

## Documentation des fonctions et objets de Vision Budget Script

Un fichier VBS doit contenir des déclarations globales ou des définitions de procédures ou fonctions. Eviter de placer du code directement dans un fichier VBS en dehors de toute procédure ou fonction.

Directives spéciales (# en premier caractère de la ligne) :

- #RESET : Purge tous les modules chargés
- #RUN=xxx : Lance la procédure xxx après chargement du code
- #TIMEOUT=xxxx : Fixe un timeout xxxx en ms (danger plante VBW)
- #INCLUDE=xxxx : Insère le fichier xxxx à la position courante
- #STOP : Le reste du fichier sera ignoré

Les procédures suivantes seront lancés automatiquement par VBW si elles existent :

- OnStart() : Au lancement de VBW
- OnStop() : A l'arrêt de VBW
- OnOpen() : A l'ouverture d'un compte
- OnClose() : A la fermeture d'un compte (il est alors possible de bloquer la fermeture avec un appel à la méthode « DontClose » )

Si le fichier AUTOLOAD.VBS existe dans le répertoire utilisateur il sera chargé au démarrage de VBW (il peut contenir la définition de la procédure « OnStart » ou de variables globales par exemple).

Vous pouvez utiliser une macro d'un script VBS comme script Minitel pour cela remplacer le répertoire du script Minitel par \* suivi du nom de la macro à lancer.

Il est conseillé de commencer tout fichier VBS par « option explicit » pour éviter d'utiliser des variables non déclarées.

Les pages suivantes décrivent les objets utilisables et leur méthodes et propriétés.

Les mots en **gras** représentent des noms d'objets, de constantes, de méthodes ou de propriétés.

Les mots en *italique* représentent des types :

*Integer* = nombre entier

*Float* = nombre à virgule

*String* = chaîne de caractères

*Bool* = booléen (valeurs possibles = **True** ou **False**)

Les paramètres entre [crochets] sont facultatifs

---

*Note* : Le but de cette documentation n'est pas de fournir des informations sur le langage Visual Basic mais sur les objets de Vision Budget utilisables.

Pour un aperçu sur Visual Basic Script consultez le fichier **VBS.PDF**

Pour plus de renseignements sur Visual Basic Script consultez le site Microsoft :

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vtoriVBScript.asp>

---

**Objet : VisionBudget.Application**

Remarque : Cet objet est affecté par défaut à la variable globale **VBW**.

Méthodes :

**Print**(string Mess) ou **PrintLine**(string Mess)

Affiche un message dans la fenêtre résultat

**DrawLine**

Trace une ligne horizontale dans la fenêtre résultat

**ClearPrintOut**

Efface la dernière fenêtre résultat ouverte

**DiscardPrintOut** ([bool Close])

Détache la fenêtre résultat (si Close est **True** la fenêtre sera fermée)

**GraphClear**([bool NewWindow], [string Title])

Efface la fenêtre graphique ou ouvre une nouvelle vide si NewWindow présent et True

**SetGraphZone**(integer X, integer Y, bool SameRatio)

Indique la taille en pixels pour les axes X et Y, ou 0 si pas d'ajustement automatique de l'échelle

Si SameRatio est **True** alors l'échelle sera la même pour X et Y (la plus petite des deux)

**GraphDraw**(integer DrawType, [integer X1], [integer Y1], [integer X2], [integer Y2], [string S])

Ajoute un dessin de type DrawType. Les types possibles et leurs actions sont les suivants :

- **vbwNone** : ignorer mais provoquer un update de l'affichage si **GraphAutoUpdate** = True
- **vbwLine** : tracer ligne entre X1,Y1 et X2,Y2 avec la plume
- **vbwRect** : tracer rectangle entre X1,Y1,X2,Y2 avec la plume
- **vbwFill** : remplir entre X1,Y1,X2,Y2 avec la brosse
- **vbwFillRect** : tracer rectangle plein entre X1,Y1,X2,Y2 avec la brosse et la plume
- **vbwEllipse** : tracer ellipse dans rectangle X1,Y1,X2,Y2 avec la brosse et la plume
- **vbwSetPie** : définir le rectangle X1,Y1,X2,Y2 pour les camemberts suivants
- **vbwSetPieSqr** : définir le carré X1,Y1,X2 pour les camemberts suivants
- **vbwPie** : Tracer un camembert de X1,Y1 à X2,Y2 avec la plume et la brosse  
(Attention : Utilise le sens trigonométrique = inverse des aiguilles d'une montre)
- **vbwPieP** : Tracer un camembert de X1% à Y1% avec la plume et la brosse (S en option)
- **vbwText** : Afficher le texte S en X1,Y1
- **vbwTextZone** : Afficher le texte S dans la zone X1,Y1,X2,Y2 - voir **vbwColor** pour Style
- **vbwFont** : Sélectionner la police S avec la taille X1, la largeur Y1 et les attributs Y2  
Si X2=**vbwNoScale** alors X1 et Y1 sont en pixel quelle que soit l'échelle en cours  
Y2 peut être 0 ou la somme de vbwItalic, vbwUnderline, vbwBold (ou vbwThin)  
ou vbwSystemFont ou vbwFixedSystemFont (X1 et Y1 alors ignorés)  
Si Y1=0 alors la largeur est automatique en fonction de la taille X1

- **vbwBrush** : sélectionner la brosse de couleur R=X1, V=Y1, B=X2 et de type Y2

**vbwBrushNull** (brosse invisible sauf pour **vbwFill**)

**vbwBrushSolid**



**vbwBrushHorizontal**



**vbwBrushVertical**



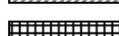
**vbwBrushFDiagonal**



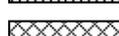
**vbwBrushBDiagonal**



**vbwBrushCross**



**vbwBrushDiagcross**



- **vbwBkColor** : Sélectionne la couleur R=X1, V=Y1, B=X2 pour le fond
- **vbwColor** : Sélectionne la couleur R=X1, V=Y1, B=X2 pour la plume  
Y2=0 ou une valeur somme de styles pour zone de texte (**vbwTextZone**)  
Position H : **vbwZLeft** ou **vbwZCenter** ou **vbwZRight**  
Position V : **vbwZVTop** ou **vbwZVCenter** ou **vbwVBottom**  
(Affichage sur une seule ligne si vbwZVxxx utilisés)  
**vbwZAutoRect** : Largeur et Hauteur de la zone automatique  
**vbwZAutoLine** : Saut de ligne automatique
- **vbwPen** : Sélectionner la plume de taille X1, et de type Y1  
Si X2=**vbwNoScale** alors X1 est en pixel quelle que soit l'échelle en cours  
X1 doit être à 0 ou 1 si Y1 est différent de **vbwPenSolid**  
Y1 = une des valeurs suivantes
  - vbwPenNull** (plume invisible)
  - vbwPenSolid**                    \_\_\_\_\_
  - vbwPenDash**                    - - - - -
  - vbwPenDot**                    .....
  - vbwPenDashDot**                - . - . - .
  - vbwPenDashDotDot**            - . - . - . - .

**AddMenu**(*string* Item, *string* Sub)

Insère le menu de nom Item pour lancer la procédure de nom Sub

La procédure « HELP\_Sub » sera lancée si l'utilisateur demande de l'aide sur le menu

**DeleteMenu**([*string* Item])

Efface le menu de nom Item ou tous les menus si Item absent ou vide

**Quit**()

Quitte sans condition et sans confirmation l'application (dangereux – en cas d'urgence seulement)

**Exit**()

Quitte normalement l'application sauf si refus de l'utilisateur

[*Cpt*] = **OpenCpt**(*string* Key)

Ouvre le compte de clé Key et retourne un objet de type *Cpt*

Il est nécessaire de faire un appel à la méthode **Close()** du *Cpt* après usage pour fermer le compte

[*UsrCpt*] = **NewUsrCpt**()

Ouvre un compte fictif pour afficher des données utilisateurs et retourne un objet de type *UsrCpt*

Il est nécessaire de faire un appel à la méthode **Close()** du *UsrCpt* après usage

[*integer*] = **ItemIndex**(*string* Key)

Retourne l'index (n° d'ordre) du compte de clé Key (nom abrégé)

[*CptEntry*] = **Item**(*integer* i)

Retourne un objet de type *CptEntry* correspondant au compte ayant l'index n°i

[*integer*] = **Count**()

Retourne le nombre de comptes dans VBW

[*CptEntry*] = {Iterator pour For Each}

Retourne chaque *CptEntry* dans un For Each - Utiliser par exemple « for each X in VBW »

## **ClearList()**

Vide la ListBox

## **AddListItem**(*string* Item, [*integer* Data])

Ajoute un élément Item dans la ListBox avec éventuellement la donnée Data associée

## [*integer*] = **ListShow**(*string* Title)

Affiche la ListBox avec le titre Title et retourne le n° de choix (index) ou -1 si annulé

Attention, la listbox est trié après affichage, utilisez **GetListData** sur l'index pour retrouver l'item

## [*string*] = **GetListItem** (*integer* index)

Retourne l'Item de la liste en position index

## [*integer*] = **GetListData** (*integer* index)

Retourne la Data associée à l'Item de la liste en position index (utile pour retrouver un item)

## **ListSelect** (*integer* index)

Sélectionne un item par défaut dans la ListBox (0 = premier item inséré)

## **Sleep** (*integer* ms)

Interruption du script pendant ms millisecondes

## [*string*] = **FormatNumber** (*float* Nombre, [*integer* Decimale])

Retourne une chaîne de caractère avec le nombre formaté (par défaut avec 2 décimales)

## [*float*] = **EvalNumber** (*string* Texte)

Retourne la valeur d'une chaîne de caractère formatée

## [*bool*] = **OpenReadFile** (*integer* index, *string* NomFichier)

Ouvre le fichier index (0-9) de chemin NomFichier en lecture et retourne **True** si Ok

## [*bool*] = **OpenWriteFile** (*integer* index, *string* NomFichier, [*bool* Append])

Ouvre le fichier index (0-9) de chemin NomFichier en écriture et retourne **True** si Ok

Si Append présent et **True** alors fichier ouvert en ajout, sinon fichier existant écrasé.

## [*bool*] = **WriteFile** (*integer* index, *string* Data)

Ecrit Data dans le fichier index (0-9) (doit être ouvert en écriture) et retourne **True** si Ok

## [*string*] = **ReadFile** (*integer* index)

Retourne ligne lu dans le fichier index (0-9) (doit être ouvert en lecture) et retourne vide si erreur.

## [*bool*] = **EndOfFile** (*integer* index)

Retourne **True** si on est à la fin du fichier index (0-9) (doit être ouvert en lecture).

## **CloseFile** (*integer* index)

Ferme le fichier index (0-9) (doit être ouvert en lecture ou écriture)

## **DeleteFile** (*string* NomFichier)

Efface le fichier de chemin NomFichier

## **RenameFile** (*string* Ancien, *string* Nouveau)

Renomme/déplace le fichier de chemin Ancien en Nouveau

## [*bool*] = **ChangeDir** ([*string* NomRep])

Change de répertoire vers NomRep ou vers le répertoire des fichiers comptes si absent ou vide

Si NomRep est égal à « \* » alors le répertoire programme de VBW.EXE sera utilisé

## [*string*] = **FileSelect** (*integer* mode, *string* Desc, *string* Ext)

Permet de sélectionner un fichier. Retourne le chemin complet du fichier ou vide si annulé

Desc représente la description du type de fichier suivi de | et d'un pattern (\*.ext)

Il peut y avoir plusieurs types séparés par des |

Exemple : « Fichier texte (\*.TXT)|\*.TXT|Tous (\*.\*)|\*. \* »

Ext est l'extension (ex : Txt ) à utiliser par défaut (si l'utilisateur ne précise pas d'extension)

Mode est la somme des valeurs suivantes :

**vbwLoad** ou **vbwSave** : Indique si boîte d'ouverture ou de sauvegarde

**vbwMustExist** : Indique que le fichier ouvert doit exister

**vbwCreatePrompt** : Demande une confirmation de création si le fichier n'existe pas

Propriétés :

[*string*] **TitleStr**

Titre de la fenêtre résultat attachée ou de la prochaine fenêtre résultat ouverte

[*integer*] **PrintEffect**

Somme des types d'affichage des messages : **vbwNormal**, **vbwDouble**, **vbwCenter**  
Utilisé lors des affichages dans la vue résultats (avec Print)

[*string*] **Clipboard**

Contient le contenu du presse papier en texte (après un copier dans une application)

[*bool*] **GraphAutoUpdate**

Si **True** alors un appel à **GraphDraw()** provoque un update de la vue associée (lent).

[*Cpt*] **LastDoc**

Si la fenetre active est un compte alors retourne un objet de type **Cpt** vers ce compte

Il n'est pas nécessaire de faire un appel à **Close()** lorsque vous avez terminé avec **LastDoc**

[*UsrCpt*] **LastUsrDoc**

Si la fenetre active est un compte utilisateur alors retourne un objet de type **UsrCpt** vers ce compte

Il n'est pas nécessaire de faire un appel à **Close()** lorsque vous avez terminé avec **LastUsrDoc**

---

## Objet : Cpt

Cette objet représente un compte de n'importe quel type. Utilisez **Entry.Type** pour connaître le type du compte qui peut conditionner les méthodes utilisables. Toujours fermer avec **Close** un **Cpt** ouvert à l'aide de la méthode **VBW.OpenCpt**.

### Méthodes :

#### **Close**([*bool* AllView])

Ferme le compte si pas d'autre vue ouverte vers ce compte.

Le compte est fermé même si la propriétés Busy est **True**.

Si AllView est **True** ferme toutes les vues ouvertes de ce compte et ferme le compte ensuite

#### **DontClose**()

Lors d'un appel automatique de OnClose utilisez DontClose pour refuser de fermer le compte.

#### **ShowView**([*integer* Type])

Affiche une nouvelle vue de Type : **vbwViewNormal**, **vbwViewInfos**, **vbwViewGraph**  
**VbwViewBudget** ou **vbwViewPrev**

#### [*EcrInf*] = **GetInfo**()

Retourne une écriture de type **EcrInf** avec les informations d'un compte écritures

#### [*bool*] = **ReadEcrAt** (*integer* Key)

Lit l'écriture située à l'index Key et retourne **True** si Ok. Ecriture copiée dans **Ecr**.

#### [*bool*] = **WriteEcr** ()

Enregistre l'écriture stockée dans **Ecr** et retourne **True** si Ok. **Ecr.Key** doit être valide

Toujours utiliser **CreateEcr** avant d'utiliser **WriteEcr** pour un nouveau enregistrement

L'objet **Ecr** peut être modifié si Vision Budget détecte une erreur de liaison.

#### **CreateEcr** ()

Crée une nouvelle écriture vide. La copie de l'écriture sera stockée dans **Ecr**.

**Ecr.Key** contiendra alors une clé valide correspondant à cette nouvelle écriture.

#### [*bool*] = **DeleteEcr**(*integer* Key, [*bool* Confirm])

Supprime l'écriture avec confirmation sauf si Confirm présent et **False**

Retourne **True** si suppression réalisée

#### [*integer*] = **NewQuery**(*string* DateDeb, *string* DateFin, [*string* Filter])

Ouvre un query entre DateDeb et DateFin et retourne nombre de Ecr disponibles

Si DateDeb ou DateFin vide ou 00/00/0000 alors la date sera ignorée

Filter peut contenir les mots clés suivants séparés par des « , » :

NoPrev, NoSubEcr, OnlyCB, OnlyCB[xxx], Cat[xxx] avec xxx = nom de Cat ou carte

#### [*bool*] = **ReadNextEcr**()

Lit l'écriture suivante d'un query et retourne si Ok (**false** si pas de query en cours ou si fini)

[**BudgetEntry**] = **GetBudgetEntry**(*string* Key)

Retourne un objet **BudgetEntry** correspondant à la catégorie Key

**EvaluateBudget** (*integer* Annee, [*string* Date])

Evalue le budget pour le compte et pour l'année spécifiée.

Date représente la date charnière entre YTD et ETC (date du jour si vide)

[**EcrBudget**] = **GetBudgetEcr**([*integer* Key])

Retourne une écriture de type **EcrBudget** vide si pas de Key ou Key = -1

Retourne une écriture de type **EcrBudget** avec l'écriture de clé Key sinon

[*bool*] = **SetBudgetEcr**(**EcrBudget** Ecr)

Modifie l'écriture Ecr du budget dans la base ou crée une nouvelle si Ecr.Key=-1

Retourne **True** si Ok ou **False** si erreur.

[*bool*] = **DeleteBudgetEcr**(*integer* Key)

Supprime l'écriture budget dans la base

Retourne **True** si Ok ou **False** si erreur.

[*string*] = **SelectCat**(*string* Titre)

Affiche une boîte de choix d'une catégorie et retourne la catégorie sélectionnée

**AjustBud**(*bool* decr, *string* cat, *float* Mont)

Ajuste le budget Cat du compte de Mont (décrémente si decr est **True** sinon Incrémente)

Mont doit en principe avoir le même signe que le budget :

Décrémenter un budget -1000 de -100 donne un budget de -900

Décrémenter un budget 1000 de 100 donne un budget de 900

Incrémenter un budget -1000 de -100 donne un budget de -1100

Incrémenter un budget 1000 de 100 donne un budget de 1100

[*bool*] = **EditEcrDialog**(*integer* Key)

Lance la boîte de dialogue d'édition d'une écriture pour l'écriture Key ou une nouvelle si -1

Utilise l'objet **EcrDialog** pour définir le comportement de la boîte de dialogue et le retour

Retourne **True** si boîte validée

**ImportCatFrom**(*string* CptKey)

Importe l'ensemble des catégories définies dans le compte CptKey

[*bool*] = **EnterReleve**(*string* Date, *float* Mont, *bool* CardMode, [*string* CardName])

Active si possible le mode relevé pour la date et le montant indiqué

Si CardMode est True alors le mode carte est actif pour la carte CardName si présent

En mode carte, le paramètre Date peut être 00/00/0000 pour une date automatique

[*bool*] = **StopReleve**(*bool* Cancel)

Termine le relevé en cours et valide si possible sauf si Cancel est **True**

**ChangeReleveTarget**(*string* Date, [*float* Mont, *bool* ChgMont])

Change la date cible (si non vide) et le montant cible si ChgMont présent et **True**.

[*bool*] = **SelectReleveEcr**(integer Key, bool Check, [*bool* Ask])

Change l'écriture Key en mode relevé ou non (selon Check). Ask = **True** par défaut

Si plusieurs appels successifs, il est possible de positionner Ask à **False** sur les autres appels

Pour que la même date valeur soit demandée pour l'ensemble des écritures (si besoin)

[*bool*] = **IsCheckedEcr**(integer Key)

Retourne **True** si l'écriture Key est en mode relevé

[*float*] = **GetSolde**(integer Key)

Retourne le montant d'une écriture ou le solde du compte si Key est égal à

VbwSoldeToday : Solde à ce jour

vbwSoldeDR : Solde à la date du dernier relevé

vbwSoldeEnd : Solde Normal en fin d'année

vbwSoldeEndPrv : Solde prévisionnel en fin d'année

vbwSoldeNDR : Spécial en mode relevé pour nouveau solde relevé

vbwSoldeBalance : Pour un compte action = Balance des mouvements

vbwSoldeValue : Pour un compte action = Montant du portefeuille à ce jour

vbwSoldePMVR : Pour un compte action = plus ou moins values déjà réalisées

vbwSoldePMV : Pour un compte action = plus ou moins values totales

#### Propriétés :

[*bool*] **Busy** : Si **True** alors le compte ne pourra pas être fermé par l'utilisateur (**False** par défaut)

Attention car tant que = **True** il sera impossible de quitter VBW sauf avec VBW.Quit()

[*Ecr*] **Ecr** : Objet de type *Ecr* qui contient l'écriture en cours

[*CptEntry*] **Entry** : Objet de type *CptEntry* qui contient les informations sur le compte

[*Budget*] **Budget** : Objet de type *Budget* qui contient les catégories/budgets du compte

[*EcrDialog*] **EcrDialog** : Objet de type *EcrDialog* utilisé par **EditEcrDialog**

[*string*] **SubOnReleveChg** : Procédure à lancer lors de la modification d'une écriture en relevé

[*integer*] **ClickedEcr** : Clé de l'écriture concernée lors de l'appel de **SubOnReleveChg**.

[*integer*] **Selected** : Position de la ligne actuellement sélectionnée

[*integer*] **SelectionSize** : Nombre de lignes sélectionnées en + ou – par rapport à **Selected**

---

---

## Objet : EcrInf

Cet objet est retourné par l'objet **Cpt** (méthode **GetInfo**) uniquement si l'objet **Cpt** est de type écritures.

### Propriétés :

- [*string*] **DR** : Date du dernier relevé
- [*string*] **Titulaire** : Titulaire du compte
- [*string*] **NomBanque** : Nom de la banque
- [*string*] **CodeEtablissement** : Code de l'établissement de la banque (RIB)
- [*string*] **CodeGuichet** : Code du guichet (RIB)
- [*string*] **NumCpt** : Numéro du compte (RIB)
- [*string*] **CleRib** : Clé RIB
- [*string*] **AdrTitulaire** : Adresse du titulaire
- [*string*] **AdrBanque** : Adresse de la banque
- [*string*] **LieuPaiement** : Lieu de paiement
- [*string*] **Monnaie** : Monnaie du compte (sur un seul caractère)
- [*float*] **Taux** : Taux des intérêts si le compte a des intérêts
- [*bool*] **Interets** : **True** si le compte a des intérêts
- [*bool*] **NoDebit** : **True** si le compte ne permet pas les débits
- [*bool*] **NoDecBud** : **True** si l'utilisateur ne souhaite pas décrémenter le budget par défaut
- [*float*] **SoldeMin** : Solde Minimum sur ce compte avant erreur

---

## Objet : Budget

Cet objet est retourné uniquement par la propriété **Budget** d'un objet **Cpt**. Il sert à parcourir la liste des catégories ou à éditer les catégories.

### Méthodes :

#### **Delete**(string Key)

Supprime le budget de catégorie Key

[bool] = **Set**(string Key, string LongName)

Edite la catégorie Key et retourne **True** si une nouvelle est créée

[bool] = **Add**(string Key, string LongName)

Ajoute la catégorie Key et retourne **True** si n'existe pas déjà

[bool] = **ChangeMaster**(string Key, string Master)

Affecte à la catégorie Key la catégorie principale Master et retourne **True** si possible

**Show**(string Key, bool Visible)

Change l'état de visibilité de la catégorie Key et retourne **True** si Ok

[**BudgetEntry**] = **Item**(integer i)

Retourne un objet de type **BudgetEntry** correspondant à la catégorie ayant l'index n°i

[integer] = **Count**()

Retourne le nombre de catégories dans le budget

[**BudgetEntry**] = {Iterator pour For Each}

Retourne chaque **BudgetEntry** dans un For Each

Utiliser par exemple « for each X in Cpt.Budget »

---

## Objet : CptEntry

Cet objet est retourné par l'objet **VBW** (méthodes **Item** ou **Iterator**) ou par la propriété **Entry** d'un objet **Cpt**.

### Propriétés :

[string] **Key** : Clé du compte (vide si UsrCpt)

[string] **LongName** : Nom complet du compte (vide si UsrCpt)

[string] **Type** : Type de compte parmi les suivants :

« Ecritures » = **vbwTypeEcr** : Si le compte contient des écritures

« Compte Titre » = **vbwTypeAct** : Si le compte contient des actions

« Compte externe » = **vbwTypeExt** : Si le compte ne contient rien

« Utilisateur » = **vbwTypeUsr** : Si le compte est de type CptUsr

[string] **Path** : Chemin d'accès du répertoire du compte (vide si UsrCpt)

[bool] **IsOnMinitel** : Si **True** alors le compte est configuré pour les scripts Minitel

---

## Objet : BudgetEntry

Cet objet est retourné par l'objet **Cpt** (méthode **GetBudgetEntry**) ou l'objet **Budget** (méthodes **Item** ou **Iterator**).

### Méthodes :

[float] = **Month**(integer m)

Si budget évalué, retourne le montant du mois m (1 à 12)

### Propriétés :

[string] **Key** : Clé de la catégorie (vide si erreur)

[string] **LongName** : Nom complet de la catégorie

[bool] **HasSubCat** : **True** si cette catégorie a des sous catégories

[string] **MasterCat** : Clé de la catégorie principale associée ou vide si pas de catégorie principale

Note : **MasterCat** sera forcément vide si **HasSubCat** est **True**

[bool] **IsVisible** : **True** si catégorie visible dans les états

[bool] **IsEval** : **True** si budget évalué pour cette catégorie

[float] **YTD** : Si budget évalué, montant du début de l'année à ce jour

[float] **ETC** : Si budget évalué, montant de ce jour à la fin de l'année

[float] **Budget** : Si budget évalué, montant au budget

[integer] **FirstEcrKey** : Clé de la première écriture dans le budget ou -1 si pas d'écriture

---

## Objet : EcrBudget

Cet objet est utilisé par l'objet **Cpt** (retourné par la méthode **GetBudgetEcr** et utilisé par la méthode **SetBudgetEcr**).

### Méthodes :

[float] = **GetMonth**(integer m)

Retourne le montant du mois m (1 à 12)

**SetMonth**(integer m, float Mont)

Modifie le montant du mois m (1 à 12)

### Propriétés :

[integer] **Key** : Clé de l'écriture (-1 si erreur lors d'un **GetBudgetEcr**)

[string] **Cat** : Clé de la catégorie associée à l'écriture

Note : Si cette valeur est modifiée pour une écriture existante, **SetBudgetEcr** retournera **False**

[string] **Lib** : Libellé à utiliser pour cette écriture

[integer] **Day** : Jour de l'opération pour cette écriture ou -1 si l'écriture est un budget

[bool] **IsBudget** : **True** si cette écriture est un budget et **False** si c'est une prévision

[string] **LnkCpt** : Clé du compte lié si l'écriture est un virement vide sinon

[integer] **NextEcrKey** : Clé de l'écriture suivante dans le budget ou -1 si plus d'écriture

Note : Si cette valeur est modifiée pour une écriture existante, **SetBudgetEcr** retournera **False**

---

## Objet : Ecr

### Méthodes :

**Clear()** : Vide l'écriture

**[Ecr] = GetClone()** : Renvoi une copie de l'écriture

### Propriétés :

**[string] Text** : Libellé de l'écriture

**[integer] Key** : Clé de l'écriture (position unique dans la base de l'écriture lue ou à modifier)

**[string] DateVal** : Date valeur de l'écriture

**[string] DateOpe** : Date opération de l'écriture

**[integer] TypeOpe** : Type d'opération parmi les suivants :

**vbwOpeNor** : Type Normal

**vbwOpeCbC** : Type Carte ou chèque CB/CHQ

**vbwOpeVir** : Virement (+/-)

**vbwOpeSec** : Sous écriture

**vbwOpeInt** : Intérêts

**vbwOpePrm** : Prime

**vbwOpeSim** : Ecriture simulée ou temporaire

*Une opération dans ce mode sera supprimée automatiquement à l'ouverture du compte.*

**vbwOpeSol** : Solde d'un autre compte ou lié au DDE

**vbwOpeEqu** : Opération équivalente à celle d'un autre compte (écriture liée)

**[integer] ModeOpe** : Mode de l'opération parmi les suivants :

**vbwModeNor** : Opération en mode normal

**vbwModeRel** : Opération relevée (rapprochée)

**vbwModeSup** : Opération supposée relevée (au cours d'un rapprochement)

*Une opération dans ce mode sera basculée en vbwModeNor à l'ouverture du compte.*

**vbwModePrv** : Opération en mode Prévision

**vbwModeBud** : Opération en mode Budget

**[string] NameOpe** : Nom de l'opération (correspondant au type)

**[string] Cat** : Catégorie de l'écriture

**[float] Qte** : Quantité d'action de l'écriture (pour un compte titres uniquement)

**[float] Value** : Valeur de l'écriture (positive si crédit et négative si débit)

**[integer] UserNum** : Numéro utilisateur associé à l'écriture (exemple n° de chèque)

**[integer] LinkedEcr** : Index (clé) de l'écriture liée à cette écriture

**[string] LinkedCpt** : Nom du compte (clé) de l'écriture liée à cette écriture ou Nom de carte/chq

**[bool] HasSubEcr** : **True** si l'écriture a des sous écritures

**[integer] LinkedSEB** : Index (clé) de l'écriture du budget liée à cette écriture ou de la sous écriture

Si HasSubEcr est **True**, cette propriété représente l'index de la sous écriture suivante

Si non, cette propriété contient une valeur positive égale à l'index de l'écriture budget liée

Ou une valeur négative si l'écriture a été modifiée et n'est donc plus liée au budget

Toutes les propriétés ne sont pas significatives en fonction du type de compte ouvert, un compte écriture exploite l'intégralité des propriétés sauf **Qte**.

Un compte titres n'utilise que les propriétés : **Key**, **DateVal**, **Value**, **ModeOpe** (= index libellé spécifique), **Qte**, **UserNum** (= code de l'action), **LinkedEcr**, **LinkedCpt** et en lecture seulement : **Text** (= nom de l'action), **NameOpe** (= libellé spécifique)

---

## Objet : EcrDialog

### Méthodes :

#### **Reset()**

Réinitialise les paramètres de la boîte de dialogue (à lancer en premier)

#### **CopyEcr(Ecr Source)**

Copie les informations de l'écriture source comme écriture d'initialisation

### Propriétés :

[*Ecr*] **Ecr** : Objet de type *Ecr* avec les valeurs de retour et éventuellement d'initialisation

[*float*] **Limit** : Indique un montant absolue maximum si Limit>0 ou Minimum si Limit<0

[*bool*] **Preset** : Indique si la boîte doit être pré-remplie avec Ecr et/ou Card

[*bool*] **Write** : Indique si les informations seront enregistrées dans la base de données en retour

[*integer*] **Button** : Bouton qui a été sélectionné pour fermer la boîte :

**vbwBOK** : Bouton Ok

**vbwBCancel** : Bouton Cancel

**vbwBVentil** : Bouton Ventililer

**vbwBVentilN** : Bouton Ventilation Suivante

**vbwBVentilP** : Bouton Ventilation Précédente

[*integer*] **AutoValid** : Bouton à sélectionner automatiquement à l'ouverture ou **vbwBCancel** si aucun

[*integer*] **MasterKey** : Clé de l'écriture principale si il y a eu ventilation sinon -1

[*bool*] **DisableCheck** : Désactive les vérifications avant fermeture de la boîte

[*bool*] **DecBud** : Indique si la case DecBud est cochée (normalement retour uniquement)

[*bool*] **DisableDecBud** : force la case DecBud grisée (utilise alors DecBud en entrée)

[*bool*] **EnableDecBud** : force la case DecBud non grisée (utilise alors DecBud en entrée)

[*bool*] **DisableRel** : Indique si la case Mode Relevé est désactivée

[*bool*] **DisableNor** : Indique si la case Mode Normal est désactivée

[*bool*] **DisablePrv** : Indique si la case Mode Prévision est désactivée

[*bool*] **DisableBud** : Indique si la case Mode Budget est désactivée

[*bool*] **DisableTypOpe** : Indique si la liste des types d'opérations est désactivée

[*bool*] **DisableLnkCpt** : Indique si la liste des comptes/cartes est désactivée

[*bool*] **DisableMont** : Indique si la saisie du montant est désactivée

[*bool*] **DisableCat** : Indique si la sélection de la catégorie est désactivée

[*bool*] **DisableLib** : Indique si la sélection du libellé est désactivée

[*bool*] **DisableVentil** : Indique si le bouton Ventilation est désactivé

[*bool*] **DisableVentilN** : Indique si le bouton Ventilation Suivante est désactivé

[*bool*] **DisableVentilP** : Indique si le bouton Ventilation Précédente est désactivé

[*bool*] **IsOk** : Indique si Ok pour valider la saisie (non implémenté)

[*string*] **ValidateMac** : Procédure à lancer pour valider la saisie (non implémenté)

---

## Objet : UsrCpt

Ce compte fictif permet d'afficher des données utilisateurs indépendantes de la base de données de Vision Budget. Une vue proche de la vue écriture permet de formater l'affichage sur une à dix colonnes et de récupérer des actions de l'utilisateur sur les enregistrements à l'aide de procédures.

Toujours fermer avec **Close** un objet **UsrCpt** ouvert avec **VBW.NewUsrCpt**.

### Méthodes :

#### **Close**([*bool* AllView])

Ferme le compte fictif si pas d'autre vue ouverte vers ce compte.

Le compte est fermé même si la propriété Busy est **True**.

Si AllView est **True** ferme toutes les vues ouvertes de ce compte et ferme le compte ensuite

#### **DontClose**()

Lors d'un appel automatique de OnClose utilisez DontClose pour refuser de fermer le compte.

#### **SetHeader**(*integer* Num, *integer* Size, [*string* Name], [*integer* Mode])

Indique les paramètres d'un entête de colonne pour la prochaine vue qui sera ouverte

Num : est la position de la colonne (à partir de 0)

Size : est la taille de la colonne

Name : est le nom de la colonne à afficher dans l'entête

Mode : est le mode parmi vbwColCenter, vbwColLeft, vbwColRight, vbwColSymbol

#### **Purge**()

Efface toute les données du compte

#### **SetMem**(*integer* NumMem, *string* Data)

Sauvegarde dans une des 10 mémoires (0 à 9) une chaîne de caractères

#### [*string*] **GetMem**(*integer* NumMem)

Retourne la chaîne de caractères stockée dans une des 10 mémoires (0 à 9)

#### **ShowView**()

Affiche une nouvelle vue avec les paramètres définis précédemment

#### **Sort**(*integer* Col, [*bool* IsDate])

Trie selon la colonne Col, si IsDate est **True** alors le contenu sera considéré comme une date

#### **NotifyChange**(*bool* Reload)

Indique aux vues ouvertes d'afficher à nouveau les données en provenance du compte fictif.

Si Reload est **True** alors la liste sera rechargée (utile si le nombre de ligne a changé).

#### [*integer*] **AddData**([*integer* Key])

Ajoute une donnée dans la liste avec le n° de clé Key et retourne sa position dans la liste

#### [*integer*] **GetDataKey**(*integer* Pos)

Retourne le n° de clé pour la donnée en position Pos dans la liste

#### [*bool*] **SetData**(*integer* Pos, *integer* Col, *string* Data)

Modifie le texte de la colonne Col de la donnée en position Pos dans la liste – retourne **True** si Ok

[*bool*] **SetColor**(*integer* Pos, *integer* Color)

Modifie la couleur d'affichage (0-4) de la donnée en position Pos dans la liste – **True** si Ok  
Couleurs : vbwUsrCIBlack, vbwUsrCIBlue, vbwUsrCIRed, vbwUsrCIGray, vbwUsrCIGreen

[*string*] **GetData**(*integer* Pos, *integer* Col)

Retourne le texte de la colonne Col de la donnée en position Pos dans la liste

[*integer*] **FindData**(*integer* Col, *string* Data, [*integer* Start])

Retourne la position où l'on trouve Data dans la colonne Col à partir de la position Start

**ViewSelect**(*integer* Pos, [*integer* Size])

Sélectionne la ligne Pos dans la vue sur Size éléments (positif ou négatif)

**SetInfo**(*integer* Pos, *string* Texte)

Modifie un texte dans la zone détails (5 positions de 0 à 4 disponibles dans la zone détails).

[*Cpt*] = **AttachCpt**(*string* Key)

S'il n'est pas encore ouvert, ouvre le compte de clé Key et retourne un objet de type **Cpt**  
Il est inutile de faire un appel à la méthode **Close()** du **Cpt** car cet appel sera alors ignoré  
L'objet **CptUsr** se chargera de fermer les comptes attachés automatiquement  
Si **Busy** est **True** alors tous les comptes attachés seront ouvert en mode **Busy=True**

[*bool*] = **IsAttached**(*string* Key)

Retourne **True** si le compte de clé Key est déjà attaché (et donc ouvert) à ce **CptUsr**

### Propriétés :

[*bool*] **Busy** : Si **True** alors le compte ne pourra pas être fermé par l'utilisateur (**False** par défaut)

[*integer*] **Count** : Nombre de lignes dans le compte utilisateur (ajoutées avec **AddData**)

[*bool*] **ShowDetails** : Si **True** alors la zone détails (infos) sera affichée

[*string*] **TitleStr** : Titre de la prochaine fenêtre qui sera ouverte

[*integer*] **AutoSizeHdr** : N° de la colonne dont la taille est automatique (expansion automatique)

[*bool*] **MultiSelect** : Si **True** alors la sélection de plusieurs lignes sera possible (voir **SelectionSize**)

[*string*] **SubOnClose** : Procédure à lancer lors de la fermeture du compte (utilisez **DontClose**)

[*string*] **SubOnDestroy** : Procédure à lancer lors de la fermeture définitive du compte

[*string*] **SubRefresh** : Procédure à lancer pour rafraîchir les données de la fenêtre

[*string*] **SubNewView** : Procédure à lancer pour afficher une nouvelle vue standard pour ce compte

Si cette propriété est vide, une nouvelle vue sera ouverte avec les paramètres en cours

[*string*] **SubGraphic** : Procédure à lancer pour afficher une vue graphique pour ce compte

[*string*] **SubInfos** : Procédure à lancer pour afficher une vue Infos pour ce compte

[*string*] **SubFilterDate** : Procédure à lancer pour Filtrer les dates d'affichage de ce compte.

[*string*] **SubProperty** : Procédure à lancer pour Editer les Propriétés du compte

[*string*] **SubHeader** : Procédure à lancer quand un en-tête de colonne est cliqué par l'utilisateur

[*integer*] **HeaderClicked** : N° de la colonne cliquée (pour **SubHeader**)

[*string*] **SubActivate** : Procédure à lancer quand le menu paramétrable est activé ou ENTER pressé

[*string*] **MenuActivate** : Menu à afficher dans le menu paramétrable.

[*string*] **SubCheck** : Procédure à lancer quand la 1<sup>ère</sup> colonne (0) est sélectionnée ou TAB pressé

[*integer*] **Selected** : Position de la ligne actuellement sélectionnée (pour **SubCheck** et **SubActivate**)

[*integer*] **SelectionSize** : Nombre de lignes sélectionnées en + ou – par rapport à **Selected**

[*string*] **CheckItem** : Contenu de la 1<sup>ère</sup> colonne (0) de la ligne concernée par **SubCheck**