacher' des enregistrements pour effectuer un traitement et les récupérer ensuite.

- Remplir un tableau (dynamique) avec les n° des enregistrements à cacher.

```
ndim est un entier
tEnr est un tableau de 0 entiers sur 8 octets
ndim=Dimension(tEnr,Dimension(rEnr)+1)
tEnr[ndim]=HNumEnr(FICHIER)
```

- Utiliser hRaye() pour les enlever des parcours du traitement.

```
POUR i= 1 A ndim
    hRaye(FICHIER,tEnr[i],hIgnoreIntégrité)
FIN
```

- Lancer le traitement, les enregistrement rayés ne sont plus lisible
- Récupérer les enregistrements par lecture et modification.

```
POUR i= 1 A ndim
    HLit(FICHIER,tEnr[i])
    HModifie(FICHIER,tEnr[i],hIgnoreIntégrité)
FIN
```

Hop : les enregistrements sont à nouveau là : magique !

NB :

- il faut veiller à ne pas quitter le traitement avant récupération des enregistrement ou bien alors se prémunir d'un mécanisme de sauvegarde/ restauration du fichier.

- Il faut aussi veiller à se prémunir d'un réindexation avec compactage du fichier pendant la durée du traitement sous peine de perdre les informations.

A utiliser de préférence avec des fichier temporaires donc.

2. Afficher les composantes des clés composées

Les clés composées sont désormais disponibles sous forme de chaînes via HconstruitValClé() qui simplifie grandement la gestion des clés composées.

Reste que les valeurs sont complétées par hValMin (ou hValMax).

hValMin est le caractère 0 de la table ascii, lequel correspond aussi en C/C++ (langage dans lequel est créé WinDev) au caractère de fin de chaîne.

Ainsi le code suivant

```
cCle est une chaine = HConstruiValCle(Fichier,Cle,val1,val2...)
info(cVal)
```

...ne retourne que la valeur de la première partie de la clé. Info interprète le premier caractère hValMin comme la fin de la chaîne.

Pour résoudre ce problème, il faut simplement remplacer les caractères hValMin par des caractères affichables (un espace par exemple), ce qui donne :

```
cCle est une chaine = HConstruiValCle(Fichier,Cle,val1,val2...)
info(remplace(cVal,hvalmin,' '))
```

3. SQL INSERT : Récupérer le dernier identifiant auto (HF & MySQL)

FAQ aimablement soumise par KPiTN :

Récupérer le dernier identifiant inséré

BDD Hyperfile :

```
ReqNav est une source de donnée
HExécuteRequêteSQL(ReqNav,'Insert Into Navires (LibNavire) Values (' + TxtNavire
+
')')
DernierIdentifiant=ReqNav.idnavire
```

BDD Mysql :

```
ReqNav est une source de donnée
HExécuteRequêteSQL(ReqNav,'ConnectSQL',hRequêteSansCorrection,'insert into
navires
(libnavire) values (' + TxtNavire+ '')')
HExécuteRequêteSQL(ReqNav,'ConnectSQL',hRequêteSansCorrection,Minuscule('SELECT
LAST_INSERT_ID() as idnavire'))
HLitPremier(ReqNav)
DernierIdentifiant=ReqNav.idnavire
```

IV. CATEGORIE : WINDEV ETATS

1. Etat sur un tableau dynamique

Pour imprimer un Etat basé sur un tableau dynamique : 2 pièges simples à lever.

1/ La portée du tableau. Il faut que le tableau dynamique soit une globale de la fenêtre qui appelle l'état sinon la syntaxe 'NomFen.NomTableau' pose des problèmes.

2/ Le parcours de l'état.

La source de données de l'état doit être 'un fichier texte' et 'Je veux programmer la lecture du fichier'.

Enfin le code de votre état doit réaliser le parcours du tableau

```
// Initialisation de l'état
nIndice est un entier

// Après lecture
nIndice++
si nIndice>dimension(NomFenetre.Nomtableau) alors renvoyer annuler

// Avant impression du corps
// Affectation des champs calculés
Champ1=nomFen.nomtableau[indice,1]
...
```

En Résumé, la source de données d'un état détermine l'exécution des blocs de codes et notamment ceux relatifs aux parcours. Cette configuration fichier texte et parcours programmé est la seule ou l'on peut faire ce que bon nous semble.

2. Booléens, de belles coches et plus de carrés vides

WD7.x propose en standard de remplacer les booléens dans les états par des coches. C'est une excellente chose. Un détail, quand le booléen est à faux, un carré vide remplace le carré coché. Certain pourront ne pas apprécier la nuance.

La solution est fort simple, utiliser un champ calculé texte avec la police winding (détail>style>police) et afficher soit rien soit le caractère de coche (254 en police windings) :

```
// Code avant impression du champ
MoiMême=' '
SI FICHER.RUBRIQUE ALORS MoiMême=Caract(254)
```

3. Appliquer les modification iParametre() pour changer de bac

Modifier les paramètres d'une imprimante par programmation WinDev est possible. Reste qu'il faut faire prendre en compte ces modifications à l'imprimante...

Voici la marche à suivre :

Lancer iParamètre

Lancer une impression suivie d'un iRAZ

Exemple :

```
CONSTANT
IMP_BAC = 'HAUT'
IMP_DEFAULT = 'BAS'
FIN

// Procédure de changement de bac
Procédure ChangeBac(pcBac=IMP_HAUT)
iparametre('ALIMENTATIONPAPIER='+pcBac)
iparametre('DEFAULT=OUI') // paramètre valable que pour l'impression en
cours
// Impression pour forcer les paramètres
iimprime('COUCOU')
// Annulation de l'impression
iRAZ
Fin

// exemple d'utilisation
changebac(IMP_BAS)
iImprimeEtat(...)
ChangeBac()
```

V. CATEGORIE : FENETRES

1. Onglets, ordre de navigation et touche entrée

Dans une fenêtre de type fiche où tous les champs ne sont pas répartis dans les onglets, la navigation (au clavier par tab ou entrée) entre les champs qui ne sont pas dans l'onglet et ceux qui y sont pose des problèmes avec la navigation via la touche entrée.

La navigation via la touche entrée est obtenue en n'ayant aucun bouton de type validation dans la fenêtre.

Si on positionne les champs hors onglet avant ceux de l'onglet, la navigation dans l'onglet n'est pas obtenue. On passe directement de l'onglet aux boutons (derniers dans l'ordre de navigation).

La solution est très simple :

quand on arrive dans l'onglet on doit simuler l'appui sur la touche tab qui va passer le curseur au premier champ de l'onglet !

Ajouter le code suivant dans le code de prise de focus de l'onglet :

```
EnvoieTouche('{TAB}', Handle(MoiMême..Nom))
```

2. Répertoires dans un Treeview

Astuce de la part de Philippe RAMIN, Amélioration de l'idée de Laurent LOISELEUX. Code original sur le site de l'asso.

```
//Déclarations globales :
gEnroule est un entier
gImage1 est une chaîne=fRépertoireTemp()+'Image1.bmp' // Image d'une
disquette
gImage2 est une chaîne=fRépertoireTemp()+'Image2.bmp' // Image d'un disque
dur
gImage3 est une chaîne=fRépertoireTemp()+'Image3.bmp' // Image d'un CD
gImage4 est une chaîne=fRépertoireTemp()+'Image4.bmp' // Image du Répertoire
enroulé
gImage5 est une chaîne=fRépertoireTemp()+'Image5.bmp' // Image du Répertoire
déroulé

//Vous mettez ce que vous voulez dans les images moi j'ai créé 5 champs images
invisibles (Image1-->5) contenant le code init suivant :

Dc est un entier=dDébutDessin(MoiMême)
hIcon est un entier=0
API('SHELL32','ExtractIconExA','Shell32.dll',6,&hIcon,Null,1)
SI hIcon>1 ALORS
SI API('USER32','DrawIcon',Dc,1,1,hIcon)=0 ALORS
Erreur(ErreurInfo())
FIN
API('USER32','DestroyIcon',hIcon)
FIN
dSauveImageBMP(MoiMême,fRépertoireTemp()+MoiMême..Nom+'.bmp')

//( Dans l'ordre des images : Index=6, 8, 40, 3 et 4 (correspond à l'index de
l'icone dans le Shell32.dll) )

//Créer le treeview <MyLittleTree> bien sur...

//-----
//Initialisation du treeview avec la liste des disques

//Attention :
Si vous utilisez la même méthode que moi pour les images il faut s'assurer
mettre le code suivant dans l'initialisation de la fenêtre (question de DC),
sinon vous pouvez le mettre dans l'init du treeview avec des images en dur :

TheString est une chaîne=NetListeDisque(-1)
cCpt est un entier=1
OneString est une chaîne=ExtraitChaîne(TheString,cCpt,RC)
FichierTemp est une chaîne=''
BOUCLE
    SELON SansEspace(ExtraitChaîne(OneString,2,TAB))
    CAS 'FD':

ArbreAjoute(MyLittleTree,ExtraitChaîne(OneString,1,TAB)+':',gImage1,gImage1)
    CAS 'HD':

ArbreAjoute(MyLittleTree,ExtraitChaîne(OneString,1,TAB)+':',gImage2,gImage2)
    CAS 'CD':

ArbreAjoute(MyLittleTree,ExtraitChaîne(OneString,1,TAB)+':',gImage3,gImage3)
    FIN
    cCpt++
    OneString=ExtraitChaîne(TheString,cCpt,RC)
    SI OneString~='' ALORS
        SORTIR
    FIN
FIN
//-----
-----
```

```
//Code d'évènement du treeview

//Au double-clic:
SI gEnroule=aEnroule ALORS
MaSelection est une chaîne=ListeRep(ArbreSelect(MoiMême))
SI PAS MaSelection~='' ALORS
Info('Ma sélection : '+MaSelection)
FIN
FIN

//A l'enroulé/déroulé:
gEnroule=ArbreEtat(MyLittleTree,ArbreSelect(MoiMême))

//-----
//Procédure de LISTEREP utilisée par le double-clic du tree

PROCEDURE ListeRep(xRepertoire)
// Listage de répertoire

cRepertoire est une chaîne=Remplace(xRepertoire,TAB,'')
cString est une chaîne=NetListeRep(cRepertoire+'*.*')
cString=ExtraitChaîne(cString,1,RC)
SI cString~='' ALORS
RENVOYER cRepertoire
FIN
cCpt est un entier=1
cOneRep est une chaîne=ExtraitChaîne(cString,cCpt)
cDeroule est un booléen=Faux
TANTQUE cOneRep<>EOT
SI PAS cOneRep~='.' ET PAS cOneRep~='..' ALORS
ArbreAjoute(MyLittleTree,xRepertoire+TAB+cOneRep,gImage4,gImage5)
cDeroule=Vrai
FIN
cCpt++
cOneRep=ExtraitChaîne(cString,cCpt)
FIN
ArbreDéroule(MyLittleTree,xRepertoire)
SI cDeroule ALORS
RENVOYER ''
SINON
RENVOYER cRepertoire
FIN

//-----
//Destruction (fermeture de la fenêtre)

// En option : Le nettoyage peut être fait avant mais pas dans le même
traitement que dDébutDessin...
dFinDessin(Image1)
dFinDessin(Image2)
dFinDessin(Image3)
dFinDessin(Image4)
dFinDessin(Image5)

// En option : supprime les bmp dynamiques
fSupprime(gImage1)
fSupprime(gImage2)
fSupprime(gImage3)
```

```
fSupprime(gImage4)
fSupprime(gImage5)
```

3. Info / Erreur / OuiNon personnalisés

Posté à P. Burton sur ozdev.net et traduit ici bas, cette astuce ne vaut que pour wd7.5 :

Aller dans Projet\Description du projet/Onglet Style

Cocher la dernière case : 'Appliquer le thème de gabarit sur les fenêtres d'information système...'

Voilà.

Une fenêtre est créée 'WinDevMessageBox', lui appliquer le style que l'on veut. Mais **NE PAS MODIFIER** le reste (codes icones...).

Maintenant elle sera appelée à chaque appel de Info() ou Erreur().

Merci Peter

4. Supprimer un champ créé avec ChampClone()

ChampClone permet de créer une copie de champs existant dans une fenêtre.

L'inverse est possible par l'API

```
AppelDLL32('user32','DestroyWindow',handle(cNomChamp))
```

Limites :

- Pour les champs composés (tables, onglets...) cela ne marche pas ! Il faudrait disposer du handle du composant (colonne, volet) et tester.
- Impossible de réutiliser un champ cloné puis supprimé. Si l'API supprime le contrôle dans la fenêtre, le contexte fenêtre de WinDev n'est pas rafraîchi (le champ existe encore).

5. Calculer le CRC 16 utilisé dans le protocole XModem

Extrait d'un post de Kristian Paradis sur le forum de daussy.org :

J'avais demandé il y a peu de l'aide pour calculer le CRC 16 utilisé dans le protocole XModem.

J'ai finalement été capable d'adapter un algorithme de C++ que j'ai trouvé alors voila, si quelqu'un d'autre en a besoin.

(Ou si vous trouvez une faille, vous pouvez m'en faire part! ;))

```
Procédure mCalculerCRC(_Chaine)

//Renvoie une chaine sur 2 octets.
eBit est un entier
eOctet est un entier
eCaract est un entier sans signe
eCRC est un entier sans signe
cChaine est une chaine = _Chaine

pour eOctet = 1 a taille(cChaine)
```

```
eCaract = asc(cChaine[[eOctet]])*0x100
eCRC = ouexbinaire(eCRC,eCaract)
pour eBit = 0 a 7
    si etbinaire(eCRC,0x8000) <> 0 alors
        eCRC = eCRC * 2
        eCRC = ouexbinaire(eCRC,0x1021)
    sinon
        eCRC = eCRC * 2
    FIN
FIN

eCRC = etbinaire(eCrC,0xFFFF)

FIN

renvoyer caract(partieentiere(eCrC/256)) + caract(modulo(eCrC,256))
```

Kristian Paradis

6. Comment savoir si une fenêtre est Ouverte ou Fermée ?

Pour savoir si une fenêtre est 'Ouverte (en exécution)' ou 'Fermée', il suffit d'utiliser la fonction de gestion des fenêtres FenEtat.

Exemple :

```
// Connaître l'état de la fenêtre 'fen2200'
Etat_fen2200 = FenEtat(fen2200)
SI Etat_fen2200 = inexistant ALORS
    Traitement...
FIN
```

Les principales valeurs possibles renvoyées par FenEtat sont :

- Actif (la fenêtre est ouverte et possède le focus)
- AffichageSeulement (la fenêtre est ouverte mais ne possède pas le focus)
- Inexistant (la fenêtre n'existe pas ou n'est pas ouverte)
- Invisible (la fenêtre ne possède pas le focus et est invisible à l'écran)

7. Comment afficher un document PDF dans une fenêtre ?

Il suffit pour cela de créer une fenêtre contenant un champ HTML, auquel on associera le chemin du fichier PDF, plutôt qu'une URL.

Voici un morceau de code pour vous aiguiller :

```
// MonIE est le nom du champs HTML
MonIE..Valeur = <chemin complet du fichier PDF>
```

Voilà à l'ouverture de la fenêtre le fichier PDF est chargé dans la fenêtre. Attention toutefois, il est impératif que le client Acrobat Reader soit présent sur la machine qui devra afficher le fichier PDF.

Code fournit par DrCharly93

8. Comment donner le focus à un champ particulier ?

A l'ouverture d'une fenêtre, dans le but d'éviter un clic de souris, il peut être intéressant de donner le focus à un champ (de saisie ou autre) directement. Pour cela, il suffit d'utiliser le code suivant :

```
EcranPremier(<champQuiDoitPrendreLeFocus>)
```

9. Comment donner le focus à une ligne d'une table lors d'un clic droit ?

Il suffit de placer le code suivant dans l'évènement 'Clic sur bouton droit' de votre table :

```
nligne est un entier  
nCol est un entier  
ResInfo est une chaîne  
x est un entier  
y est un entier  
// sélection sur la table <NomDeVotreTable>  
x = PoidsFaible(CurseurPos())  
y = PoidsFort(CurseurPos())  
nligne = TableInfoXY(<NomDeVotreTable>, tiNumLigne + tiOrigineEcran, x, y)  
nCol = TableInfoXY(<NomDeVotreTable>, tiNumColonne + tiOrigineEcran, x, y)  
TableSelectPlus(<NomDeVotreTable>,nligne)
```

VI. CATEGORIE : WINDEV PROJET

1. Identificateur xxx inconnu ou inaccessible ici...

Lors de l'ouverture d'un projet ou de la recomposition du graphe, WinDev peut retourner le message suivant :

'Identificateur X inconnu ou inaccessible ici: pour accéder à un élément hors portée, utilisez une déclaration externe'

En fait cette erreur est due au fait qu'il a détecté un appel à un élément qui ne figure pas dans la liste des éléments du projet (fenêtre, classe...)

Ce peut être du au fait qu'un collaborateur a ajouté cet élément au projet dans le cas d'un développement en groupe.

Pour résoudre cela il faut noter le nom de l'élément X et l'ajouter au projet : Projet > liste des composants du projet > Ajouter

2. Icône personnalisée dans les barres de titre

Plutôt que de passer sur toutes vos fenêtres et dans le détail, indiquer votre icône de projet en remplacement le l'icône par défaut (le fameux WD), voici donc comment faire :

1- Lors du développement, laissez cet information vide.

2- Assistant compilation : après le nom de l'exécutable, indiquer l'icône de votre projet.

C'est tout, cet icône sert lors de la création des raccourcis mais aussi dans les barres de titre de vos fenêtres.

3. Connaître le répertoire des données en installation réseau

Généralement le répertoire d'exécution sert aussi de répertoire pour les données. Reste qu'en installation réseau, ce répertoire des données n'est pas celui de fRepEnCours(). Il faut donc retrouver ce répertoire par programmation.

Dans l'initialisation du projet, on va utiliser tout simplement le chemin d'un des fichiers de la base. Cette possibilité est disponible avant même le lancement de la création de fichiers physiques.

```
gcRepDonnées est une chaine=Monfichier..répertoire
...
hCreationSiInexistant('*')
...
```

gcRepDonnée contient alors le chemin du répertoire des données sur le serveur de fichier... La globale peut être utilisée en lieu et place de fRepEnCours() pour localiser les fichiers de données.

Noter que ce fonctionnement est aussi valide pour les projets installés normalement.

VII. CATEGORIE : WINDEV DIVERS

1. WinDev et les cartes SoundBlaster

Merci à KPiTN pour l'information.

Le driver pour la carte son Creative Sound Blaster live! 5.1 Digital fait planter absolument tous les programmes exécutables générés avec WinDev 7.5.

La cause du problème est un programme nommé CTHELPER.EXE
Il suffit de le supprimer de la base de registre.

```
Démarrer->exécuter -> regedit
Chercher dans :
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
Une clé appelé : CTHELPER
Supprimer la clé et redémarrer
```

2. Trouver le pilote ODBC Hyperfile

Pour trouver le driver ODBC sur les fichier HF

Version 7 :

<http://www.pcsoft.fr/st/telec/WinDev7/index.html>

et sélectionner la page de la dernière version.

Dans cette page aller en bas dans le paragraphe téléchargement :
"ODBC sur HF (optionnel)" et prendre le fichier

Version 5.5 :

contacter les services commerciaux PCSoft : pcsoft@pcsoft.fr

Précision : Les drivers ODBC HF sont en lecture seule !

VIII. CATEGORIE: WINDEV GABARITS/DESIGN

1. Magenta des BMP et boutons

L'affectation d'un BMP en image à un bouton en utilisant le parcours de fichiers ou en saisissant nom complet du fichier (description>Général>image), il arrive que le magenta (transparent) du bmp ne soit pas ignoré.

Solution :

Utiliser le bouton 'Catalogue' et sélectionner la collection 'Répertoire'.

Y sélectionner une des images de votre projet et valider.

Cette méthode permet de palier au problème de transparence du magenta pour les bmp.